Zadania OPSS

wszystkie zadania zostały pobrane ze strony
http://opss.safo.biz/

wersja poprawiona

Bartosz Kolada

Spis treści

Trening: Liczby Fibonacciego....5
Problem Euklidesa....6
Wenusjańskie działki....7
Laboratoryjne rozważania....8
Liczby pierwsze....9
Kwadrat jedynek....10
Piłeczka....11
Silnia....13
Żabka....14
Suma....15
Dziwne własności jedenastu....16
KN - liczby....17
Stary Testament....18
Prosta gra....20
Iloczyn cyfr....21
Superdługa suma....22
Dzielni baloniarze....23
Sortowanie....24
Waga binarna....25
Dwubarwne wieże Hanoi....26
Potęga....27
Tratwa....28
Kamyki....29
Tort....31
Firanki....32
Zagadkowy ciąg....33
AB-drzewo....34
Domino....36
Sznurki....37
Mikołaj....41
Obliczenia równoległe....42
Binarne Bingo....44
Drogi....46
Plaster miodu....48
Hetmani....50
Robot....51
Wojna....52
Włamanie....53
Dr Judym....54
Lotki....56
Liczby wyważone....57
Piętnastka....58
Przeprowadzka....60
Flamaster....61
OPS....62
Automat....63
Obława....65
Kosmiczne sygnały....67
Cwany Lutek....69
Chomiki Edka....70
Skoczek....71

Komputerowa telepatia........73
Czekoladka........75
Liczby HEX-palindromiczne........76
Nadajniki........77
Lublin-Kraków........78
Sztuka przetrwania........79
Segregacja........81
Szyfr........82
Rekursywna bakteria czwórkowa........84
Maszyna drukarska........86
Kąt obrotu........88
Adresy IP........91
Suma cyfr........93
Szkrable........94
Gugle........96
Jedenastka........98
Doskonałe trójkąty pitagorejskie........100
Marsjańskie skały........101
Dźwięczne słowa........103
Pantofelek........104
Optymalizator........106
Hodowla alg........107
Normy........109
Wyrażenia........110
Transport........112
Dzień Dziecka........114
Komputerowa telepatia 3D........116
Zabawa........118
Pikoboty........119
Płyty........121
Starożytna maszyna........122
Licytacja........123
Multyplikator........124
Klocki........125
Wzorce bitowe........127
Generis........128
Geolog........130
Rezerwacje........132
Samotnik........134
Kosmita........135
Łaty........137
Owce........139
Zależności........140
Samoopisujące się liczby........142
Płytki........143
Rachmistrz........144
Ciąg........145
Komórka........146
Weselne toasty........147
Ułamki........149
Protest ekologów........150
Grafy animalne........152

Rybki ... 155
Serwery ... 157
OPSSML ... 158
Giełda Rzeczy Wartościowych ... 161
Czterysta dwadzieścia dwa ... 164
Drogowcy ... 166
Grupy krwi ... 168
Suma cyfr ... 170
Chodnik ... 171
Iloczyny ... 173
Partie ... 174
Robaczek ... 176

Trening: Liczby Fibonacciego

ID:1000

Limit czasu: 1.00 s

Limit pamięci: 1024 kB

Liczby Fibonacciego zdefiniowane są następująco:

F(0)=1,

F(1)=1,

F(n)=F(n-1)+F(n-2), n>1.

Jaka jest wartość **F(n)**?

Zadanie

Napisz program, który dla każdego zestawu danych:

- Wczyta liczbę *n* ze standardowego wejścia,
- Obliczy wartość **F(n)**,
- Zapisze wynik na standardowe wyjście.

Wejście

Pierwsza linijka wejścia zawiera dokładnie jedną liczbę całkowitą d, $1 \le d \le 1000$, określającą liczbę zestawów danych. Każdy zestaw zajmuje jedną linijkę wejścia i zawiera dokładnie jedną liczbę naturalną *n* z przedziału [0..45].

Wyjście

I-ta linia wyjścia powinna zawierać dokładnie jedną wartość równą **F(n),** gdzie *n* jest liczbą z *i*-tego zestawu.

Przykład

Dla następującego wejścia:

```
3
4
9
14
```

poprawną odpowiedzią jest:

```
5
55
610
```

Problem Euklidesa

ID: 1001

Limit czasu: 1.00 s

Limit pamięci: 1536 kB

Dawno, dawno temu pewien uczony mędrzec Euklides miał zagadkę nie do rozwiązania. Myślał, myślał i nie mógł znaleźć odpowiedzi na nurtujące go pytanie: "Jak znaleźć największą liczbę naturalna taka, aby dzieliła dwie wybrane liczby naturalne?". Pomóż Euklidesowi rozwiązać dręczący go problem...

Zadanie

Napisz program, który dla każdego zestawu danych:

- Wczytuje liczby a,b ze standardowego wejścia,
- Oblicza wartość **NWD(a,b)**,
- Wypisuje wynik na standardowe wyjście.

Wejście

Pierwsza linia zawiera dokładnie jedna liczbę n, $1 \leq n \leq 50000$, będąca liczba zestawów danych. W n kolejnych liniach występują poszczególne zestawy danych. Każdy zestaw składa się z dwóch liczb a,b, $1 \leq a,b \leq 100000000$, oddzielonych pojedynczą spacją.

Wyjście

Program powinien wypisać na standardowe wyjście n linii. I-ta linia powinna zawierać dokładnie jedna liczbę naturalna, będącą największym wspólnym dzielnikiem liczb występujących w i-tym zestawie danych.

Przykład

Dla następującego wejścia:

```
3
2 6
4 14
5 7
```

poprawnym rozwiązaniem jest:

```
2
2
1
```

Wenusjańskie działki

ID:1002

Limit czasu: 1.00 s

Limit pamięci: 1536 kB

Na planecie Wenus urządzono spis majętności mieszkańców. Każdy Wenusjanin musiał powiedzieć, jak duża jest jego działka. Ale, jak to w kosmosie, sposób mierzenia działek jest dosyć skomplikowany. Każda działka ma swój rozmiar bazowy i wymiar, natomiast miara wielkości działki jest rozmiar bazowy działki podniesiony do potęgi wymiar. Wenusjanie nie mogą sobie poradzić z taka metoda przeprowadzania pomiaru. Pomóż im!

Zadanie

Napisz program, który dla każdego zestawu danych:

- Wczytuje liczby *a,b* ze standardowego wejścia,
- Oblicza wartość $\mathbf{a}^{\mathbf{b}}$,
- Wypisuje wynik na standardowe wyjście.

Wejście

Pierwsza linia zawiera dokładnie jedna liczbę *n*, 1<=*n*<=200000, będąca liczbą zestawów danych. W n kolejnych liniach występują poszczególne zestawy danych. Każdy zestaw składa się z dwóch liczb *a,b,* 1=<*a*<=5, 1<=*b*<=10, oddzielonych pojedyncza spacja..

Wyjście

Program powinien wypisać na standardowe wyjście n linii. I-ta linia powinna zawierać dokładnie jedną liczbę naturalna, będącą wartością wyrażenia $\mathbf{a}^{\mathbf{b}}$, gdzie *a,b* to liczby występujące w *i*-tym zestawie danych.

Przykład

Dla następującego wejścia:

```
3
2 6
3 3
1 10
```

poprawnym rozwiązaniem jest:

```
64
27
1
```

Laboratoryjne rozważania

ID:1003

Limit czasu: 1.00 s

Limit pamięci: 1536 kB

W laboratorium botanicznym Instytutu Doświadczalnego Matfizańskiego Uniwersytetu Nauk Przyrodniczych zasiano unikalną roślinę. Pierwszego dnia, co oczywiste, jej łodyga miała wysokość zero. Każdego kolejnego dnia dokonywano pomiaru o ile wydłużyła się łodyga w stosunku do dnia poprzedniego. Dokonano zaskakującego odkrycia: każdego dnia roślina wzrastała o całkowita liczbę centymetrów! Napisz program, który umożliwi botanikom obliczenie wysokości rośliny po określonej liczbie przeprowadzonych pomiarów.

Zadanie

Napisz program, który dla każdego zestawu danych:

- Wczytuje liczby $n, a_1,...,a_n$ ze standardowego wejścia,
- Oblicza wartość $a_1+a_2+...+a_n$,
- Wypisuje wynik na standardowe wyjście.

Wejście

Pierwsza linia zawiera dokładnie jedną liczbę naturalną C, $1<=C<=200000$, będącą liczbą zestawów danych. W C kolejnych liniach występują poszczególne zestawy danych. Każdy zestaw składa się z liczby n, $1=<n<=100000$, oraz następujących po niej n liczb a_i,$0<=a_i<=1000$, $1<=i<=n$, oddzielonych pojedyncza spacja.

Wyjście

Program powinien wypisać na standardowe wyjście C linii. I-ta linia powinna zawierać dokładnie jedną liczbę naturalną, będącą sumą liczb $a_1,..,a_n$, gdzie n oraz a_i, $1<=i<=n$ to liczby występujące w i-tym zestawie danych.

Przykład

Dla następującego wejścia:

```
2
3 1 2 3
4 0 2 0 0
```

poprawnym rozwiązaniem jest:

```
6
2
```

Liczby pierwsze

ID:1004

Limit czasu: 1.50 s

Limit pamięci: 1536 kB

Liczba pierwsza to liczba naturalna większa od 1, która dzieli się tylko przez 1 i przez siebie samą (a zatem ma dokładnie dwa dzielniki naturalne).

Zadanie

Napisz program, który wyznacza *n*-tą w kolejności liczbę pierwszą.

Wejście

Pierwsza linia zawiera dokładnie jedną liczbę *C*, 1<=*C*<=200000, będącą liczbą zestawów danych. W *C* kolejnych liniach występują poszczególne zestawy danych. Każdy zestaw składa się z jednej liczby naturalnej 1<=*n*<=15000.

Wyjście

Program powinien wypisać na standardowe wyjście *C* linii. *I*-ta linia powinna zawierać dokładnie jedną liczbę naturalną, będącą *n*-tą w kolejności liczbą pierwszą dla zestawu o numerze *I*.

Przykład

Dla następującego wejścia:

```
4
3
2
5
7
```

poprawnym rozwiązaniem jest:

```
5
3
11
17
```

Kwadrat jedynek

ID:1005

Limit czasu: 1.00 s

Limit pamięci: 1536 kB

Pewnego Marokańczyka Ibn-al-Banna fascynowały właściwości liczb składających się z samych jedynek (1, 11, 111, 1111, 11111, 111111, ...).

Zadanie

Pomóż Ibn-al-Bannanowi w znalezieniu kwadratu liczby składającej się z samych jedynek.

Wejście

Pierwsza linia zawiera dokładnie jedną liczbę k, $1 \leq k \leq 500$, będącą liczbą zestawów danych. W k kolejnych liniach występują poszczególne zestawy danych. Każdy zestaw składa się z jednej linii zawierającej dokładnie jedną liczbę n ($0 < n \leq 200$) oznaczającą liczbę jedynek w liczbie, którą należy podnieść do kwadratu.

Wyjście

Program powinien wypisać na standardowe wyjście k linii. Dla każdego zestawu danych program powinien wypisać kwadrat wejściowej liczby składającej się z n jedynek.

Przykład

Dla wejścia:

```
1
2
```

poprawnym wyjściem jest:

```
121
```

Piłeczka

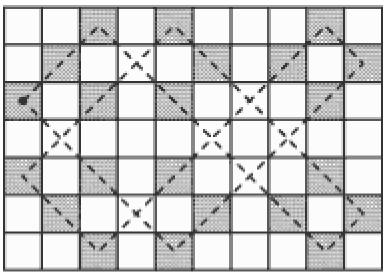
ID:1006

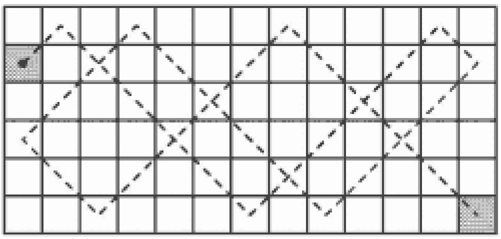
Limit czasu: 1.00 s

Limit pamięci: 1024 kB

Na prostokątnej szachownicy składającej się z *m x n* kwadratowych pól wybieramy jedno pole znajdujące się na krawędzi szachownicy - nazywamy je "polem startowym". Następnie umieszczamy na jego środku piłeczkę i wprawiamy ją w ruch tak, aby toczyła się po szachownicy. Średnica piłeczki równa jest szerokości (i wysokości) każdego z pól szachownicy. Kąt pomiędzy kierunkiem ruchu piłeczki a krawędzią szachownicy wynosi 45 stopni. Piłeczka odbija się od krawędzi szachownicy następująco: jeżeli piłeczka dotknie krawędzi szachownicy wtedy każda składowa prędkości piłeczki prostopadła do krawędzi, z którą nastąpiło zetkniecie, zostaje odwrócona. Na początku piłeczka zostaje rozpędzona w kierunku wzrostu wartości współrzędnych (w przypadku gdy pole startowe ma największa wartość której ze współrzędnych, piłeczka natychmiastowo odbija się od krawędzi).

Przypisujemy punkt polu szachownicy za każdym razem gdy piłeczka przetacza się przez obszar wnętrza pola. Grę uważamy za zakończona jeżeli punkt zostanie przypisany polu startowemu. Jaka jest liczba pól szachownicy, którym przypisano nieparzysta liczbę punktów? Poniższe rysunki obrazują problem. Trasa piłeczki zaznaczona jest linią przerywaną. Pola z nieparzystą liczba punktów są zaznaczone kolorem szarym.





Zadanie

Napisać program, który dla każdego zestawu danych wejściowych, należącego do ciągu kilkunastu zestawów danych:

- wczytuje ze standardowego wejścia wymiary szachownicy oraz współrzędne pola startowego,
- znajduje liczbę pól, którym przypisano nieparzystą liczbę punktów,
- wypisuje wynik na standardowe wyjście,

Wejście

Pierwsza linia wejścia zawiera jedną liczbę całkowita *d,* 1<=*d*<=20, która oznacza liczbę zestawów danych. Po niej, w kolejnych liniach, następują zestawy danych wejściowych - każdy zestaw znajduje się w jednej linii. Linia taka zawiera cztery liczby całkowite *x, y*, a, b oddzielone pojedynczymi spacjami. Liczby te to odpowiednio: *x*'owy i *y*'owy wymiar szachownicy oraz *x*'owa oraz *y*'owa współrzędna pola startowego. Liczby x oraz y są większe niż 2, liczba pól szachownicy nie przekracza 10^9, pole startowe leży na krawędzi szachownicy.

Wyjście

I-ta linia wyjścia powinna zawierać jedną liczbę całkowita, która jest równa liczbie pól szachownicy, którym przypisano nieparzystą liczbę punktów.

Przykład:

Dla wejścia:

```
2
10 7 1 5
13 6 1 5
```

poprawna odpowiedzią jest:

```
22
2
```

Silnia

ID:1007

Limit czasu: 1.00 s

Limit pamięci: 1536 kB

Niech n będzie nieujemną liczbą całkowitą. Liczbę n! (czytaj n-silnia) definiuje się następująco:

Jeśli $n \leq 1$, to $n! = 1$. Dla $n > 1$, $n!$ jest równe iloczynowi wszystkich liczb od 1 do n, czyli $n! = 1*2*...*n$. Na przykład 4!=1*2*3*4=24.

Zadanie

Napisz program, który:

- wczyta ze standardowego wejścia nieujemną liczbę całkowitą n,
- policzy cyfrę jedności w zapisie dziesiętnym liczby $n!$,
- wypisze wynik na standardowe wyjście.

Wejście

Pierwszy i jedyny wiersz standardowego wejścia zawiera dokładnie jedną nieujemną liczbę całkowitą n, $0 \leq n \leq 30000$.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia Twój program powinien zapisać dokładnie jedną cyfrę równą cyfrze jedności w zapisie dziesiętnym liczby n!.

Przykład

Dla danych wejściowych:

```
4
```

poprawną odpowiedzią jest:

```
4
```

Żabka

ID:1010

Limit czasu: 1.00 s

Limit pamięci: 1536 kB

Żabka porusza się po prostej ścieżce złożonej z *n* pól. Startowe położenie to pole numer 1, a końcowe - *n*. Żabka porusza się tylko w kierunku pola końcowego. Żabka może przesunąć się jednym skokiem minimalnie o *kmin* pól, a maksymalnie o *kmax* pól.

Zadanie

Napisz program, który obliczy na ile sposobów żabka może osiągnąć swoje położenie końcowe, jeżeli porusza się tak że każdy następny skok jest co najmniej takiej samej długości jak poprzedni.

Wejście

Pierwsza linia zawiera dokładnie jedną liczbę m, $1 \leq m \leq 100$, będąca liczbą zestawów danych. W m kolejnych liniach występują poszczególne zestawy danych. Każdy zestaw składa się z liczb naturalnych *n, kmin, kmax,* oddzielonych pojedynczą spacją. ($2 \leq n \leq 1000$, $1 \leq kmin \leq kmax \leq 1000$).

Wyjście

Program powinien wypisać na standardowe wyjście *m* linii. *I*-ta linia powinna zawierać dokładnie jedną liczbę naturalną, będąca liczbą możliwych ruchów dających żabce osiągniecie pola końcowego. Liczba ta nie będzie większa od 2^{31}.

Przykład

Dla następującego wejścia:

```
4
10 2 5
15 5 8
20 1 20
22 2 7
```

poprawnym rozwiązaniem jest:

```
5
2
490
72
```

Suma

ID:1011

Limit czasu: 2.00 s

Limit pamięci: 1536 kB

Zadanie

Twoim zadaniem jest napisać program, który wyliczy sumę wszystkich liczb całkowitych leżących pomiędzy 1 a *N* (włącznie).

Wejście

Pierwsza linia zawiera dokładnie jedną liczbę naturalną *n*, 1<=*n*<=200000, będącą liczbą zestawów danych. W *n* kolejnych liniach występują poszczególne zestawy danych. Każdy zestaw składa się z jednej liczby całkowitej N.

Wyjście

Program powinien wypisać na standardowe wyjście *n* linii. I-ta linia powinna zawierać sumę wszystkich liczb całkowitych leżących pomiędzy 1 a N. Gwarantujemy, że wartość bezwzględna sumy nie przekroczy 2^31.

Przykład

Dla następującego wejścia:

```
4
3
2
5
7
```

poprawnym rozwiązaniem jest:

```
6
3
15
28
```

Dziwne własności jedenastu

ID:1012

Limit czasu: 1.00 s

Limit pamięci: 1536 kB

Jedenaście jest na swój sposób "magiczną" liczbą. Aby otrzymać dowolną potęgę liczby 11 nie trzeba mozolnie mnożyć 11 * 11 * 11 * ... (n czynników) można osiągnąć to nieco łatwiej np. przez zbudowanie takiej piramidy:

```
      1 + 1 = 2
    1 + 2 + 1 = 4
  1 + 3 + 3 + 1 = 8
1 + 4 + 6 + 4 + 1 = 16
........................
```

Zadanie

Twoim zadaniem jest wyznaczyć liczbę M = 11^n.

Wejście

Pierwsza linijka wejścia określa liczbę zestawów danych ($0 < i \leq 500$). Każdy zestaw danych składa się z jednej linijki, w której pojawia się liczba n ($0 \leq n \leq 200$).

Wyjście

Dla każdego zestawu danych odpowiedz powinna składać się z jednej liczby M (M = 11^n).

Uwaga. Liczba cyfr liczby M nie przekroczy 256.

Przykład

Dla następującego wejścia:

```
2
3
10
```

poprawnym wyjściem jest:

```
1331
25937424601
```

KN - liczby

ID:1013

Limit czasu: 1.00 s

Limit pamięci: 1024 kB

Rozważmy liczby zawierające dokładnie N cyfr, zapisane w systemie K.

Liczbę nazwiemy KN-poprawną, jeśli w jej zapisie w systemie K nie wystąpią dwa sąsiadujące ze soba zera.

Na przykład:

1010230 jest poprawną 7-cyfrowa liczbą KN (n=7, k=4)

1000198 nie jest poprawną liczbą KN (sąsiadujące zera)

0121235 nie jest 7-cyfrową, lecz 6-cyfrową liczbą KN (n=7, k=7).

Zadanie

Mając dane dwie liczby naturalne N i K, napisz program, który wyznaczy liczbę wszystkich poprawnych KN liczb.

Możesz założyć, że: $2 \leq K \leq 10$; $2 \leq N$; $4 \leq N+K \leq 18$.

Wejście

Dwie naturalne liczby N i K .

Wyjście

Liczba wszystkich poprawnych KN liczb.

Przykład

Dla wejścia:

```
2
10
```

poprawnym rozwiązaniem jest:

```
90
```

Stary Testament

ID:1014

Limit czasu: 1.50 s

Limit pamięci: 1536 kB

Mnisi podczas studiowania Starego Testamentu natrafili na liczne fragmenty zapisane szyfrem atbasz (tradycyjną hebrajską odmianią szyfru podstawieniowego).

Atbasz polega na zastąpieniu litery położonej w pewnym miejscu, licząc od początka alfabetu, literą położoną w tym samym miejscu licząc od końca.

Pracowici mnisi stworzyli różne alfabety, wypisali wszystkie odnalezione fragmenty i uświadomili sobie jak dużo jeszcze przed nimi pracy.

Pomoż zatem biblistom (bo zależy im na wiedzy i czasie) rozkodować znalezione fragmenty.

Wejście:

W pierwszym wierszu znajduje się alfabet (są to znaki ascii z przedziału [33..126] czyli ['!'..'~']. Nie muszą być posortowane zgodnie z ich kodami ascii! W następnej linii znajduje sie liczba 0 < N < 1000000 fragmentów znalezionych przez mnichów.

W kolejnych N liniach znajdują się zakodowane słowa szyfrem atbasz (każde zawiera minimalnie jeden znak i maksymalnie 32 znaki)

Wyjście:

W kolejnych N wierszach powinny pojawić się słowa rozkodowane przez Twój program zgodnie z kolejnością ich wypisania przez mnichów

Przykład:

Dla wejścia:

```
ZaBoW9#At
3
ZtA
Bo99###aa
ttAA##99WWooBBaZ
```

Poprawnym rozwiązaniem jest:

```
tZa
#9ooBBBAA
ZZaaBBooWW99##At
```

Uwaga:

Litery w alfabecie nie powtarzają się.

Rozróżniana jest wielkość liter.

Prosta gra

ID:1015

Limit czasu: 1.00 s

Limit pamięci: 1024 kB

Mamy do dyspozycji szachownicę nieskończonej wielkości oraz znajdujące się na niej pionki ułożone w prostokąt (wypełniony) o wymiarach $m \times n$ ($1 \leq m, n \leq 1000$). Pionki leżą na polach planszy. Jeden pionek leży dokładnie na jednym polu planszy.

Rozpoczynamy rozgrywkę dla jednego gracza, zgodnie z następującymi zasadami gry. Każdy pionek może przeskoczyć nad innym pionkiem ("zbić go") wzdłuż linii poziomej lub pionowej na planszy. Pionka, który został zbity, usuwamy z planszy i wykluczamy go z dalszej gry. Celem gry jest uzyskanie jak najmniejszej liczby pionków na szachownicy.

Zadanie

Mając zadaną parę liczb m oraz n, oddzielonych od siebie w pliku wejściowym spacją, należy napisać program, który określi najmniejszą możliwą liczbę pionków pozostałych na szachownicy.

Wejście

Liczby m i n oddzielone spacją.

Wyjście

Najmniejsza liczba pionków pozostałych na szachownicy.

Przykład

Dla wejścia:

3 4

poprawnym rozwiązaniem jest:

2

Iloczyn cyfr

ID:1016

Limit czasu: 0.50 s

Limit pamięci: 1024 kB

Twoim zadaniem jest znaleźć najmniejszą dodatnią liczbę całkowitą Q, tak aby iloczyn jej cyfr był równy N.

Wejście

Wejście zawiera dokładnie jedną liczbę naturalną N ($0 \leq N \leq 10^9$).

Wyjście

Twój program powinien wypisać dokładnie jedną liczbę Q spełniającą warunki zadania. Jeśli taka liczba nie istnieje program powinien wypisać liczbę -1.

Przykład

Dla wejścia:

```
10
```

poprawną odpowiedzią jest:

```
25
```

Superdługa suma

ID:1017

Limit czasu: 2.00 s

Limit pamięci: 512 kB

Twórcy nowego języka programowania D++ stwierdzili, że jakiegokolwiek limitu by nie stworzyli dla typu SuperLongInt, programiści i tak czasami muszą operować jeszcze większymi liczbami. Ograniczenie do 1000 cyfr okazało się zbyt małe... Twoim zadaniem jest znalezienie sumy dwóch liczb naturalnych, których maksymalna liczba cyfr wynosi 1000000.

Wejście

Pierwsza linia wejścia zawiera jedną liczbę naturalną N ($1 \le N \le 1000000$), określającą liczbę cyfr wejściowych liczb. (aby długości liczb były takie same, mogą być dodane na początku nieznaczące zera). Następnie na wejściu podane są te dwie liczby - wypisane są one w dwóch kolumnach, oddzielonych jedną spacją. Każda kolumna określą jedną liczbę. Obie liczby są większe od 0, a długość ich sumy nie przekracza N.

Wyjście

Wyjście powinno zawierać dokładnie N cyfr w jednej linii, reprezentujących sumę tych dwóch liczb.

Przykład

Dla wejścia:

```
4
0 4
4 2
6 8
3 7
```

poprawną odpowiedzią jest:

```
4750
```

Dzielni baloniarze

ID:1018

Limit czasu: 1.00 s

Limit pamięci: 1024 kB

Dziesięciu matematyków leci balonem nad Oceanem Spokojnym. Kiedy przekroczyli równik, zdecydowali uczcić to zdarzenie, otwierając butelkę szampana. Niestety, korek wybił dziurę w balonie. Wodór zaczął się ulatniać, powodując opadanie balonu. Wkrótce spadnie do oceanu, a baloniarze zostaną zjedzeni przez wygłodniałe rekiny.

Ale nie wszystko stracone. Jeden z baloniarzy może się poświęcić wyskakując z balonu, w celu przedłużenia życia pozostałym chociaż na krótką chwilę. Ciągle istnieje jeden problem - kto ma wyskoczyć? Matematycy wymyślili, że każdy z nich napisze liczbę naturalną ai nie mniejszą niż 1 i nie większą niż 10000. Następnie obliczają magiczną liczbę N, która jest liczbą wszystkich dodatnich dzielników iloczynu a1*a2*...*a10. Na przykład, liczbą dodatnich dzielników liczby 6 jest 4 (dzielniki: 1,2,3,6). Bohater (matematyk, który wyskoczy) jest wyznaczony przez ostatnią cyfrę N.

Twoim zadaniem, jest wyznaczenie tej cyfry.

Wejście

Wejście zawiera dziesięć liczb, każda w oddzielnej linii.

Wyjście

Wyjście powinno zawierać jedną cyfrę 0-9 (ostatnia cyfra liczby N).

Przykład

Dla wejścia:

```
1
2
6
1
3
1
1
1
1
1
```

poprawną odpowiedzią jest:

```
9
```

Sortowanie

ID:1019

Limit czasu: 1.00 s

Limit pamięci: 1024 kB

Frma MIRACLE, znany potentat na rynku baz danych, opracowała nowy sposób sortowania, który będzie zastosowany w najnowszym produkcie o nazwie kodowej Miracle 13k. W obecnej wersji metoda pozwala sortować niemalejąco tylko liczby naturalne z przedziału 1..1000.

Algorytm polega na zamianie miejscami dwóch sąiednich elementów. Na przykład - jeśli chcemy posortować ciąg 3 1 2 to musimy zamienić miejscami 1 i 3 - otrzymamy ciąg 1 3 2 - nastepnie 3 - 2. Wykonaliśmy zatem 2 zamiany. W celu wytestowania metody potrzebny jest program, który policzy minimalną liczbę zamian potrzebną do posortowania ciągu liczb z przedziału 1..1000. Niestety, programista pracujący nad algorytm miał wypadek i zadanie to powierzono Tobie.

Wejście

W pierwszej linii znajduje się liczba 1≤N≤1000 określająca ilość liczb w ciągu. W drugiej lini jest N liczb z przedziału 1..1000 rozdzielonych spacją.

Wyjście

Na wyjściu powinna się znaleźć jedna liczba określająca minimalną liczbę zamian w ciągu wejściowym.

Przykład

Dla następujących danych:

```
3
3 1 2
```

poprawnym rozwiązaniem jest:

```
2
```

Waga binarna

ID:1020

Limit czasu: 1.50 s

Limit pamięci: 2048 kB

Waga binarna to specyficzne urządzenie, które może dokonywać pomiarów dowolnych wielkości z przedziału (0,1) z ustaloną dokładnością. Dokładność wagi ustala się pokrętłem, które można ustawić na pozycji 1 lub 2, lub 3, lub ..., lub 10. Gdy dokładność jest ustawiona na ^{m}, to waga dokonuje pomiarów z dokładnością do $1/2^m$. Wyniki pomiarów wagi są zapisywane w postaci par (l,m). Taka para oznacza, że dokładność wagi jest ustawiona na m i wskazanie wagi wynosi l, czyli ciężar ważonego przedmiotu wynosi $l/2^m$ (l jest liczbą naturalną i oczywiście $0 < l < 2^m$, gdyż wspominaliśmy, że waga wskazuje wielkości z przedziału (0,1)).

Zadanie

Twoim zadaniem jest napisanie programu, który uporządkuje wyniki pomiarów od najmniejszych do największych. Wyniki pomiarów zadane są w postaci par (l,m). Różne pary oznaczające takie same wyniki (np. (1,2) i (2,3) należy uporządkować rosnąco według wskazań, czyli pierwszych elementów w parach.

Wejście

Program powinien czytać dane z wejścia standardowego. W pierwszym wierszu danych podana jest liczba n ($1 \le n \le 20000$) oznaczająca liczbę par. W kolejnych n wierszach podane są pary liczb l_i i m_i, po jednej parze w wierszu; l_i i m_i są oddzielone jedną spacją. Dla każdej pary spełnione są warunki: $1 \le m_i \le 10$ oraz $0 < l < 2m_i$.

Wyjście

Program powinien pisać wynik na wyjście standardowe. Wynikiem powinno być n par liczb podanych na wejściu, ale w takiej kolejności, by pary odpowiadające mniejszym wartościom pomiarów występowały przed parami odpowiadającymi większym wartościom. Takie same pomiary należy zapisać niemalejąco według wskazań. Każdą parę należy zapisać w takiej samej postaci, w jakiej była podana na wejściu.

Przykład

Dla danych wejściowych:

```
4
1000 10
3 10
5 3
250 8
```

poprawną odpowiedzią jest:

```
3 10
5 3
250 8
1000 10
```

Dwubarwne wieże Hanoi

ID:1021

Limit czasu: 1.00 s

Limit pamięci: 1024 kB

Wieże Hanoi to tradycyjna zabawa-łamigłówka polegająca na nakładaniu krążków na słupki. Dysponujemy n krążkami o średnicach 1,2, ...,n i trzema słupkami, które nazwiemy A, B i C. Każdy krążek ma w środku dziurkę, która pozwala nałożyć krążek na słupek. Początkowo wszystkie krążki znajdują się na słupku A i są ułożone począwszy od największego (na dole) do najmniejszego (na górze). Zabawa polega na przeniesieniu wszystkich krążków na jeden z wolnych słupków (powiedzmy B) zgodnie z następującymi zasadami:

- w jednym ruchu wolno nam wziąć jeden krążek leżący na górze na jednym ze słupków i położyć go na górze na innym słupku;
- na każdym słupku zawsze musi być zachowany porządek, tzn. krążki muszą leżeć w kolejności od największego (na dole słupka) do najmniejszego (na górze).

Krążki nałożone na jeden słupek nazwiemy wieżą. Podsumowując powyższe zasady, możemy stwierdzić, że:

- nie jest możliwe wyciągnięcie krążka ze środka wieży lub włożenie krążka do środka wieży;
- nie wolno brać więcej niż jeden krążek na raz;
- nie wolno kłaść większego krążka na mniejszym.

Celem w tej zabawie jest przeniesienie wieży z jednego słupka na drugi w najmniejszej, możliwej liczbie ruchów.

Dwubarwne wieże Hanoi, to nieco zmodyfikowana odmiana powyższej układanki. Jak poprzednio mamy trzy słupki i n krążków o średnicach 1,2,...n. Tym razem jednak krążki o średnicach nieparzystych (1,3,5,...) są białe, a krążki o średnicach parzystych (2,4,6,...) są czarne. Celem zabawy jest przeniesienie (zgodnie z podanymi wyżej zasadami) wszystkich krążków białych na słupek B, a krążków czarnych na słupek C.

Zadanie

Napisz program, który wyliczy minimalną liczbę ruchów potrzebnych do ułożenia krążków białych na słupku B, a krążków czarnych na słupku C.

Wejście

Program powinien czytać dane z wejścia standardowego. W pierwszym wierszu podana jest liczba n ($0 \le n \le 1000$) oznaczająca liczbę krążków.

Wyjście

Program powinien pisać wynik na wyjście standardowe. Wynikiem powinna być jedna liczba oznaczająca minimalną liczbę ruchów potrzebnych do rozdzielenia białych i czarnych krążków.

Przykład

Dla danych wejściowych:

```
6
```

poprawną odpowiedzią jest: `45`

Potęga

ID:1022

Limit czasu: 1.00 s

Limit pamięci: 1024 kB

Dla danej nieujemnej liczby całkowitej n należy znaleźć cyfrę jedności (ostatnią cyfrę w zapisie dziesiętnym) liczby 3^n.

Zadanie

Napisz program, który dla każdego zestawu danych:

- wczyta liczbę n ze standardowego wejścia,
- wyznaczy cyfrę jedności liczby $3n$,
- zapisze wynik na standardowe wyjście.

Wejście

Pierwsza linijka wejścia zawiera dokładnie jedną liczbę całkowitą d, $1 \leq d \leq 10$, określającą ilość zestawów danych. W kolejnych d liniach wejścia znajdują się zestawy danych. Każdy zestaw danych składa się z jednej linii i zawiera jedną nieujemną liczbę całkowitą n, $0 \leq n < 10200$.

Wyjście

Twój program powinien wypisać d liczb, każdą w osobnym wierszu. Liczba w wierszu i powinna byc odpowiedzią dla i-tego zestawu danych.

Przykład

Dla danych wejściowych:

```
3
2
3
2005
```

poprawną odpowiedzią jest:

```
9
7
3
```

Tratwa

ID:1023

Limit czasu: 1.00 s

Limit pamięci: 1024 kB

Spróbuj rozwiązać następującą zagadkę:

"Statek płynie z Warszawy do Gdańska dobę, a z Gdańska do Warszawy dwie doby. Ile płynie tratwa z Warszawy do Gdańska?"

Krótkie wyjaśnienie: statek płynie po Wiśle i posiada własny napęd o stałej w czasie mocy, a zatem posiada stałą prędkość względem wody. Kiedy płynie z prądem, pokonuje ten sam dystans szybciej niż płynąc pod prąd. Tratwa nie posiada własnego napędu i porusza się z prędkością prądu Wisły.

Wyjątkowo podamy rozwiązanie zagadki. Tratwa płynie z Warszawy do Gdańska 4 doby. Twoim zadaniem będzie jednak rozwiązanie tego problemu dla dowolnych danych.

Wejście:

W jedynym wierszu wejścia znajdują się 2 liczby całkowite: N, M, $1 \leq N < M \leq 100000$, gdzie N oznacza czas w jakim statek pokonuje trasę z prądem, a M oznacza czas w jakim ten sam statek, pokonuje tę samą trasę pod prąd.

Wyjście:

Wynikiem jest jeden wiersz zawierający liczbę oznaczającą czas w jakim trasę pokonuje tratwa. Gwarantujemy, że dane będą tak dobrane, że wynikiem będzie dodatnia liczba całkowita mniejsza od 2.

Przykład:

Dla danych:

```
1 2
```

poprawną odpowiedzią jest:

```
4
```

Kamyki

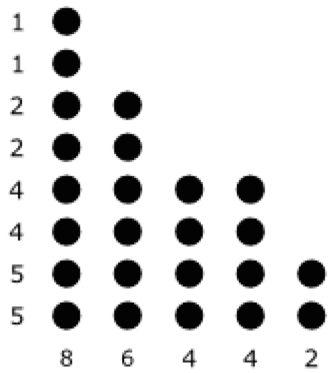
ID:1024

Limit czasu: 1.00 s

Limit pamięci: 1536 kB

Pewien matematyk Dobromir oraz pewien fizyk Albert z zamiłowania zajmują się układaniem specyficznych układanek z drobnych kamyków. Ich układanki posiadają ciekawą właściwość: składają się z ciągu kolumn ułożonych z kamyków tak, że każda następna kolumna (licząc od lewej strony) zawiera nie więcej kamyków niż poprzednia. Wszystkie kolumny zaś są "wyrównane" do dołu (zaczynają się od najniższego wiersza układanki).

Swoje układanki Dobromir zwykł opisywać, jak na matematyka przystało, jako ciąg liczb, używając następującej konwencji (nazwijmy ją umownie konwencją M): pierwsza liczba oznacza ilość wszystkich kamyków w całej układance, zaś kolejne liczby oznaczają ilość kamyków w kolejnych kolumnach, rozpoczynając od kolumny w której jest najwięcej kamyków. Albert, nie chcąc być gorszym od Dobromira, wymyślił swój, nieco inny sposób opisywania układanek (nazwijmy go umownie konwencją F). Używał w tym celu również ciągu liczb, ale takiego, w którym kolejne liczby oznaczają ilość kamyków w poszczególnych wierszach układanki (rozpoczynając od wiersza znajdującego się na samym dole układanki, w którym jest najwięcej kamyków).



Rys. Przykładowy układ kamyków (konwencja M: 24 8 6 4 4 2; konwencja F: 5 5 4 4 2 2 1 1).

Pewnego dnia obaj uczeni spotkali się, aby porównać kolekcje swoich układanek, okazało się bowiem, że wiele z nich znajduje się w zbiorach obydwu panów. Chcieli takie egzemplarze wyodrębnić, jednakże metoda wizualna porównywania układanek "na oko" okazała się bardzo żmudna i mało skuteczna. Dobromir i Albert postanowili zatem porównywać swoje dzieła wg opisów. Pojawiła się jednak kolejna przeszkoda: przecież opisy układanek dokonywane są w różnych konwencjach! Podjęli wspólnie decyzję, że będą "przepisywać" opisy Dobromira do formatu, w jakim dokonywał swoich opisów Albert. Pomóż uczonym przebrnąć przez ogromną ilość konwersji opisów układanek z konwencji M do konwencji F...

Wejście:

W pierwszym wierszu wejścia znajduję się liczba całkowita C, $1 \le C \le 100$, oznaczająca liczbę zestawów danych. W kolejnych wierszach znajdują się zestawy danych, z których każdy zawiera dokładnie 1 wiersz, będący opisem pewnej układanki w konwencji *M.* W wierszu każdego zestawu znajduje się liczba *N,* oznaczająca ilość wszystkich kamyków w układance, $1 \leq N \leq 1000000$, oraz po spacji oddzielone pojedynczymi spacjami liczby a_1, a_2, ..., a_i ($1 \leq i \leq 1000$; $1 \leq a_i \leq 1000$) określające ilości kamyków w kolejnych kolumnach układanki.

Wyjście:

W *C* wierszach wyjścia należy podać wyznaczony dla każdego zestawu danych opis układanki w konwencji *F*. Kolejne liczby stanowiące opis powinny być oddzielone pojedynczymi spacjami.

Przykład:

Dla danych:

```
2
24 8 6 4 4 2
19 5 3 3 3 2 1 1 1
```

poprawną odpowiedzią jest:

```
5 5 4 4 2 2 1 1
8 5 4 1 1
```

Tort

ID:1025

Limit czasu: 1.50 s

Limit pamięci: 4096 kB

Małgosia przygotowywała urodzinowe przyjęcie. Upiekła tort i zajęła się układaniem listy gości.

Liczbę gości N wyznaczała następująco: wybrała pewną liczbę cięć K, a następnie podzieliła tort K prostymi na maksymalną możliwą liczbę kawałków N. Każdy zaproszony gość dostałby więc dokładnie jeden kawałek tortu.

Małgosia nie mogła się zdecydować jak dobrać liczbę cięć K. Zapisywała więc liczby cięć i odpowiadające im liczby gości w prototypowej bazie danych Miracle 13k, nad którą pracował jej ojciec. Wersje testowe (i nie tylko testowe) tej bazy mają to do siebie, że ulegają awarii w najbardziej nieodpowiednich momentach. Nie inaczej było tym razem. Pech chciał, że część danych uległa zniszczeniu i udało się odzyskać tylko liczby gości.

Jak teraz Małgosia ma podzielić tort? Los przyjęcia spoczął w rękach ojca Małgosi. Jak wiadomo - nieszczęścia chodzą parami - rodzic Małgosi został wezwany na niezwykle ważne konsultacje do siedziby firmy Miracle i wróci dopiero na przyjęcie. Kolejny już raz cała nadzieja spoczęła w Twoich rękach. Pomóż Małgosi podzielić tort.

Wejście

Na wejściu znajduje się liczba zestawów danych C, $1 \le C \le 65535$. W kolejnych C wierszach znajdują się liczby N, $1 \le N < 2^{31}$, określające maksymalną ilość gości zaproszonych na urodziny.

Wyjście

Dla każdej liczby N, na wyjściu powinna się znaleźć liczba cięć K jakie wykonać powinna Małgosia.

Przykład

Dla wejścia:

```
4
1
2
4
7
```

poprawną odpowiedzią jest:

```
0
1
2
3
```

Firanki

ID:1026

Limit czasu: 3.00 s

Limit pamięci: 6144 kB

Jeśli zdarzyło Ci się zawieszać firanki, być może zauważyłeś, że bardzo dobrym pomysłem na równomierne zaczepienie firanki za pomocą żabek jest przypięcie końców firanki do skrajnych żabek, a następnie wyznaczenie środkowej żabki i przypięcie jej na środku firanki. Powstają w ten sposób 2 nieprzypięte obszary firanki, które przypinamy analogicznie (rekursywnie) - wyznaczamy ponownie środkową żabkę i przypinamy ją na środku firanki, itd.

Jednak nie zawsze możemy wyznaczyć środkową żabkę, zwłaszcza wtedy gdy musimy ją wybrać z parzystej liczby żabek. Chcąc przyczepić ładnie firankę bierzemy linijkę i wyznaczamy punkty zaczepienia 2 "środkowych" żabek. Cały czas staramy się, o ile jest to możliwe, wyznaczać jedną środkową żabkę.

Zadanie

Twoim celem będzie wyznaczenie dla zadanej liczby żabek, ile razy będziemy zmuszeni wziąć do ręki linijkę.

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita d, $1 \le d \le 500000$. W kolejnych d wierszach znajdują się liczby żabek (n, $3 \le n < 2^{31}$), dla których należy wyznaczyć ilość pomiarów potrzebnych do równomiernego rozmieszczenia żabek.

Wyjście

W d liniach wyjścia należy podać dla każdej zadanej na wejściu liczby żabek, liczbę potrzebnych pomiarów linijką.

Przykład

Dla danych:

```
2
18
15
```

poprawną odpowiedzią jest:

```
1
6
```

Zagadkowy ciąg

ID:1027

Limit czasu: 2.50 s

Limit pamięci: 6144 kB

Michał zawsze bardzo lubił matematykę, więc - co zrozumiałe - był pojętnym uczniem. Rozwiązywał problemy matematyczne znacznie szybciej niż koledzy w klasie.

Pewnego dnia, na lekcji, gdy Michał zrobił już zadania z ćwiczeń, trochę ze znużenia, trochę z ciekawości zapisał w zeszycie ciąg: *1121231234123451234561234567...* - każda cyfra była w odzielnej kratce.

Zaintrygowany profesor szybko odgadł, w jaki sposób jego uczeń zbudował ciąg i chcąc go "zagiąć", zapytał, czy potrafi powiedzieć jaka cyfra znajduje się na zadanej pozycji. Michał odpowiedział dyplomatycznie, że problem wymaga głębszych przemyśleń i obiecał, że na następnej lekcji przedstawi rozwiązanie.

Po powrocie do domu zabrał się ostro do pracy - najpierw musi zobaczyć jak wyglądają wyniki dla "małych" liczb z przedziału $1..2^{31}-1$. Michał nie posiada komputera. Pomóż mu znaleźć rozwiązanie dla "małych" przypadków.

Wejście

Pierwsza liczba C, $1 \le C \le 100$, określa liczbę zestawów danych. W kolejnych C liniach znajdują się liczby N, $1 \le N < 2^{31}$, określające pozycję cyfry w ciągu.

Wyjście

Na wyjściu powinno znaleźć się C cyfr, każda w osobnej linii, równych odpowiednio cyfrze na N-tej pozycji w ciągu.

Przykład

Dla danych:

```
3
3
4
5
```

poprawnym rozwiązaniem jest:

```
2
1
2
```

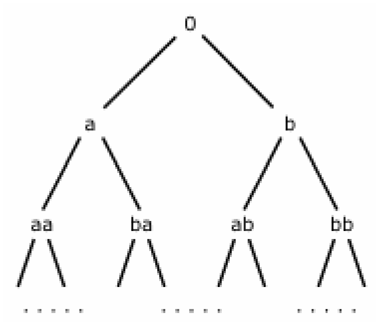
AB-drzewo

ID:1028

Limit czasu: 2.50 s

Limit pamięci: 16384 kB

Zdefiniujmy AB-drzewo jako pełne drzewo, którego węzłami będą słowa nad alfabetem {a, b}, korzeniem będzie słowo puste "0", a dla dowolnego słowa w, jego synami będą słowa: {xw: x należy do {a, b}}. Synami słowa w są 2 słowa powstałe w wyniku doklejenia jednej z liter {a, b} na początku słowa w. Wszystkie słowa w drzewie są różne. Wszelkie wątpliwości powinien wyjaśnić poniższy schemat:



Pełnym AB-drzewem o wysokości h nazywamy AB-drzewo w którym wszystkie liście są słowami o długości h. Z pełnego AB-drzewa o zadanej wysokości (równej długości słowa znajdującego się w liściu), będziemy usuwać pewne słowa, wraz z potomkami (poddrzewami). Twoim zadaniem będzie wyznaczenie liczby słów które pozostaną w drzewie po usunięciu zadanych słów wraz z potomkami.

Wejście

W pierwszym wierszu wejścia znajduje się liczba naturalna C, $1 \leq C \leq 10$, oznaczająca liczbę zestawów danych. W kolejnych wierszach znajdują się zestawy danych. W pierwszym wierszu każdego zestawu danych znajduje się liczba naturalna H, $1 \leq H \leq 30$ - jest to wysokość AB-drzewa. W drugim wierszu każdego zestawu danych znajduje się liczba naturalna N, $0 \leq N \leq 50000$, oznaczająca ilość słów które chcemy usunąć z drzewa wraz z ich poddrzewami. W kolejnych N wierszach znajdują się słowa które będziemy usuwać. Wszystkie słowa znajdują się w pełnym AB-drzewie o wysokości H, i nie ma wśród nich słowa pustego (korzenia).

Wyjście

W kolejnych liniach wyjścia powinny znaleźć się liczby słów które pozostały w drzewach z kolejnych zestawów danych.

Przykład

Dla danych:

```
1
2
```

```
3
b
ab
aa
```

poprawną odpowiedzią jest:

```
3
```

Domino

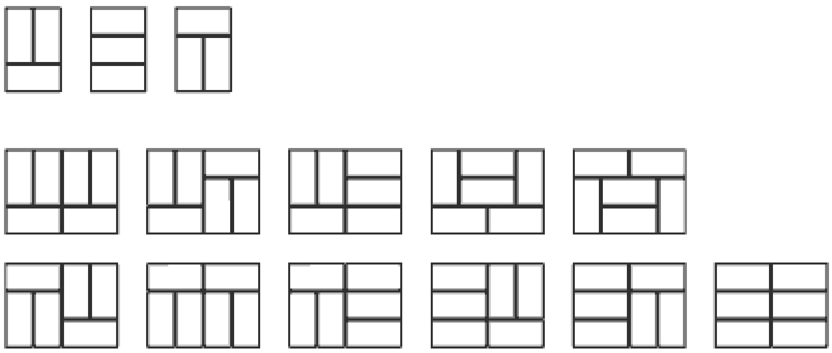
ID:1029

Limit czasu: 1.50 s

Limit pamięci: 4096 kB

Masz do dyspozycji nieograniczoną ilość kamieni domina, z których każdy ma wymiary 2x1. Twoim zadaniem będzie obliczenie, na ile różnych sposobów można za pomocą nierozróżnialnych kamieni domina pokryć prostokąt o wymiarach 3x*N*.

Na rysunku pokazano wszystkie sposoby na jakie można pokryć dominem prostokąty 3x2 i 3x4. Jest ich odpowiednio 3 i 11.



Rys. Wszystkie sposoby pokrycia prostokątów o wymiarach 3x2 i 3x4 kamieniami domina.

Wejście

W pierwszym wierszu wejścia znajduje się liczba naturalna C, $1 \le C \le 2000$, oznaczająca liczbę zestawów danych. Każdy zestaw danych składa się z jednego wiersza, zawierającego liczbę naturalną N, $1 \le N \le 100000$.

Wyjście

Dla każdego zestawu danych należy podać jeden wiersz wyniku zawierający liczbę $P \bmod 10^6$, gdzie P jest liczbą sposobów na jakie można pokryć prostokąt o wymiarach 3x*N*.

Przykład

Dla danych:

```
2
2
4
```

prawidłową odpowiedzią jest:

```
3
11
```

Sznurki

ID:1030

Limit czasu: 1.50 s

Limit pamięci: 4096 kB

Dwaj koledzy grają w grę polegającą na dowiązywaniu kolejnych kawałków sznurka do siebie, aż do momentu, gdy taka konstrukcja urwie się pod własnym ciężarem. Pierwszy sznurek dowiązywany jest przez gracza rozpoczynającego grę do specjalnego haczyka o nieskończonej wytrzymałości, a kolejne sznurki dowiązywane na zmianę przez obu graczy, zawsze do sznurka który był dowiązywany przez poprzednika.

Gracze mają do dyspozycji wiele rodzajów sznurków, o różnej wadze i różnej wytrzymałości. Wytrzymałość sznurka jest wprost proporcjonalna do jego wagi, a współczynnik wytrzymałości taki sam dla wszystkich wag sznurków. Zapasy sznurków każdej wagi są nieskończone. Gracze na zmianę dowiązują do sznurka położonego najniżej sznurek o wybranej wadze (tym samym i wytrzymałości), a przegrywa ten, po czyim ruchu któryś ze sznurków się zerwie.

Współczynnik wytrzymałości określa krotność ciężaru jaki wytrzymuje sznurek w odniesieniu do jego własnej wagi. Sznurek musi udźwignąć także własny ciężar. Sznurek zrywa się, gdy waga sznurków przywiązanych niżej od niego, wraz z jego wagą, przekracza wagę sznurka pomnożoną przez współczynik wytrzymałości.

Twoim zadaniem jest stwierdzenie, dla zadanego zbioru różnych wag dostępnych w grze sznurków oraz współczynnika wytrzymałości, czy gracz który rozpoczyna grę wybierając pierwszy sznurek, ma strategię wygrywającą. To znaczy, czy może tak wybrać pierwszy i następne sznurki, aby niezależnie od wyborów przeciwnika wygrać grę. Gracze na zmianę nie tylko dowiązują wybrane sznurki, ale także dokonują wyborów, znając wcześniejsze ruchy przeciwnika.

Wejście

W pierwszym wierszu wejścia znajduje się liczba $C, 1 \leq C \leq 100$, oznaczająca ilość zestawów danych. W kolejnych wierszach znajdują się zestawy danych. W pierwszym wierszu każdego zestawu danych znajduje się liczba $N, 1 \leq N < 100$ - jest to liczba różnych wag sznurków. W kolejnym wierszu znajduje się N liczb całkowitych dodatnich, mniejszych od 1000, posortowanych rosnąco, oddzielonych pojedynczymi spacjami. Są to wagi sznurków dostępnych w grze. W ostatnim wierszu każdego zestawu znajduje się liczba naturalna - współczynnik wytrzymałości - WW, $1 \leq WW \leq 100$, WW=Wytrzymałość/Waga, dla każdego sznurka.

Wyjście

Dla każdego zestawu danych należy wydać na standardowe wyjście linię zawierającą słowo "tak", jeśli gracz rozpoczynający grę ma strategię wygrywającą, a słowo "nie" w przeciwnym wypadku.

Przykład:

Dla danych:

```
4
3
```

```
1 2 3
4
4
1 2 4 10
12
5
90 91 92 93 999
100
4
450 900 901 902
4
```

poprawną odpowiedzią jest:

```
tak
nie
tak
nie
```

Prehistoryczny komputer

ID:1031

Limit czasu: 1.00 s

Limit pamięci: 6144 kB

Grupa archeologów z Bitlandii zorganizowała wyprawę naukową na Półwysep Bajtocki w celu przeprowadzenia wykopalisk. Podejrzewali znaleźć tam szczątki pradawnej cywilizacji Andorxorów, znanej powszechnie z posiadania wysokiego poziomu maszyn liczących. Już drugiego dnia wykopalisk wydawało się, że uczeni odnieśli sukces: udało się im wydobyć dziwne urządzenie, które z wyglądu przypominało prehistoryczny komputer! Szczęśliwi badacze byli przekonani, że teraz będą mogli rozszyfrować wszystkie zagadki Andorxorów i zgłębić wszystkie tajniki ich przeogromnej wiedzy. Jednak zadanie to okazało się dużo trudniejsze niż mogli kiedykolwiek przypuszczać. Po uruchomieniu maszyny pojawił się napis:

Witaj w świecie Andorxorów!

$f_0 = 0,$

$f_n = n^3 * (f_{n-1}+1) + n^2, n > 0$

$n = 54128$

$f_n \bmod 3331 = ???$

... a po dwóch sekundach maszyna wyłączyła się. Po ponownym uruchomieniu pojawiał się analogiczny napis, lecz różniący się od poprzedniego zadaną wartością *n*. Archeolodzy zaczęli wpisywać losowe liczby w odpowiedzi na pytanie zadane przez maszynę myśląc, że uda im się ją oszukać. Jednak nic z tego nie wyszło - błędna odpowiedź powodowała ponowne uruchomienie "komputera", to zaś generowało kolejne zapytanie, za każdym razem dla innego *n*.

Uczonym nie pozostało nic innego, tylko mozolnie obliczać wartości f_n *mod* 3331 dla kolejnych *n* tak, aby móc odpowiednie wyniki wprowadzić do maszyny przed upływem dwóch sekund... Jest to jednak dla nich ogromnie trudne zadanie...

Zadanie

Pomóż archeologom uruchomić prehistoryczną maszynę.

Wejście

W pierwszej linijce wejścia znajduje się liczba C, $1 \le C \le 5$, oznaczająca liczbę zestawów danych. W kolejnych C wierszach wejścia znajdują się zestawy danych. Każdy zestaw składa się z liczby naturalnej n, $0 \le n \le 100000000$.

Wyjście

Dla każdego zestawu danych, dla liczby n, na wyjściu powinna znaleźć się wartość f_n *mod* 3331, będąca hasłem do uruchomienia urządzenia.

Przykład

Dla danych wejściowych:

```
3
2
```

```
10
225
```

prawidłowym rozwiązaniem jest:

```
28
254
959
```

Mikołaj

ID:1032

Limit czasu: 1.00 s

Limit pamięci: 4096 kB

Małgosia dostała na Mikołaja klocki z wypisanymi literami alfabetu. Układanie słów z klocków było pasjonującą rozrywką. W trakcie zabawy odkryła, że niektóre słowa czytane od lewej do prawej i odwrotnie wyglądają tak samo. Opowiedziała tacie o swoich odkryciach. Ojciec wyjaśnił córce, że z dowolnego słowa można usunąć pewną liczbę liter (nawet równą zeru) i otrzymać niepuste słowo, które wygląda tak samo, gdy jest czytane od prawej do lewej i odwrotnie. Małgosia, po chwili zastanowienia, spytała, na ile sposobów da się to zrobić. Zafrasowany rodzic, jak przystało na programistę firmy MIRACLE, postanowił napisać program, który rozwiąże ten problem. Niestety, z powodu nawału pracy nad modułem sortującym nowej bazy danych zadanie to zlecił Tobie. Dodał jeszcze, że słowo musi zawierać co najmniej jedną literę i dwa sposoby usuwania liter są różne, jeśli usuwamy litery na różnych pozycjach (nie jest ważna kolejność usuwania). Np. ze słowa *AAA* możemy usunąć literę na pozycji 1, 2 lub 3 (trzy różne sposoby) i otrzymamy trzy słowa *AA* (**AA, A*A, AA**).

Wejście

W pierwszym wierszu wejścia znajduje się liczba C, $1 \leq C \leq 100$, oznaczająca liczbę zestawów danych. W kolejnych C liniach znajdują sie zestawy danych. Każdy zestaw danych to opis słowa - składa się z dużych liter alfabetu angielskiego, o długości L, $1 \leq L \leq 50$.

Wyjście

Na wyjściu, dla każdego słowa wejściowego, trzeba podać liczbę sposobów usuwania liter, tak by otrzymane słowa czytane od lewej do prawej, wyglądały tak samo, jak czytane od prawej do lewej.

Przykład

Dla danych:

```
4
X
XX
ALA
AAA
```

poprawną odpowiedzią jest:

```
1
3
5
7
```

Obliczenia równoległe

ID:1033

Limit czasu: 1.50 s

Limit pamięci: 20480 kB

Profesor Przemysław pasjonuje się obliczeniami równoległymi. Wykorzystuje sieć komputerową uczelni i nocami prowadzi eksperymenty przy użyciu maszyn w pracowniach. Obliczenia te przebiegają najczęściej według następującego schematu:

- Najpierw program wykonujący obliczenia jest uruchamiany na jednym z komputerów. (Nazwiemy go komputerem głównym).
- Uruchomiony program dzieli dane na małe fragmenty i rozsyła je do wszystkich komputerów w sieci. Każdy komputer w sieci otrzymuje inną, niepodzielną i przesyłaną w pojedynczym pakiecie porcję danych.
- Komputery wykonują obliczenia dla danych które otrzymały.
- Komputer główny czeka na wyniki od wszystkich komputerów, a następnie po odebraniu wszystkich wyników częściowych, wykonuje jakieś czynności związane z obliczeniem na ich podstawie wyniku końcowego.

Nie bez znaczenia jest wybór komputera na którym będą wykonywane obliczenia. Nie wszystkie komputery w sieci są ze sobą bezpośrednio połączone. (Zawsze istnieje jakaś droga pomiędzy komputerami, ale nie zawsze jest to połączenie bezpośrednie). Komputery komunikując się między sobą korzystają więc czasami z komputerów pośrednich, na zasadach podobnych do tych jakie panują we współczesnych sieciach komputerowych. Jeśli komputer otrzymuje porcję danych do obliczeń, która nie jest do niego adresowana, przekazuje ją dalej, tak aby dotarła do adresata najkrótszą drogą. Komputer rozsyłający dane również wysyła je w kierunku który gwarantuje przekazanie danych adresatowi najkrótszą drogą. Oczywiste jest, że wybierając komputer który ma służyć jako główny komputer wykonujący obliczenia, trzeba zminimalizować ilość pracy jaka zostanie wykonana podczas rozsyłania danych i zbierania wyników cząstkowych. To będzie właśnie Twoje zadanie.

Sieć komputerowa na uczelni ma strukturę drzewiastą. To znaczy, że pomiędzy dwoma komputerami istnieje zawsze dokładnie jedna droga. Znaczy to też, że można tę sieć opisać przypisując każdemu komputerowi (za wyjątkiem jednego) komputer dla niego nadrzędny (tak jak w drzewie każdy węzeł za wyjątkiem korzenia ma jednego ojca). Komputery zostały ponumerowane kolejnymi liczbami naturalnymi począwszy od 1, przy czym komputer o numerze 1 nie ma komputera nadrzędnego - jest to główny router uczelni. Profesor może wybrać dowolny komputer do prowadzenia obliczeń, ale nie ma bezpośredniego dostępu do routera, gdyż znajduje się on za drzwiami z zamkiem szyfrowym, a Profesor nie zna kodu wejściowego. Komputer o numerze 1 tak jak wszystkie komputery prowadzi więc obliczenia cząstkowe, ale nie może zostać wybrany jako komputer główny.

Twoim zadaniem będzie wyznaczenie dla zadanej sieci komputerowej, komputera przy wyborze którego, suma pracy wykonywanej przez WSZYSTKIE komputery sieci podczas rozsyłania danych będzie najmniejsza. Jeśli będzie kilka takich komputerów, wybieramy ten o najmniejszym numerze. Poprosimy Cię też o podanie sumy pracy wykonywanej przez wszystkie komputery podczas rozsyłania danych, przy założeniu, że dokonano optymalnego wyboru. Przyjmujemy, że jednostką pracy wykonywanej przez komputer jest operacja wysłania lub przekazania dalej pakietu danych. Przypominamy, że nie można wybrać komputera o numerze 1.

Wejście

W pierwszym wierszu wejścia znajduje się liczba C, $1 \le C \le 30$, oznaczająca ilość zestawów danych. W kolejnych wierszach znajdują się zestawy danych. W pierwszym wierszu każdego zestawu danych znajduje się liczba N, $2 \le N \le 100000$ - jest to ilość komputerów w uczelnianej sieci. W kolejnych N-1 wierszach znajdują się numery komputerów nadrzędnych kolejnych komputerów, od 2 do N, po jednym w każdym wierszu. (W terminologii "drzewowej" są to numery ojców w drzewie.)

Wyjście

Dla każdego zestawu danych należy wypisać 1 wiersz wyniku, zawierający 2 liczby naturalne oddzielone spacją. Pierwsza z nich to numer komputera wybranego jako komputer główny, a druga to suma pracy wykonanej przez wszystkie komputery podczas rozsyłania danych od komputera głównego do pozostałych komputerów, przy założeniu że komputer główny wybrano optymalnie.

Przykład

Dla danych:

```
2
7
1
1
2
2
3
3
9
9
2
9
4
4
4
4
1
```

poprawną odpowiedzią jest:

```
2 11
4 12
```

Binarne Bingo

ID:1034

Limit czasu: 1.50 s

Limit pamięci: 4096 kB

Zenek prowadzi niewielkie kasyno. Główną atrakcją w jego kasynie, jest "Binarne Bingo dla dwojga" - gra którą sam wymyślił (przynajmniej tak mu się wydaje), a w której dwóch graczy oczekuje na pewien ciąg wyników rzutów monetą.

Dokładniej: Każdy z graczy posiada wzorzec składający się z liter "**O**", "**R**", oznaczających odpowiednio orła i reszkę. Krupier rzuca monetą zapisując kolejne wyniki, a pierwsze wystąpienie wzorca oznacza wygraną posiadacza wzorca. Wystąpienie wzorca o długości L następuje, gdy ostatnie L zapisanych przez krupiera wyników, pokrywa się ze wzorcem. Czasami może zdarzyć się remis - gdy obaj gracze w tym samym momencie stwierdzą uzgodnienie wzorca. Wtedy gra jest powtarzana od początku.

Zenek chciałby zalegalizować swoje kasyno. Poszedł więc do odpowiedniego urzędu, i otrzymał do wypełnienia wiele formularzy. Okazało się, że aby zarejestrować "Binarne Bingo ..." musi podać w formularzu, dla każdej zadanej pary wzorców, który wzorzec ma większe szanse wygranej. Cała atrakcja gry Zenka, wynikała przecież z tego, że gracze mogą wybierać wzorce z ogromnej liczby dostępnych wzorców. Zenek nie jest w stanie poradzić sobie z wymaganiami urzędników i oczekuje Twojej pomocy. Pomóż Zenkowi zalegalizować kasyno.

Wejście

W pierwszym wierszu wejścia znajduje się liczba naturalna C, $1 \le C \le 200$, oznaczająca liczbę zestawów danych. W kolejnych wierszach znajdują się zestawy danych, po jednym zestawie w wierszu. Zestaw danych składa się z 2 niepustych wzorców, wybranych przez obu graczy, oddzielonych pojedynczą spacją. Wzorzec jest ciągiem znaków, którego elementami mogą być wyłącznie duże litery "**O**" lub "**R**". Długość wzorca nie przekracza 30 znaków.

Wyjście

W C wierszach wyjścia należy umieścić odpowiedzi dla poszczególnych zestawów danych, przy czym odpowiedzią jest dokładnie jedna liczba:

- 1 - jeśli wzorzec pierwszy daje większe szanse na wygraną,
- 2 - jeśli wzorzec drugi daje większe szanse na wygraną,
- 0 - jeśli oba wzorce dają takie same szanse na wygraną.

Przykład

Dla danych:

```
6
OORO OROO
OROO ROOO
OORO ROO
ROR ORO
ORO RO
```

```
RO RORO
```

poprawną odpowiedzią jest:

```
1
1
2
0
2
1
```

Drogi

ID:1035

Limit czasu: 1.50 s

Limit pamięci: 6144 kB

Zarząd znanej na rynku firmy informatycznej MIRACLE, tworzącej specjalistyczne oprogramowanie bazodanowe, chce pomóc swoim programistom w pokonywaniu drogi z ich mieszkań do pracy. Po wstępnej analizie problemu Zarząd dokonał niecodziennego odkrycia: okazało się, że domy programistów leżą na okręgu, którego środkiem jest siedziba firmy!

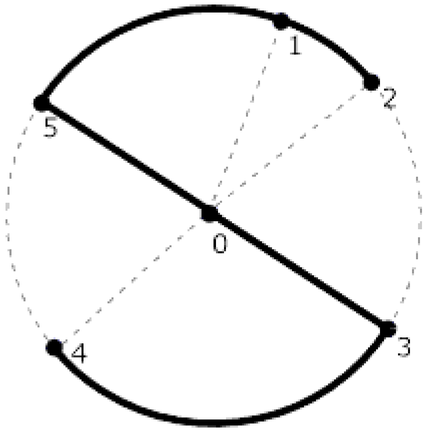
Firma chce wybudować sieć dróg w taki sposób, aby każdy programista, korzystając z nowych połączeń, mógł dojechać do jej siedziby.

Przepisy jednoznacznie określają warunki jakie powinna spełniać droga: jedna droga może łączyć dwa budynki i nie może przebiegać przez żaden inny budynek. Dodatkowo, jedna droga może albo łączyć bezpośrednio mieszkanie programisty z siedzibą firmy (inaczej mówiąc: łączyć środek okręgu - siedzibę firmy - z punktem leżącym na okręgu - mieszkaniem programisty) albo łączyć dwa sąsiadujące ze sobą mieszkania programistów (czyli: należeć do okręgu). Inne sposoby przeprowadzania dróg są sprzeczne z przepisami.

Zarząd firmy akceptuje wyłącznie takie projekty budowy, w których liczba niezbędnych dróg jest najmniejsza.

Zadanie

Twoim zadaniem jest wyznaczenie liczby wszystkich możliwych projektów budowy dróg, spełniających wymagania Zarządu firmy MIRACLE.



Rys. Przykładowy projekt budowy dróg (0 - siedziba firmy, 1-5 - mieszkania programistów).

Wejście

W pierwszej linijce wejścia znajduje się liczba naturalna C, $1 \le C \le 100$, oznaczająca liczbę zestawów danych. W kolejnych C wierszach wejścia znajdują się zestawy danych. Każdy zestaw składa się z liczby naturalnej n, $3 \le n \le 1000$, oznaczającej liczbę programistów zatrudnionych w firmie.

Wyjście

Dla każdego zestawu danych, dla liczby n, na wyjściu powinna znaleźć się liczba naturalna, będąca liczbą możliwych projektów budowy dróg spełniających wymagania Zarządu firmy MIRACLE.

Przykład

Dla danych wejściowych:

```
2
3
5
```

prawidłowym rozwiązaniem jest:

```
16
121
```

Plaster miodu

ID:1036

Limit czasu: 2.00 s

Limit pamięci: 16384 kB

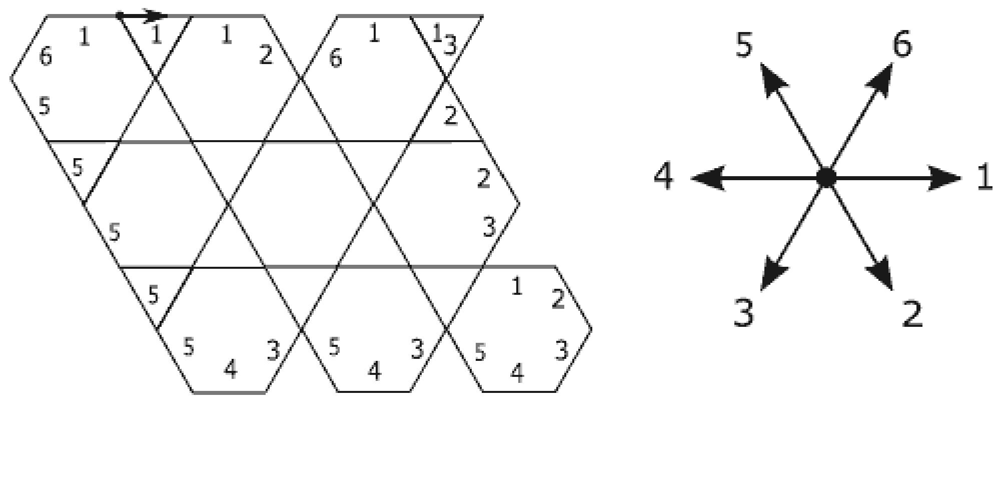
Pszczoły, jak wiadomo, budują plastry miodu z przylegających do siebie sześciokątów foremnych. Jednak, na jednej z pasiek, pszczoły zbudowały plaster w inny sposób. Plaster ma następującą budowę:

- sześciokąty foremne stykają się ze sobą tylko wierzchołkami,
- przestrzeń między szesciokątami wypełniają trójkąty równoboczne,
- trójkąty i sześciakąty stykają się tylko wzdłuż swoich krawędzi,
- każdy trójkąt styka się krawędzią przynajmniej z jednym sześciokątem.

Zgodnie z najnowszymi unijnymi przepisami, miody wytwarzane na plastrach składających się z innych figur niż sześciokąty, są objęte wyższą stawką podatku VAT.

Na szczęście pszczoła pilnująca poprawności plastra zorientowała się, że coś jest nie tak. Pszczoły zastanawiają się, czy opłaca się przekształcać strukturę plastra, tak aby składał się z samych sześciokątów. Sześciokąty foremne będą bezproblemowo przenoszone w inne miejsca plastra, problemem są natomiast trójkąty. Podczas naprawy plastra, miód zawarty w trójkątach jest spisany na straty. Pszczoły nie wiedzą czy zyski związane z objęciem miodu niższą stawką podatku VAT zrekompensują straty miodu spowodowane naprawą struktury. Pomóż pszczołom stwierdzić, czy opłaca się dostosowywać plaster do unijnych norm.

Pszczoły dysponują opisem kształtu plastra otrzymanym poprzez obejście plastra na około. Opis składa się z ciągu kierunków wzdłuż których poruszano się obchodząc plaster po obwodzie. Pszczoły budują plaster w taki sposób, że nigdy nie ma w nim dziur, dzięki czemu taki opis kształtu plastra jest w pełni wystarczający i od dawna stosowany przez pszczoły. Obwód plastra jest łamaną zamkniętą, a to znaczy, że mogą istnieć punkty do których pszczoła dotrze więcej niż jeden raz, zanim obejdzie cały plaster. Dla zadanej zamkniętej trasy po której pszczoła obeszła plaster na około, oblicz liczbę zawartych wewnątrz plastra trójkątów.



Rys. Struktura miodu którą stworzyły pszczoły wraz z przykładową trasą przejścia.

Wejście

W pierwszym wierszu wejścia znajduję się liczba całkowita C, $1 \leq C \leq 100$, oznaczająca liczbę zestawów danych testowych. W kolejnych wierszach znajdują się zestawy danych testowych. W pierwszym i jedynym wierszu każdego zestawu danych znajduje się ciąg cyfr z zakresu 1..6 opisujący jednoznacznie zamkniętą trasę, którą musiała pokonać pszczoła obchodząc nieprawidłowy plaster miodu. Pomiędzy cyframi znajdującymi się w jednym wierszu nie występują inne znaki. Każdy wiersz wejścia zawiera co najwyżej 100000 cyfr.

Wyjście

W C wierszach wyjścia należy podać wyznaczoną dla każdego zestawu danych liczbę trójkątów równobocznych (tylko tych najmniejszych, nie zawierających wewnątrz innych figur) zawartych w plastrze miodu zbudowanym przez pszczoły z tej pasieki.

Przykład

Dla danych:

```
2
112611322312345345345555561
123423456156
```

poprawną odpowiedzią jest:

```
13
0
```

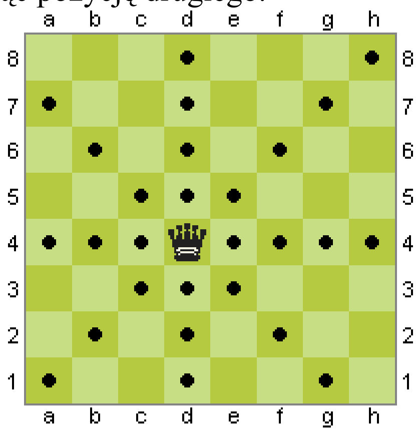
Hetmani

ID:1037

Limit czasu: 3.00 s

Limit pamięci: 16384 kB

Małgosia dostała w prezencie urodzinowym szachy. Nigdy wcześniej nie grała w tę grę i dopiero uczy się jej zasad. Naukę rozpoczęła od poznania ruchów figury, która najbardziej jej się spodobała, czyli hetmana. Hetman na szachownicy może poruszać się o dowolną ilość pól w dowolnym kierunku (do przodu, do tyłu, na boki i na ukos). Mówimy, że dwa hetmany szachują się, jeśli jeden z nich może w jednym ruchu zająć pozycję drugiego.



Rys. Dozwolone ruchy hetmana na przykładowej szachownicy.

Zadanie

Małgosia zastanawia się, ile jest różnych ustawień k hetmanów na szachownicy o wymiarach nxn pól, tak aby żadne dwa z nich nie szachowały się. Pomóż Małgosi rozwiązać ten problem.

Wejście

W pierwszym wierszu dana jest liczba naturalna d oznaczająca ilość zestawów danych, $1 \le d \le 10$. W następnych wierszach podane są kolejne zestawy danych. Każdy zestaw składa się z jednego wiersza, w którym znajdują się, oddzielone pojedynczą spacją, liczby naturalne n oraz k, $1 \le n \le 12$, $1 \le k \le n^2$, oznaczające odpowiednio: "wymiar" szachownicy oraz ilość hetmanów.

Wyjście

W d wierszach wyjścia należy podać wyznaczoną dla każdego zestawu liczbę różnych ustawień k hetmanów na szachownicy nxn tak, aby żadne dwa z nich nie szachowały się.

Przykład

Dla danych:

```
2
3 2
8 8
```

prawidłową odpowiedzią jest: 8

```
92
```

Robot

ID:1038

Limit czasu: 1.00 s

Limit pamięci: 8192 kB

W pewnym nowoczesnym mieście zbudowano ronda według nowej technologii. Każde rondo ma n lamp. Przed zmierzchem wypuszczany jest robot, którego zadaniem jest włączenie wszystkich lamp. Jednak programista dokonał pewnych błędów w programie robotów i wyprodukowano serię n robotów, które zamiast podchodzić do wszystkich lamp podchodziły tylko do k pierwszych (k - numer seryjny robota). Na dodatek, po podejściu do lampy robot przełącza wyłącznik (jeżeli był włączony to wyłącza, jeśli wyłączony to włącza). Następnie roboty podchodzą do k następnych lamp i tak w kółko, aż nie zostanie im wydany rozkaz zakończenia prac, który jest realizowany po zakończeniu przełączania aktualnych k lamp. Projektanci ronda załamali ręce. Jednak zauważyli, że część robotów jest w stanie zapalić wszystkie lampy (przykładowo dla n=4 wszystkie roboty są w stanie tego dokonać; dla n=3 tylko roboty o numerze seryjnym 1 i 3). Wszyscy wielce uradowaniu już zabrali się do selekcji odpowiednich robotów, gdy Urząd Miasta i Gminy zarządził wielkie oszczędności - na każdym rondzie może palić się tylko jedna lampa. To doszczętnie dobiło zespół. Jednak Ty, młody, inteligentny informatyk wiesz, że niektóre roboty potrafią tego dokonać. Dlatego też postanowiłeś napisać program, który policzy ile z serii n robotów mogłoby obsłużyć rondo o n lampach.

Wejście

W pierwszej linii znajduje się liczba naturalna D, $1 \leq D \leq 100$, oznaczająca liczbę zestawów danych. Każdy zestaw składa się z jednej linii zawierającej liczbę całkowitą n, $1 \leq n \leq 10^9$.

Wyjście

Dla każdego zestawu danych w osobnej linii wypisz liczbę robotów, które mogą obsłużyć rondo o n lampach.

Przykład

Dla wejścia:

```
2
4
6
```

poprawną odpowiedzią jest:

```
2
2
```

Wojna

ID:1039

Limit czasu: 1.00 s

Limit pamięci: 4096 kB

Pewnego słonecznego dnia na beztroskie królestwo Bajtocji najechał bezwzględny król Małych i Miękkich. Wstrząśnięci mieszkańcy po pierwszym szoku wywołanym tak nagłą napaścią podjęli próby obrony. Jednak pomimo zaciętego oporu, jaki stawiali, duża część ich wojsk dostała się do niewoli. Wśród nich znalazł się także ukochany nasz Linus T. Jednak on, w przeciwieństwie do reszty współwięźniów, postanowił uciec z niewoli. Naprzeciw wszystkim przeciwnościom losu powiodło mu się. Teraz, będąc na wolności, zaczął się martwić o pozostawiony dom i rodzinę. Dlatego też pierwsze co zrobił, to udał się do najbliższego pubu i zaczął wypytywać jak przebiegają działania wojenne... Dowiedział się, że bardzo duży obszar jest pod okupacją wroga. Wówczas zaczął się zastanawiać, czy jego dom też się tam znajduje. Twoim zadaniem jest pomóc Linusowi T. stwierdzić czy jego dom znajduje się na terenach okupowanych czy nie.

Teren okupowany jest opisany łamaną zamkniętą, której krawędzie nie przecinają się. Do terenu okupowanego zaliczają się również punkty znajdujące się na odcinkach łączących kolejne punkty łamanej.

Wejście

W pierwszej linii znajduje się liczba naturalna D, $1 \leq D \leq 100$, oznaczająca liczbę zestawów danych. W pierwszej linii każdego zestawu danych znajdują się współrzędne domu Linusa T. (współrzędne naturalne x, y; $0 \leq x, y \leq 10^9$. W drugiej linii znajduje się liczba K, $3 \leq K \leq 1000$, określająca liczbę punktów ograniczających teren okupowany przez wroga, a następnie K par współrzędnych kolejnych punktów (współrzędne naturalne kx, ky; $0 \leq kx, ky \leq 109$).

Wyjście

Dla każdego zestawu danych wypisz w osobnej linii pojedyncze słowo *TAK* lub *NIE* oznaczające, czy dom znajduje się na terenie okupowanym.

Przykład

Dla wejścia:

```
3
4 4
5 0 0 0 6 1 2 6 6 6 0
2 2
3 0 0 0 10 10 0
5 5
3 10 10 11 13 13 11
```

poprawnym wyjściem jest:

```
TAK
TAK
NIE
```

Włamanie

ID:1040

Limit czasu: 1.50 s

Limit pamięci: 4096 kB

Tomek, podejrzewając swojego szefa o malwersacje finansowe, postanowił włamać się do firmowego sejfu i udowodnić mu przestępstwo. Jednak taki plan posiada kilka słabych punktów. Jednym z nich jest włamanie się do sejfu, który posiada cyfrowy zamek. Będąc małym chłopcem Tomek często oglądał MacGyver'a, który w jednym z odcinków miał podobny problem. Na filmie udało mu się rozwiązać go w następujący sposób: za pomocą mąki stwierdził, które klawisze na zamku są tłuste (a więc używane) - dmuchnął mąką w klawiaturę i do tych tłustych klawiszy przykleiła się mąka ;) - a następnie, wiedząc jak długi jest kod, wystukał po kolei wszystkie możliwe kombinacje. Tomek postanowił skorzystać z podobnej metody. Jednak po kupieniu mąki i stwierdzeniu jakie klawisze są używane, zaczął się zastanawiać ile czasu zajmie mu dostanie się do środka. Twoim zadaniem jest oszacowanie tego czasu (przy założeniu że na jedną kombinację potrzebuje 1 sekundy).

Wejście

W pierwszej linii znajduje się liczba naturalna D, $1 \leq D \leq 100$, oznaczająca liczbę zestawów danych. Każdy zestaw składa się z jednej linii zawierającej dwie liczby całkowite: K, $1 \leq K \leq 10$, oraz N, $K \leq N \leq 24$, gdzie K oznacza liczbę różnych cyfr, zaś N długość kodu.

Wyjście

Dla każdego zestawu danych wypisz w osobnej linii czas jaki zajmie Tomkowi włamanie się do sejfu w formacie: dni:godziny:minuty:sekundy

Przykład

Dla wejścia:

```
2
3 4
4 4
```

poprawną odpowiedzią jest:

```
0:0:0:36
0:0:0:24
```

Dr Judym

ID:1041

Limit czasu: 1.50 s

Limit pamięci: 2048 kB

Pewnego dnia doktor Judym zauważył na elektrokardiografie, że elektrokardiogram jego pacjenta w pewnym przedziale czasowym, jest wykresem wielomianu. Znając własności wielomianów doktor wpadł na pomysł prognozowania przyszłego wyglądu krzywych EKG swoich pacjentów na podstawie wykonanych badań. Pomogłoby mu to w ocenie stanu zdrowia i zapobieganiu chorobom krążenia. Do wyznaczenia elektorokardiogramu niezbędna jest znajomość wartości pewnego wielomianu w danej jednostce czasu. Doktor nie potrafi napisać programu komputerowgo obliczjącego te wartości. Zna jedynie współczynniki wielomianu oraz punkty na osi czasu. Pomóż mu znaleźć krzywą EKG.

Wejście:

W pierwszym wierszu znajduje się liczba zestawów danych $0 < d \leq 100$. Każdy zestaw danych składa się z trzech wierszy: w pierwszym mamy daną chwilę t w której chcemy wyznaczyć wartość wielomianu, w drugim stopień wielomianu $0 \leq N \leq 100000$, zaś w trzecim $N + 1$ współczynników wielomianu oddzielonych spacjami poczynając od współczynnika przy najwyższej potędze. Na przykład, dla wielomianu $W(t) = a0 + a1 * t + a2 * t^2 + a3 * t^3$ mamy ciąg: *a3 a2 a1 a0*. Współczynniki wielomianu są liczbami rzeczywistymi - podanymi z dokładnoscią do 3 miejsc po przecinku.

Wyjście:

Na wyjściu w kolejnych d wierszach powinny znaleźć się szukane wartości - liczby rzeczywiste - zaokrąglone do 3 miejsc po przecinku. Elektrokardiogram powinien mieścić się na ekranie elektrokardiografu - można więc bezpiecznie założyć, że wyniki będą mieściły się w standardowym typie rzeczywistym.

Przykład:

Dla danych:

```
3
0.100
4
-0.700 3.000 0.000 8.000 -5.000
3.000
3
3.000 0.000 0.000 10000.000
128.000
2
1.000 -1.000 1.000
```

poprawnym rozwiązaniem jest:

-4.197

10081.000

16257.000

Lotki

ID:1042

Limit czasu: 1.50 s

Limit pamięci: 4096 kB

Na imprezie integracyjnej Zakładu Matematyki Stosowanej zorganizowano konkurs w rzucaniu lotkami. Nie byłoby w tym nic nadzwyczajnego, gdyby nie to, że matematycy to dziwni ludzie i zawsze muszą utrudnić sobie życie lub wymyśleć jakieś uogólnienie, którego później nikt nie rozumie. I tak było w tym przypadku - zamiast okrągłej tarczy użyto trójkąta Pascala i liczby pierwszej. Zadaniem zaś było dla wylosowanej wcześniej liczby pierwszej trafić w liczbę przez nią niepodzielną .

Twoim zadaniem jest sprawdzić, czy jest sens organizować konkurs. Dla danej liczby pierwszej *P* oraz wysokości trójkąta Pascala *H* wyznacz ile jest pól na planszy w które można trafić. Przykładowa tarcza użyta w konkursie wygląda tak:

```
H=0                                       1
H=1                                  1         1
H=2                             1         2         1
H=3                        1         3         3         1
...                        .................................

H=i                        1       i         k         i         1
H=i+1                 1       1+i       i+k       k+i       i+1        1
```

Wejście:

W pierszym wierszu znajduje się liczba zestawów danych $0 < d \leq 100$. Każdy zestaw składa się z dwóch liczb: liczby pierwszej *P*, $2 \leq P \leq 1000000$ i *H*, $0 \leq H \leq 10000$ oddzielonych spacją.

Wyjście:

Dla każdego zestawu danych powinieneś wypisać liczbę pól (liczb) na planszy (w trójkącie Pascala o wysokości *H*), które nie dzielą sie przez *P*.

Przykład:

Dla danych:

```
3
2 2
3 4
7 6
```

Rozwiązaniem jest:

```
5
12
28
```

Liczby wyważone

ID:1043

Limit czasu: 1.50 s

Limit pamięci: 2048 kB

Liczbą wyważoną nazwiemy dodatnią liczbę naturalną posiadającą tyle samo dzielników parzystych co nieparzystych. Twoim zadaniem będzie wyznaczenie dla zadanej dodatniej liczby N najmniejszej liczby wyważonej większej od N.

Wejście

W pierwszym wierszu wejścia znajduje się liczba C, określająca ilość zestawów danych, $1 \leq C \leq 100$. Każdy z C zestawów danych składa się z jednego wiersza zawierającego jedną dodatnią liczbę naturalną N, składającą się co najwyżej z 200 cyfr.

Wyjście

Dla każdego zestawu danych, w osobnych liniach wyjścia, należy wyznaczyć najmniejszą liczbę wyważoną większą od zadanej liczby N z każdego zestawu danych.

Przykład

Dla danych wejściowych:

```
2
1
2
```

poprawną odpowiedzią jest:

```
2
6
```

Piętnastka

ID:1044

Limit czasu: 2.00 s

Limit pamięci: 16384 kB

Gra potocznie zwana "Piętnastką" polega na układaniu 15 ponumerowanych kolejno klocków. Klocki umieszczone są w kwadracie o wymiarach 4x4, a więc pozostaje jedno puste pole. Klocków nie można wyjmować, ani wysuwać poza wyznaczony kwadrat, a ruchów można dokonywać tylko i wyłącznie za pomocą przesuwania w puste pole jednego z klocków sąsiadujących z nim w pionie lub poziomie.

Gracz wygrywa, gdy uda mu się ułożyć za pomocą dozwolonych ruchów klocki tak, aby numery klocków w rzędach były ułożone kolejno i rosnąco z lewej do prawej, w kolumnach rosnąco z góry na dół, oraz puste miejsce znajdowało się w prawym dolnym rogu planszy. Okazuje się, że nie każdy początkowy układ klocków umożliwia wygraną.

Zadanie

Twoim zadaniem będzie stwierdzenie, czy dla zadanego początkowego układu klocków, możliwe będzie w skończonej liczbie ruchów doprowadzenie do wygranej. Dodatkowo nie będziemy ograniczać się do klasycznej wersji gry, ale wprowadzimy dowolne rozmiary planszy, zawsze jednak będzie tylko 1 puste pole.

Fot. Klasyczna "Piętnastka".

Wejście

W pierwszym wierszu wejścia znajduje się liczba naturalna C, $1 \le C \le 10$, oznaczająca ilość zestawów danych. Każdy zestaw danych rozpoczyna się linią zawierającą 2 liczby naturalne W i K, $2 \le W, K \le 100$. Liczby te oznaczają odpowiednio liczbę wierszy oraz kolumn w układance. W każdym z kolejnych W wierszy zestawu danych znajduje się K liczb naturalnych z zakresu $0..W*K-1$. Są to numery klocków na poszczególnych pozycjach w momencie rozpoczęcia gry. Numer 0 oznacza puste miejsce za pomocą którego dokonujemy ruchów.

Wyjście

Dla każdego zestawu danych w osobnych liniach należy wypisać słowo "tak" jeśli gracz może doprowadzić do wygranej albo słowo "nie", jeśli jest to niemożliwe.

Przykład

Dla danych:

```
2
4 4
1 6 2 4
3 13 11 7
14 5 10 8
0 9 15 12
4 4
1 2 3 4
5 6 7 8
9 10 11 12
13 15 14 0
```

poprawną odpowiedzią jest:

```
tak
nie
```

Przeprowadzka

ID:1045

Limit czasu: 1.50 s

Limit pamięci: 2048 kB

Po sukcesie rynkowym, jakim było wyprodukowanie bazy danych Mir 13k, dla firmy Miracle nastał okres dynamicznego rozwoju. Dział programistów znacznie się powiększył i postanowiono przenieść go do nowej siedziby. Aby zminimalizować koszty związane ze zmianą miejsca pracy ustalono następujący plan przeprowadzki:

1) każdy programista musi spakować swoje rzeczy do pojemnika, wynieść go przed siedzibę firmy i wstawić na końcu szeregu pojemników innych pracowników (pojemniki będą tworzyły jeden szereg w kolejności wynoszenia)

2) wszystkie pojemniki w szeregu zostaną zważone i podzielone na *K* grup z zachowaniem kolejności wynoszenia

3) jedna ciężarówka przewiezie kolejno grupy pojemników do nowej siedziby wykonując dokładnie K kursów pomiędzy siedzibami Miracle

4) pojemniki będą rozpakowane w takiej kolejności, w jakiej były wynoszone przez pracowników

Ustalono z firmą transportową, że koszt przewiezienia rzeczy pracowników jest wprost proporcjonalny do ładowności ciężarówki (ustalona liczba kursów *K* nie ma wpływu na koszt transportu). Miracle chce przewieźć pojemniki możliwie najmniejszym kosztem. Pomóż jej podzielić pojemniki na *K* grup, bez zmiany kolejności, tak by sumaryczna waga przedmiotów w najcięższej grupie była najmniejsza. Liczba kursów *K* nie ma wpływu na koszt przewozu, więc może się nawet zdarzyć sytuacja, w której ciężarówka kursuje z pustym ładunkiem.

Wejście

W pierwszym wierszu znajdują się dwie liczby całkowite $1 \leq N \leq 1000$ i $1 \leq K \leq 100$ oznaczające odpowiednio ilość pojemników (*N)* i liczbę kursów (*K*). W drugim i ostatnim wierszu jest *N* liczb całkowitych dodatnich $1 \leq x_i \leq 1000$, określających wagi poszczególnych pojemników. Wagi są podane w kolejności wynoszenie ze starej siedziby.

Wyjście

Na wyjściu trzeba podać minimalną sumaryczną wagę najcięższej grupy pojemników, przy podziale *N* pojemników na *K* grup.

Przykład

Dla danych:

```
6 3
1 1 2 3 5 7
```

poprawną odpowiedzią jest:

```
7
```

Flamaster

ID:1046

Limit czasu: 1.00 s

Limit pamięci: 2048 kB

Kasia niedawno poznała wszystkie literki w szkole. Z wielką pasją potrafiła całe dnie spędzać na pisaniu długich słów swoim ulubionym flamastrem. Pisała i pisała "tasiemce" tak długo, aż flamaster wypisał się. Kasia posmutniała. Z trudem, ale udało jej się uprosić swoją mamę, aby kupiła jej nowy pisak. Musiała jednak obiecać, że tym razem będzie bardziej oszczędna przy jego używaniu żeby wystarczył na dłużej. Kasia zaczęła zastanawiać się w jaki sposób będzie mogła zrealizować obietnicę daną mamie.

Postanowiła, że aby zaoszczędzić wkład flamastra będzie wypisywała skróconą wersję wymyślanych wyrazów. Jeśli miała zamiar napisać więcej niż dwie takie same literki obok siebie w wyrazie, to teraz napisze literkę a następnie liczbę, określającą ilość wystąpień tej literki.

Zadanie

Twoim zadaniem jest dla zadanego wyrazu, który wymyśliła Kasia, podanie skróconej wersji tego wyrazu.

Wejście

W pierwszej linijce wejścia znajduje się liczba naturalna C, $1 \leq C \leq 50$, oznaczająca ilość zestawów danych. W kolejnych C wierszach wejścia znajdują się zestawy danych. Każdy zestaw składa się z niepustego wyrazu złożonego z samych dużych liter alfabetu amerykańskiego. Długość wyrazu nie przekracza 200 znaków.

Wyjście

Dla każdego zestawu danych, dla zadanego wyrazu, na wyjściu powinna znaleźć się jego skrócona wersja.

Przykład

Dla danych wejściowych:

```
4
OPSS
ABCDEF
ABBCCCDDDDEEEEEFGGHIIJKKKL
AAAAAAAAAABBBBBBBBBBBBBBBB
```

prawidłowym rozwiązaniem jest:

```
OPSS
ABCDEF
ABBC3D4E5FGGHIIJK3L
A10B16
```

OPS.

ID:1047

Limit czasu: 2.00 s

Limit pamięci: 8192 kB

Zespół ludzi tworzących system OPSS chce stworzyć specjalny język programowania OPS., który będzie pomagał im w generowaniu testów do zadań konkursowych. Jednak ze względu na brak czasu Tobie powierzają napisanie jego interpretera. Język ma być prosty i ma korzystać z notacji postfiksowej.

Interpreter powinien wykonywać program w kolejności występowania w nim poleceń (symboli). Poleceniem może być liczba całkowita bądź jedna z 4 operacji: 'O', 'P', 'S', '.'. Po napotkaniu liczby interpreter powinien umieścić ją na szczycie stosu. Interpreter po napotkaniu operacji:

- O (Odejmij) - powinien zdjąć ze stosu liczbę L1, następnie zdjąć drugą liczbę L2 i umieścić na szczycie stosu ich różnicę (L2-L1),
- P (Przemnóż) - powinien zdjąć dwie liczby ze szczytu stosu a następnie umieścić na stosie ich iloczyn,
- S (Sumuj) - powinien zdjąć dwie liczby ze stosu a następnie na stosie umieścić ich sumę,
- . (Koniec) - powinien zakończyć wykonywanie programu, zdejmować kolejno wszystkie liczby ze stosu i wypisać je na standardowe wyjście.

Wejście

W pierwszym wierszu znajduje się liczba C, oznaczająca ilość programów, $0 < C \leq 100$. W następnych C liniach dostajemy treści programów, w których polecenia oddzielone są pojedynczymi spacjami. Każdy program kończy się operacją '.'. Programy będą gwarantować, że wartość bezwzględna wyniku każdej operacji 'O', 'P', 'S' nie przekroczy 2^{31}-1. Każdy program może się składać maksymalnie z 500000 symboli.

Wyjście

Dla każdego programu w osobnej linijce wyjścia powinien znaleźć się wynik działania danego programu w języku OPS. ;-)

Przykład

Dla danych:

```
3
6502 68000 6502 68000 6502 O P S S 2005 4 2 .
11 22 33 44 O S P .
10 9 O -1 S 9 8 7 6 5 4 3 2 1 1 P .
```

poprawną odpowiedzią jest:

```
2 4 2005 399934498
121
1 2 3 4 5 6 7 8 9 0
```

Automat

ID:1049

Limit czasu: 2.00 s

Limit pamięci: 2048 kB

Bajtoccy inżynierowie postanowili zoptymalizować wydawanie reszty w automacie z napojami. Do tej pory automat działał poprawnie, lecz liczba monet w wydawanej kwocie nie była minimalna, tzn. zdarzało się, że kwota 99B (B to balony - waluta Bajtocji) wydawana była w 99 monetach po 1B. Prowadziło to do kłopotliwych sytuacji - niektórzy Bajtoci odchodząc od automatu mieli tak wypchane kieszenie monetami, że te wypadały na ziemię, czyniąc dziury w budżecie Bajtocji. W końcu Bajtoccy inżynierowie postanowili zmienić ten stan rzeczy i nieco zmodyfikować proces wydawania reszty. Zatrudnili do tego grupę utalentowanych, młodych programistów, wśród których znalazłeś się właśnie Ty!

Zadanie

Napisz program, który dla zadanej kwoty reszty , którą należy wypłacić, oraz dla znanego zestawu nominałów monet dostępnych w automacie, obliczy optymalną (minimalną) liczbę monet, jaką powinien wypłacić automat. Bajtocki automat posiada nieskończenie wiele monet o zadanym nominale.

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita C, $0 < C \leq 100$, określająca ilość zestawów danych. Każdy zestaw składa się z dwóch wierszy. Pierwszy z nich zawiera dwie liczby całkowite N i K oddzielone pojedynczą spacją, gdzie N to kwota reszty jaką trzeba wydać, $0 \leq N \leq 1000$, zaś K to liczba dostępnych nominałów monet, $0 < K \leq 100$. W drugim wierszu zestawu znajduje się K dostępnych nominałów monet, oddzielonych pojedynczymi spacjami. Każdy nominał określony jest przez dodatnią liczbę naturalną nie większą niż 1000. Zakładamy, że żaden nominał nie występuje więcej niż jeden raz w drugiej linii zestawu.

Wyjście

Dla każdego zestawu danych program powinien wypisać na wyjście minimalną możliwą liczbę monet jaką powinien wydać automat, aby wydana reszta równa była zadanej kwocie . Jeśli automat nie może wydać reszty (bo np. nie posiada monet, którymi mógłby wydać żądaną sumę) należy wypisać 0.

Przykład:

Dla danych wejściowych:

```
3
11 3
5 3 1
1 1
2
1000 10
1 3 5 7 10 20 50 100 200 300
```

poprawną odpowiedzią jest:

3

0

4

Obława

ID: 1050

Limit czasu: 1.50 s

Limit pamięci: 16384 kB

Bandyci obrabowali bank. Będą próbowali przedostać się w bezpieczne miejsce na drugim końcu miasta. Jednak okazało się, że ich samochód jest uszkodzony i są zmuszeni uciekać pieszo. Nie będą więc, rzecz jasna, uciekać ulicami, a poprzez liczne w mieście parki i tereny niezabudowane (dla ułatwienia wszystkie takie tereny nazywać będziemy parkami). Park jest obszarem miasta ograniczonym trzema lub więcej ulicami, nie zawierający ulic i niezabudowany. Bandyci mogą przemieścić się z parku do parku tylko wtedy, gdy mają one co najmniej jedną wspólną ulicę je ograniczającą lub oba sąsiadują ze wspólnym skrzyżowaniem (wtedy bandyci szybko przebiegają na drugi koniec ulicy lub skrzyżowania). Bandyci postanowili uciekać tylko i wyłącznie poprzez parki, wybierając jedną z możliwych tras, niekoniecznie najkrótszą aby nie ułatwiać pracy ścigającej ich policji.

Przed komendantem policji stoi poważne zadanie złapania bandytów. Wie, że będą uciekać pieszo poprzez sąsiadujące parki, ale nie wie dokładnie którędy. Bandyci znajdują się w parku o numerze 1 i chcą dostać się do parku o numerze p. W parkach o numerach 1 i p bandyci czują się bezpiecznie i do tych parków policja nie ma wstępu. Parki 1 i p na całe szczęście nie sąsiadują ze sobą, więc policja nie jest bez szans. Jedną z możliwości rozpatrywanych przez komendanta, jest zastawienie zasadzek na bandytów w parkach przez które mogą się przemieszczać. Nie może jednak dopuścić do tego, by bandyci się wymknęli. Komendant chce wiedzieć, w ilu co najmniej parkach należy umieścić policyjne patrole, aby bandyci nie mieli możliwości ucieczki. Zakładamy, że jeśli park jest patrolowany, i bandyci spróbują się przez niego przedostać, zostaną złapani. Pomóż komendantowi podjąć decyzję i wyznacz minimalną liczbę parków jaką należy patrolować, aby mieć gwarancję że bandyci zostaną złapani. Jeśli liczba ta będzie zbyt duża, może nie wystarczyć radiowozów i trzeba będzie szybko szukać innego planu, więc to od Ciebie w dużej mierze zależy sukces całej operacji. Do dzieła! Każda minuta ma znaczenie!

Wejście

W pierwszym wierszu wejścia znajduje się liczba C, 1<=C<=100, oznaczająca ilość zestawów danych. W kolejnych wierszach znajdują się zestawy danych. W pierwszym wierszu każdego zestawu danych znajdują się dwie liczby, n i m, 1<=n<m<=100000. Są to odpowiednio: liczba skrzyżowań i liczba ulic w mieście. Skrzyżowania są ponumerowane od 1 do n, kolejnymi liczbami naturalnymi. W kolejnych m wierszach znajdują się pary liczb oznaczające numery skrzyżowań które łączą kolejne ulice. Dwa skrzyżowania mogą być bezpośrednio połączone co najwyżej jedną ulicą, a żadne dwie ulice nie przecinają się. W kolejnym wierszu znajduje się liczba p, 3<=p<=300 oznaczająca liczbę parków. W kolejnych p wierszach znajdują się opisy parków. W każdym wierszu opisującym park znajdują się liczby naturalne, oddzielone pojedynczymi spacjami. Pierwszą z nich jest liczba skrzyżowań (a tym samym i ulic) sąsiadujących z parkiem, a następnie po spacji oddzielone spacjami numery tych skrzyżowań. Parki są ponumerowane od 1 do p i podawane w kolejności rosnących numerów. (Przypominamy, że bandyci znajdują się w parku 1 i chcą uciec do parku p).

Wyjście

W C wierszach wyjścia należy podać wyznaczoną dla każdego zestawu minimalną liczbę patroli niezbędną do zatrzymania bandytów przy założeniu, że wybrano plan działania polegający na patrolowaniu wybranych parków.

Przykład

Dla danych:

```
1
12 22
1 2
1 3
2 4
3 4
3 5
3 6
4 6
4 7
4 8
5 6
6 7
7 8
5 9
6 10
7 10
8 10
9 10
9 12
10 12
12 11
10 11
11 8
9
3 4 7 6
4 1 2 3 4
3 3 5 6
3 4 7 8
4 5 9 10 6
3 7 8 10
3 8 10 11
3 9 12 10
3 12 11 10
```

poprawną odpowiedzią jest:

```
3
```

Kosmiczne sygnały

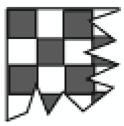
ID:1051

Limit czasu: 2.50 s

Limit pamięci: 8192 kB

Jesteś jednym z programistów agencji NASA. Pracujecie nad urządzeniem - specjalną tarczą, która ma służyć porozumiewaniu się z kosmitami za pomocą sygnałów świetlnych. Tarcza emitująca światło ma postać prostokąta, podzielonego na kwadraty o rozmiarach jednostkowych. Każdy element tarczy może być w danym momencie zapalony lub zgaszony. Naukowcy mają nadzieję, że odpowiedni dobór kombinacji zapalonych pól tarczy świetlnej pozwoli komunikować się z przybyszami z innych planet.

Programistyczny interfejs tarczy świetlnej jest jednak dość osobliwy - wyjaśnimy tylko to co konieczne, zasłaniając się tajemnicą wojskową. W momencie rozpoczynania tworzenia wiadomości, wszystkie pola są zgaszone. Wiadomość jest tworzona poprzez umieszczanie na tarczy prostokątnych szachownic, przy czym pola każdej szachownicy mają wymiary odpowiadające rozmiarom pól tarczy. Każda szachownica składa się z pól aktywnych i nieaktywnych, przeplatających się tak jak przeplatają się pola białe i czarne klasycznej szachownicy. W lewym górnym rogu każdej szachownicy znajduje się zawsze pole aktywne.



Rys. Fragment szachownicy z polami aktywnymi zaznaczonymi czarnym kolorem.

Każde pole szachownicy przylega dokładnie do jednego z pól tarczy. Pole tarczy zapala się w momencie, gdy suma pól aktywnych, które zostały na nie położone, jest nieparzysta.

Projekt jest już prawie gotowy, pozostało tylko opracowanie kontroli poprawności sygnałów. Tobie przypadło zadanie opracowania algorytmu obliczającego sumę wszystkich zapalonych pól tarczy świetlnej.

Wejście

W pierwszym wierszu wejścia znajduje się liczba zestawów danych C, 1<=C<=20. W pierwszym wierszu każdego zestawu danych znajduje się liczba N, 1<=N<=1000, oznaczająca liczbę szachownic użytych do konstrukcji wiadomości świetlnej. W każdym z kolejnych N wierszy zestawu znajdują się 4 liczby całkowite, x1,y1,x2,y2, -1000000000<=x1,y1,x2,y2<=1000000000. Określają one pozycję i rozmiary kolejnych szachownic tworzących wiadomość. Lewy górny róg szachownicy znajduje się na pozycji (x1,y1) a prawy dolny na pozycji (x2,y2). Układ współrzędnych jest tak dobrany, że x1<=x2 oraz y1<=y2.

Wyjście

Dla każdego zestawu danych należy wyznaczyć sumę zapalonych pól tarczy świetlnej.

Przykład

Dla danych:

```
5
2
0 0 3 3
2 2 5 5
2
0 0 3 3
3 3 6 6
2
0 0 3 3
0 1 3 4
2
0 0 3 3
3 1 6 4
2
1 2 5 5
0 0 3 3
```

poprawną odpowiedzią jest:

```
12
14
16
12
18
```

Cwany Lutek

ID:1052

Limit czasu: 1.00 s

Limit pamięci: 2048 kB

Doszedłeś do etapu na którym czeka na Ciebie Cwany Lutek. Aby przejść dalej musisz poprawnie odpowiedzieć na pytanie przez niego postawione. Test Lutka jest krótki i zawsze taki sam. Cwaniak rzuca dwie liczby N i K, a Ty musisz odpowiedzieć, czy liczba sposobów wskazania K przedmiotów ze zbioru wszystkich N przedmiotów (kolejność wskazywania nie ma znaczenia) jest liczbą parzystą czy nieparzystą.

Wejście

W pierwszym wierszu znajduje się liczba d, określająca ilość zestawów danych, $1 \le d \le 1000$. Każdy zestaw znajduje się w osobnej linii i zawiera dwie liczby całkowite N i K, $0 \le N, K \le 1000000000$, oddzielone pojedynczą spacją.

Wyjście

Dla każdego zestawu danych w oddzielnej linii wyjścia powinieneś wypisać jedną literę 'P' jeśli liczba sposobów jest liczbą parzystą lub 'N' jeśli jest liczbą nieparzystą.

Przykład

Dla wejścia:

```
3
100 2
7 7
19 9
```

poprawną odpowiedzią jest:

```
P
N
P
```

Chomiki Edka

ID:1053

Limit czasu: 1.50 s

Limit pamięci: 4096 kB

Edek zawsze lubił chomiki. Jeszcze w czasach szkolnych opiekował się tymi zwierzętami w pracowni biologicznej, a w domu hodował chomiki syryjskie. Po maturze wybrał studia na Wydziale Biologii. Właśnie zbliża się koniec studiów i oczywiście Edek (a teraz właściwie już Edward) wybrał temat pracy magisterskiej, który, jak zapewne już odgadliście, związany jest z hodowlą chomików. Niedawno Wydział otrzymał dwie pary chomików egzotycznej rasy, która dosyć wolno (jak na chomiki) się rozmnaża. Promotor zasugerował Edkowi, aby oszacował liczebność populacji nowej rasy po kilku latach.

Samice chomików nowej rasy rodzą, przy dobrych warunkach, jedną parę co miesiąc. Małe chomiki szybko dojrzewają i po miesiącu zdolne są już zostać rodzicami. Z dwóch par chomików, które otrzymał wydział po miesiącu urodziły się jednak, zapewne z powodu stresu związanego ze zmianą miejsca pobytu, tylko 2 młode, na szczęście była to para. W drugim miesiącu sytuacja sie powtórzyła, ale juz w nastepnych miesiącach wszystko potoczyło się normalnie czyli z trzech par "starych" urodziły się 3 pary młodych itd. Edek zaczął zliczać pary: chomików i na początku praca choć żmudna szła mu bez specjalnych trudności: pierwszy miesiąc - 3, drugi miesiąc - 4, trzeci - 7, po roku 521 par. Liczby rosły tak szybko, że Edek zaczął się obawiać czy przypadkiem chomiki nie będą mnożyć się szybciej niż on będzie dodawał!

Pomóż pracowitemu studentowi biologii oszacować liczbę chomików po wielu latach hodowli, zakładając, że nowa rasa chomików jest bardzo, bardzo długowieczna. Edek będzie zadowolony gdy podasz długość i 10 pierwszych cyfr tej liczby.

Wejście

W pierwszej linii wejścia znajduje się liczba D, określająca ilość zestawów danych, $1 \le D \le 1000$. W kolejnych wierszach wejścia znajdują się zestawy danych. Każdy z D zestawów danych składa się z wiersza zawierającego jedną dodatnią liczbę naturalną L, oznaczającą liczbę lat hodowli L, $1 \le L \le 5000$.

Wyjście

Dla każdego zestawu danych, w osobnej linii wyjścia, należy wypisać liczbę cyfr N, z których składa się poszukiwana przez Edka liczba chomików C a po spacji 10 początkowych cyfr liczby C. W przypadku gdy $N < 10$, należy wypisać dokładnie N cyfr.

Przykład

```
2
1
4
```

prawidłowym rozwiązaniem jest:

```
3 521
11 1739379600
```

Skoczek

ID:1054

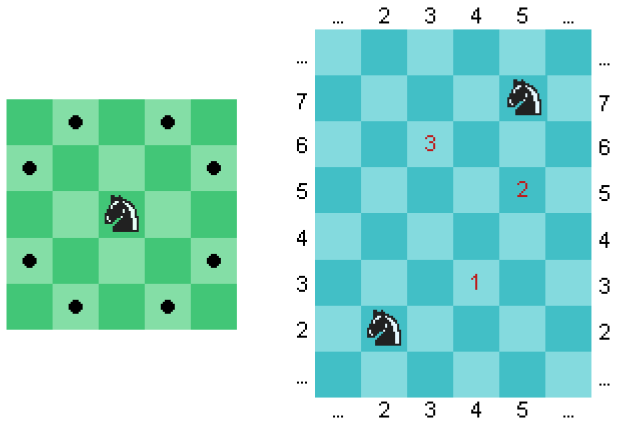
Limit czasu: 1.00 s

Limit pamięci: 6144 kB

Małgosia, jak wszyscy wiedzą, dostała w prezencie urodzinowym szachy. Poznała już zasady poruszania się hetmana (patrz zadanie: Hetman), teraz zainteresowała się figurą skoczka. Skoczek przesuwa się na najbliższe pole innego koloru niż wyjściowe, z wyłączeniem z nim sąsiadujących.

Zadanie

Małgosia zastanawia się, jaka jest najmniejsza liczba ruchów, które należy wykonać, aby przemieścić skoczka z zadanej pozycji startowej do wskazanej pozycji na nieskończonej szachownicy.



Rys. Możliwe ruchy skoczka, w 4 ruchach skoczek przemieszcza się z pola (2,2) do pola (5,7).

Wejście

W pierwszym wierszu wejścia znajduje się liczba D, określająca ilość zestawów danych, $1 \leq D \leq 5000$. W kolejnych wierszach wejścia znajdują się zestawy danych. Każdy z D zestawów danych składa się z jednej linii zawierającej, oddzielone spacjami, 4 liczby całkowite S_x, S_y, K_x, K_y, $-1000000 \leq S_x, S_y, K_x, K_y \leq 1000000$. Sx, Sy określają współrzędne startowe, K_x, K_y - współrzędne końcowe skoczka na nieskończonej szachownicy.

Wyjście

Dla każdego zestawu danych, w osobnych liniach wyjścia, należy wypisać minimalną liczbę ruchów potrzebną do przemieszczenia skoczka z pola (S_x,S_y) do pola (K_x,K_y).

Przykład

Dla danych:

```
3
2 2 5 7
3 2 4 6
```

```
0 0 2 2
```

poprawną odpowiedzią jest:

```
4
3
4
```

Komputerowa telepatia

ID:1055

Limit czasu: 1.50 s

Limit pamięci: 512 kB

Andrzej i Bartek (Alice i Bob jak lubią amerykanie) oglądali film "Ghost Busters", w którym Dr. Peter Venkman (grany przez Billa Murraya) przeprowadza badania zdolności telepatycznych studentów. Badanie to polega na pokazywaniu jednemu badanemu tzw. karty Zenera, która zawiera jeden z pięciu symboli: gwiazda, fale, krzyż, koło, kwadrat. Badany "nadawca" stara się przekazać jej obraz telepatycznie drugiemu badanemu ("odbiorcy"), który wskazuje jedną z pięciu kart.

Chłopcy postanowili powtórzyć tę zabawę przy pomocy komputerów. Po zastanowieniu doszli do wniosku, że na początek, kiedy nie mają jeszcze dużej wprawy, karty Zenera są zbyt skomplikowane i wymyślili swoje symbole: elipsę, prostokąt, trójkąt. Ustalili też, że osie elipsy i boki prostokąta będą zawsze równoległe do krawędzi obrazu, żaden z kątów trójkąta nie będzie rozwarty, co najmniej jeden bok trójkąta będzie równoległy do krawędzi obrazu. Andrzej namalował kilka takich figur czarnym pisakiem, i przesłał ich skanowane obrazy do Bartka, którego zadaniem było odgadnięcie jaki obraz został przesłany. Aby zagadka nie była zbyt prosta, pliki ze skanera zostały pozbawione nagłówków i zakodowane. Rozwiązanie utrudniała też niewielka (64 kB) pamięć komputerów, którymi dysponowali chłopcy (to był początek lat 80-tych!).

Spróbuj czy ty też jesteś dobrym komputerowym "telepatą" i czy odgadniesz, jaką figurę skanował Andrzej. Masz do dyspozycji znacznie więcej pamięci na przechowanie danych, bo aż 512 kB.

Zadanie

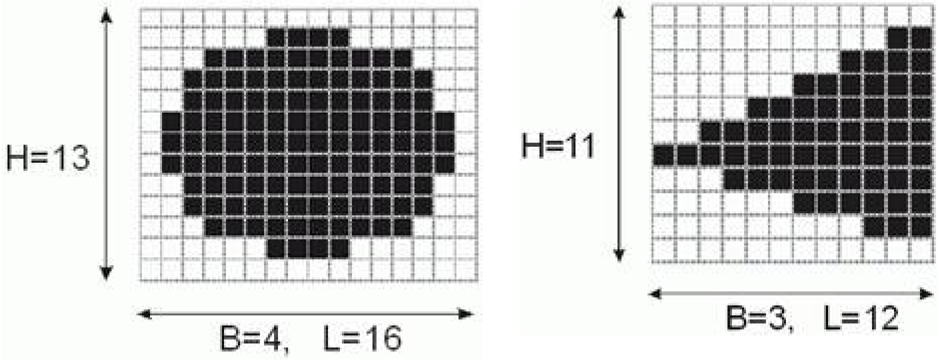
Należy podać numer identyfikujący figurę: 1 - prostokąt, 2 - elipsę, 3 - trójkąt.

Wejście

W pierwszym wierszu wejścia znajduje się liczba D, określająca ilość zestawów danych, $1 \leq D \leq 10$. Każdy z D zestawów danych składa się z dwóch wierszy. W pierwszym wierszu zestawu znajdują się dwie liczby całkowite H i B oddzielone jedną spacją. H oznacza wysokość obrazu liczoną w pikselach: $9 \leq H \leq 2000$, B liczbę znaków szesnastkowych przypadającą na jedną linię obrazu, $3 \leq B \leq 500$. Szerokość obrazu liczona w pikselach (L) jest zatem równa: L=4B. W drugim wierszu znajduje się HxB znaków ze zbioru: [0,1,2,..,9,A,B,C,D,E,F], oznaczających cyfry szesnastkowe, którymi został zakodowany skanowany rysunek.

Każda cyfra szesnastkowa oznacza 4 punkty obrazu (piksele). W jej rozwinięciu binarnym 1 oznacza kolor czarny, 0 oznacza białe tło. Pierwsza cyfra szesnastkowa opisuje lewy dolny narożnik obrazu.

Do konwersji obrazów z formatu .BMP (monochromatyczna bitmapa) można posłużyć się zamieszczonym programem.



Rys. Przykładowy obraz elipsy i trójkąta.

Wyjście

Na wyjściu, dla każdego zestawu, należy wypisać liczbę 1, 2 lub 3, która oznacza identyfikator figury.

Przykład

Dla danych:

```
2
13 4
000003C01FF83FFC3FFC7FFE7FFE7FFE3FFC3FFC1FF803C00000
11 3
00000703F1FFFFF3FF0FF03F00F003000
```

prawidłowym rozwiązaniem jest:

```
2
3
```

Czekoladka

ID:1056

Limit czasu: 1.00 s

Limit pamięci: 4096 kB

Ala obchodziła niedawno imieniny i zaprosiła swoich przyjaciół na małe przyjęcie. Każdy ze znajomych wiedząc, że jest ona wielkim łasuchem, przyniósł jej po tabliczce czekolady. Ala przygotowała dla swoich przyjaciół dużo innych smakołyków którymi mogła ich poczęstować, więc słodkie prezenty zostawiła na później.

Na drugi dzień Ala zebrała wszystkie czekolady, z zamiarem podzielenia się nimi ze swoim bratem. Zaczęła się jednak zastanawiać, czy jeżeli to ona weźmie pierwszy kawałeczek z pierwszej tabliczki, i będą brać po jednej "cegiełce" na zmianę, to kto zje ostatni kawałeczek ostatniej tabliczki?... Pomóż Ali rozwikłać ten problem.

Wejście

W pierwszym wierszu wejścia znajduje się liczba D, określająca ilość zestawów danych, $1 \leq D \leq 20$. W kolejnych wierszach wejścia znajdują się zestawy danych. W pierwszej linii jednego zestawu znajduję się liczba C, $1 \leq C \leq 100$, określająca liczbę tabliczek czekolady. W kolejnych C liniach zestawu znajdują się wymiary kolejnych tabliczek. Wymiary opisane są przez dwie liczby naturalne: a, b, $0 < a, b < 2^{31}$, oddzielone pojedynczą spacją.

Wyjście

Dla każdego zestawu danych, w osobnych liniach wyjścia, należy wypisać jedną z dwóch liczb: 0, jeśli ostatni kawałek zje Ala, 1 - jeśli ostatni kawałek zje jej brat.

Przykład

Dla danych:

```
2
4
10 20
3 3
36 3
11 99
1
5 7
```

poprawną odpowiedzią jest:

```
1
0
```

Liczby HEX-palindromiczne

ID:1057

Limit czasu: 1.00 s

Limit pamięci: 4096 kB

Nieujemną liczbę całkowitą *H* nazwiemy HEX-palindromiczną jeśli istnieje liczba naturalna $k > 1$ taka że, odwrócony zapis szesnastkowy liczby H jest taki sam jak zapis szesnastkowy liczby $H*k$ (rozpatrujemy wyłącznie zapisy szesnastkowe bez wiodących zer na początku).

Np. Liczba 17340 (szesnastkowo: 43BC) jest liczbą HEX-palindromiczną (17340*3=52020, szesnastkowo: 43BC*3=CB34).

Zadanie

Twoim zadaniem będzie wyznaczenie największej liczby HEX-palindromicznej mniejszej od zadanej liczby N.

Wejście

W pierwszym wierszu wejścia znajduje się liczba *C*, określająca ilość zestawów danych, $1 \leq C \leq 1000$. W kolejnych liniach wejścia znajdują się zestawy danych. Każdy z C zestawów danych składa się z jednego wiersza zawierającego zapis szesnastkowy liczby *N*, $N \geq 0$. Zapis szesnastkowy liczby *N* zawiera co najwyżej 10 znaków (nie występują w nim zera wiodące). Dozwolone znaki systemu szesnastkowego to cyfry 0-9, duże litery A,B,C,D,E,F.

Wyjście

Dla każdego zestawu danych, w osobnych liniach wyjścia, należy wypisać zapis szesnastkowy największej liczby HEX-palindromicznej mniejszej od zadanej liczby *N*. W przypadku, gdy taka liczba nie istnieje, należy wypisać liczbę 0.

Przykład

Dla danych wejściowych:

```
3
2000
F533
409F0
```

poprawną odpowiedzią jest:

```
10EF
43BC
21FDE
```

Nadajniki

ID:1058

Limit czasu: 1.00 s

Limit pamięci: 8192 kB

Jedna z wiodących na rynku sieci telefonii komórkowej wykonała szereg analiz komunikacji pomiędzy nadajnikami rozmieszczonymi na terenie całego kraju. W wyniku badań ustalono, że połączenie zostanie tak skonfigurowane, iż każdy nadajnik będzie komunikował się z każdym innym albo bezpośrednio, albo za pomocą tylko jednego nadajnika pośredniczącego. Aby zapewnić maksymalny komfort usług postanowiono, że w przypadku wyłączenia lub awarii pewnej liczby nadajników, komunikacja pomiędzy pozostałymi nadajnikami (o ile jest w ogóle możliwa) będzie wymagała także co najwyżej jednego nadajnika pośredniczącego. W celu ograniczenia liczby możliwych konfiguracji do sprawdzenia wykluczono połączenia typu "każdy z każdym".

Firma telekomunikacyjna rozpatrzyła efektywność i koszty każdej możliwej konfiguracji spełniającej powyższe założenia. Jednak ostatnio skradziono z firmy pewne dokumenty i nie wiadomo, czy były wśród nich wyniki badań sieci. Twoim zadaniem będzie wyznaczenie liczby możliwych konfiguracji sieci, aby można było określić, czy zaginęły wyniki badań (firma podejrzewa, że może być w to zamieszana konkurencja). Musisz wiedzieć, że w trakcie analiz nadajniki sieci komórkowej traktowano jako nierozróżnialne.

Wejście

W pierwszym wierszu wejścia znajduje się liczba zestawów danych C, $1 \leq C \leq 50$. Każdy z C zestawów danych składa się z jednego wiersza zawierającego liczbę naturalną N, $2 \leq N \leq 50$, oznaczającą całkowitą liczbę nadajników w sieci.

Wyjście

Dla każdego zestawu danych należy wyznaczyć liczbę możliwych konfiguracji komunikacji pomiędzy nadajnikami spełniających warunki zadania, przy założeniu, że nie rozróżniamy nadajników.

Przykład

Dla danych:

```
3
2
3
4
```

poprawną odpowiedzią jest:

```
0
1
4
```

Lublin-Kraków

ID:1059

Limit czasu: 1.50 s

Limit pamięci: 6144 kB

Kuba bardzo często jeździ pociągiem pomiędzy Lublinem a Krakowem. W trakcie licznych podróży zaobserwował, że liczba biletów sprzedanych przez firmę kolejową ma się nijak do liczby wolnych miejsc w przedziałach - ostatnio nawet musiał całą drogę stać na korytarzu. Postanowił to zmienić. Wymyślił system, który pozwoli firmie oszacować liczbę miejsc zajmowanych w pociągu na podstawie informacji o sprzedanych biletach. Okazało się jednak, że Kuba musi wyjechać służbowo na pewien czas i nie zdąży dokończyć systemu. Tobie jednak powierza zakodowanie najważniejszego algorytmu...

Każdy człowiek kupuje bilet na trasę od jednego miasta do drugiego, więc firma kolejowa może oszacować, że dany pasażer wsiądzie do pociągu danej linii w chwili *t0* a wysiądzie w chwili *t1*. Twoim zadaniem jest wyznaczyć maksymalną liczbę osób, które będą znajdowały się w pociągu. Pasażerowie są bardzo zdyscyplinowani, więc jeśli pociąg zatrzyma się na stacji, to najpierw opuszczają go wysiadający, dopiero później wsiadają nowi pasażerowie.

Wejście

W pierwszej linii znajduje się liczba osób B, $1 \le B \le 100000$, które zakupiły bilet na danej trasie. W kolejnych B liniach znajdują się liczby całkowite nieujemne: *t0* i *t1,* $0 \le t0 < t1 \le 2^{31} - 1$; oddzielone pojedynczą spacją. Są to odpowiednio: czas wejścia do pociągu oraz czas wyjścia z pociągu kolejnej osoby.

Wyjście

Na standardowym wyjściu powinna być wypisana jedna liczba, będącą maksymalną ilością osób znajdujących się równocześnie w pociągu.

Przykład

Dla danych wejściowych

```
3
1 2
2 3
2 4
```

poprawną odpowiedzią jest

```
2
```

Sztuka przetrwania

ID:1060

Limit czasu: 2.50 s

Limit pamięci: 16384 kB

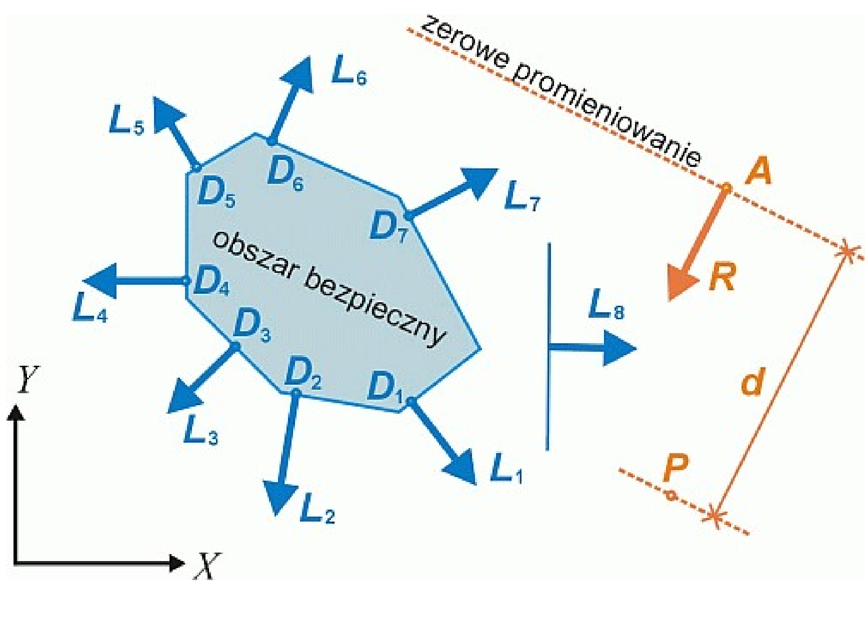
Jesteś projektantem kosmicznego robota-szperacza. Musisz oprogramować go tak, aby szybko odnajdywał punkty przestrzeni, w których panuje najmniejsze promieniowanie.

Ale do rzeczy. Trwa wojna światów, "Imperium Zła" kontratakuje i zdołało odeprzeć "Siły Jasności", a teraz przystępuje do ich niszczenia przy pomocy dział laserowych. Okazało się jednak, że istnieją obszary w przestrzeni gdzie działa nie sięgają, panuje tam jednak duże promieniowanie. Zadaniem robota jest odnaleźć miejsce o najmniejszym promieniowaniu, nie narażając się przy tym na zniszczenie. Jako dane wejściowe robot otrzymuje pozycje i główny kierunek rażenia dział oraz kierunek i intensywność wzrostu promieniowania.

Zadanie

Należy podać wartość promieniowania oraz współrzędne punktu, w którym to promieniowanie jest najmniejsze. Punkt musi należeć do obszaru leżącego poza zasięgiem dział laserowych. Dane, którymi dysponuje robot są następujące:

1. Liczba dział laserowych (N), wymiar przestrzeni (W) w której rozgrywa się wojna (W=2 - obszar płaski, W=3 - przestrzeń trójwymiarowa, W>3 - hiperprzestrzeń).
2. Działo laserowe o numerze i opisane jest przez współrzędne punktu działania ***Di*** i składowe wektora ***Li*** wskazującego główny kierunek rażenia. Działo ostrzeliwuje tylko tę połowę płaszczyzny (przestrzeni, hiperprzestrzeni), którą wskazuje wektor ***Li***. Przebywanie na prostej (płaszczyźnie, hiperpłaszczyźnie) rozdzielającej obie połowy jest bezpieczne, wkroczenie na obszar wskazywany przez wektor powoduje zniszczenie robota. Niektóre działa mogą znajdować się obszarze ostrzału (np. L_8 na rysunku) - ponieważ nie ograniczają one obszaru bezpiecznego, powinny być zignorowane.
3. Współrzędne punktu ***A*,** w którym promieniowanie jest zerowe oraz składowe wektora promieniowania ***R*,** wskazującego kierunek wzrostu i intensywność promieniowania. Wartość promieniowania w punkcie ***P***, oblicza się mnożąc moduł wektora promieniowania **|*R*|** przez odległość d punktu ***P*** od prostej (płaszczyzny, hiperpłaszczyzny) do której wektor promieniowania jest prostopadły.



Wejście

W pierwszym wierszu wejścia podana jest para liczb naturalnych, oddzielona jedną spacją: N W, gdzie N jest liczbą dział laserowych, $2<N<1024$, a W wymiarem przestrzeni, $1<W\leq10$. Drugi wiersz zawiera 2W liczb rzeczywistych oddzielonych spacjami, z których W początkowych liczb jest współrzędnymi punktu ***A*** (punktu leżącego na granicy zerowego promieniowania), a W następnych liczb oznacza składowe wektora promieniowania R**.** Dalej następuje N wierszy po 2W liczb każdy, które oznaczają współrzędne pozycji działa ***Di*** i składowe wektora ***Li*** wskazującego główny kierunek rażenia działa. Wszystkie współrzędne punktów (Aj, Dij, $1 \leq i \leq N$, $1 \leq j \leq W$) występujące na wejściu spełniają warunki: $0 \leq Aj, Dij < 10^9$. Wszystkie składowe wektorów (Rj, Lij, $1 \leq i \leq N$, $1 \leq j \leq W$) występujące na wejściu spełniają warunki: $-10^9 < Rj, Lij < 10^9$.

Wyjście

Na wyjściu, w jednym wierszu, należy wypisać W+1 liczb całkowitych oddzielonych pojedynczymi spacjami. Pierwsza liczba p oznacza wartość promieniowania w znalezionym punkcie, $0\leq p <109$, a W następnych liczb to współrzędne tego punktu, $-10^9< Xi <10^9$. Dane występujące w testach zostały tak dobrane, ze wynikami są liczby całkowite. Dane wejściowe gwarantują, że znaleziony punkt jest jednoznaczny.

Przykład

```
8 2
 80.0  70.0  -9.0 -18.0
 55.5  11.0  12.0 -17.0
 36.0   7.0  -4.0 -22.0
 16.0  18.0 -18.0 -18.0
  7.0  39.0 -24.0   0.0
 13.5  54.5  -7.0  13.0
 34.0  52.0  12.0  28.0
 56.0  31.5  29.0  16.0
 70.0  15.0  14.0   0.0
```

prawidłowym rozwiązaniem jest:

```
720 48 46
```

Segregacja

ID:1061

Limit czasu: 1.00 s

Limit pamięci: 8192 kB

Firma Miracle, idąc z duchem czasu postanowiła stać się firmą przyjazną środowisku naturalnemu (w chwili obecnej 30% energii elektrycznej pochodzi ze źródeł odnawialnych). Zarząd firmy podjął decyzję, że wszystkie odpady bedą segregowane i umieszczane w specjalnych kontenerach, w każdym inny rodzaj. Segregowanie odpadów powoduje dość duże przestoje w pracy - więc w interesie Miracle jest, aby trwało możliwie najkrócej.

Odpady będą segregowane przez robota, który w danym momencie może przenosić tylko jeden przedmiot. Czynności wykonywane przez robota sprowadzają się do wyjęcia przedmiotu z kontenera i przełożenie go do docelowego kontenera. Jeden przedmiot może być przeniesiony tylko jeden raz. Przyjmujemy, że koszt przeniesienia jednego przedmiotu wynosi 1 jednostkę czasu. Zakładamy, że kontenery mają nieograniczoną pojemność. Jeżeli przedmioty są posortowane, to robot kończy pracę.

W celu poprawnego zaplanowania całej operacji należy określić, ile czasu zajmuje segregowanie w najlepszym i najgorszym przypadku.

Wejście

W pierwszym wierszu znajduje się liczba kontenerów N, $1\leq N\leq 200$ równa liczbie rodzajów odpadów. W kolejnych N wierszach znajduje się opis zawartości kontenerów. Zawartość każdego kontenera określa N liczb $x_1,x_2,\dots, x_n$ równych odpowiednio ilości przedmiotów rodzaju 1,2..n $0\leq x_i\leq 1000$, $1\leq i\leq N$.

Wyjście

Na wyjściu powinny znaleźć się dwie liczby równe odpowiednio minimalnej i maksymalnej liczbie jednostek czasu potrzebnej na posegregowanie odpadów.

Przykład

Dla danych

```
2
2 3
4 3
```

poprawnym rozwiązaniem jest

```
5 7
```

Szyfr

ID:1062

Limit czasu: 1.00 s

Limit pamięci: 8192 kB

Agencja wywiadu, która zleca Ci od czasu do czasu tajne zadania, ma dla Ciebie kolejny problem do rozwiązania. W jej OPeracyjnej Sieci Szyfrów (OPSS) pojawił się kolejny szyfr. Według informacji agentów jest on wykorzystywany przez terrorystów do szyfrowania informacji. Na szczęście udało się znaleźć algorytm, którym kodowane są wiadomości. Twoim zadaniem jest napisanie programu, który odkoduje informację, a zatem będzie działał odwrotnie niż znaleziony algorytm. Algorytm wygląda następująco:

C/C++:

```
int foo ( int n )
{
  return n ^ (n >> 1);
}
```

Pascal:

```
function foo ( n : longint ) : longint;
begin
  foo := n xor (n shr 1);
end;
```

Jeśli Twoja funkcja będzie nazywać się **oof** wówczas wynik wywołania **oof(foo(*n*))** powinien dać liczbę *n*.

Zadanie

Napisz program, który dla zadanej liczby całkowitej *n* obliczy wartość **foo**(*n*) oraz **oof**(*n*).

Wejście

Każda linia wejścia zawiera dokładnie jedną liczbę całkowitą n, $0 \le n \le 2^{31} - 1$. Wczytywanie liczb z wejścia należy zakończyć gdy *n* będzie równe 0 - dla tego wiersza Twój program nie powinien nic wypisywać na standardowym wyjściu.

Wyjście

I-ta linia wyjścia powinna zawierać dokładnie dwie wartości: **foo**(*n*), (*n*) oddzielone pojedyńczą spacją, gdzie *n* jest *I*-tą wczytaną liczbą, **foo**(*n*) to wynik szyfrowania liczby *n*, zaś **oof**(*n*) to wynik deszyfrowania liczby *n*.

Przykład

Dla następującego wejścia:

```
1
123
1001
689
0
```

poprawną odpowiedzią jest:

```
1 1
70 82
541 689
1001 801
```

Rekursywna bakteria czwórkowa

ID:1063

Limit czasu: 1.00 s

Limit pamięci: 6144 kB

Rekursywna bakteria czwórkowa to bakteria, która żyje na powierzchniach kwadratowych. Bakteria składa się z jednego chromosomu typu B, C (chromosomy proste) lub S (chromosomu złożonego). Chromosom złożony S zbudowany jest z czterech komórek, z których każda jest pojedynczą rekursywną bakterią czwórkową.

Ale to nie wszystko co charakteryzuje taką bakterię. Jej życie możemy podzielić na dwie dekady: dekadę przeobrażania, a po niej dekadę obumierania. Podczas pierwszej dekady życia pewne komórki bakterii przekształcają się aż do momentu, kiedy już proces przeobrażania nie może zajść. Wówczas bakteria zaczyna obumierać (rozpoczyna się druga dekada). Proces przeobrażania polega na tym, że chromosomy typu S, które składają się z chromosomów tego samego typu (B lub C) stają się chromosomem tego jednego typu (B lub C). Inaczej mówiąc: 4 komórki tego samego typu łączą się, tworząc jedną komórkę danego typu.

Do zapisu budowy bakterii używa się tzw. meta-chromosomu - ciągu złożonego ze znaków B, C, S. Jeżeli bakteria składa się chromosomu prostego B lub C, wówczas jej meta-chromosom ma postać znaku odpowiednio B lub C. Jeżeli natomiast bakteria składa się z chromosomu złożonego S, wówczas jej meta-chromosom jest ciągiem opisu meta-chromosomów bakterii wchodzących w skład S.

```
+----+----+----+----+
|    |    |    |    |
| C  |  C |  C |  B |
+----+----+----+----+
|    |    |    |    |
| C  |  C |  C |  B |
+----+----+----+----+
|         |    |    |
|         |  B |  B |
|     B   +----+----+
|         |    |    |
|         |  B |  B |
+---------+----+----+
```

Rys. Rekursywna bakteria czwórkowa (cykl 1)

Meta-chromosom dla bakterii na powyższym rysunku będzie miał postać: SSCCCCSCBCBBSBBBB. Po procesie przeobrażania (każde przekształcenie trwa 1 cykl) bakteria będzie miała postać:

```
+---------+----+----+
|         |    |    |
|         | C  | B  |
|    C    +----+----+
|         |    |    |
|         | C  | B  |
+---------+----+----+
|         |         |
|         |         |
|    B    |    B    |
|         |         |
|         |         |
+---------+---------+
```

Rys. Rekursywna bakteria czwórkowa (cykl 2)

a opisujący ją meta-chromosom skróci się do ciągu: SCSCBCBBB w czasie 2 cykli.

Zadanie

Twoim zadaniem jest oprogramowanie systemu dla bakteriologów. Dla zadanej rekursywnej bakterii czwórkowej w postaci meta-chromosomu, wyznacz postać meta-chromosomu oraz numer cyklu, od którego bakteria zacznie obumierać (czyli przeobrażanie bakterii nie będzie więcej zachodzić).

Wejście

W pierwszej linii wejścia znajduje się liczba C, określająca liczbę zestawów danych, $1 \leq C \leq 30$. W kolejnych wierszach wejścia znajdują się zestawy danych. Każdy z C zestawów danych składa się z wiersza zawierającego niepusty ciąg znaków, będący meta-chromosomem. Długość ciągu nie przekracza 100000.

Wyjście

Dla każdego zestawu danych, w osobnej linii wyjścia, należy wypisać meta-chromosom oraz numer cyklu (oddzielone pojedyncza spacją), od którego bakteria zacznie obumierać.

Przykład

Dla danych:

```
3
SSCCCCSCBCBBSBBBB
SSCCCCSCCCCSCCCCSCCCC
SBBBB
```

poprawną odpowiedzią jest:

```
SCSCBCBBB 2
C 3
B 2
```

Maszyna drukarska

ID:1064

Limit czasu: 1.00 s

Limit pamięci: 4096 kB

W pewnej drukarni MAT-DRUK drukuje się wyłącznie książki matematyczne, które - jak to w świecie królowej nauk - zawierają bardzo skomplikowane wyrażenia i symbole. Niektóre mieszczą w sobie nawet do 1000 nawiasów!

Maszyna drukarska działała niezawodnie do czasu, gdy nadeszła mroźna zima. Wtedy na najczęściej używanych czcionkach: "(", ")" osiadł szron. Maszyna zaczęła losowo opuszczać znaki nawiasów "(", ")".

Dla przykładu, wyrażenie zawierające nawiasowanie:

((()))()((()))

wydrukowała z pominięciem nawiasów 3 i 11 w postaci:

(()))()(()))

Drukarnia jest w poważnych tarapatach. Aż strach pomyśleć, co może się wydarzyć, jeżeli urządzenie przez dłuższy czas będzie drukować błędne wyrażenia.

Zadanie

Pomóż drukarzom określić, jaka jest najmniejsza liczba nawiasów, które w wydrukowanym nawiasowaniu (prawdopodobnie błędnym) należy dostawić, aby otrzymać nawiasowanie poprawne. Nawiasowaniem poprawnym są ciągi:

1) ()

2) (*X*), gdzie *X* jest poprawnym nawiasowaniem

3) *XY*, gdzie *X, Y* są poprawnymi nawiasowaniami

Wejście

W pierwszej linii wejścia znajduje się liczba C, określająca liczbę zestawów danych, $1 \le C \le 100$. W kolejnych wierszach wejścia znajdują się zestawy danych. Każdy z C zestawów danych składa się z wiersza zawierającego niepusty ciąg nawiasów, który wydrukowała zepsuta maszyna drukarska. Długość ciągu nie przekracza 1000 znaków.

Wyjście

Dla każdego zestawu danych, w osobnej linii wyjścia, należy wypisać liczbę, określającą liczbę nawiasów które należy wstawić do zadanego ciągu, aby otrzymać poprawne nawiasowanie.

Przykład

Dla danych wejściowych:

```
2
)
(()))()(()))
```

poprawną odpowiedzią jest:

```
1
2
```

Kąt obrotu

ID:1065

Limit czasu: 2.00 s

Limit pamięci: 4096 kB

Znani komputerowi telepaci Andrzej i Bartek (o których przygodach można przeczytać w zadaniu "Komputerowa telepatia") po kilku miesiącach doszli do takiej wprawy, że bezbłędnie rozpoznawali wszystkie przesyłane do siebie figury. Postanowili zatem spróbować czegoś trudniejszego. Po zastanowieniu uzgodnili, że będą odgadywać kąt, o jaki obrócona została figura w stosunku do "standardowego" położenia.

Wybrali prostokąt i elipsę o proporcjach wyraźnie różniących je od kwadratu i koła, tzn. długość krótszego boku prostokąta (krótszej osi elipsy) mieści się w przedziale od 0.2 do 0.8 dłuższego boku (osi). Standardowym położeniem figury jest takie, w którym dłuższy bok (dłuższa oś) jest ułożony poziomo.

Podobnie jak kiedyś, w telepatii, figury namalowane czarnym flamastrem zostały zeskanowane, ale już bez zwracania uwagi na położenie, zatem figura może być obrócona o kąt alfa o wartości mieszczącej się w przedziale -85 ≤ alfa ≤ 90 stopni (dla ułatwienia założono, że wartość kąta jest wielokrotnością 5 stopni).

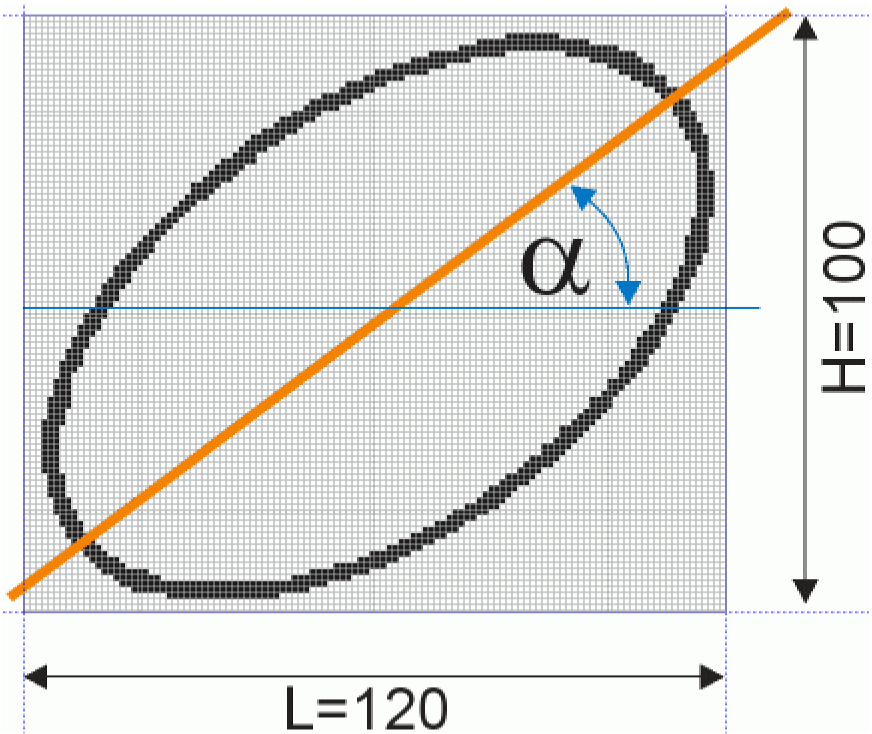
Wnętrze figury może być całkowicie zaczernione lub puste, kontur figury jest zawsze wypełniony czarnymi pikselami bez przerw. Pliki skanera przed przesłaniem zostały pozbawione nagłówków i zakodowane. Kodowanie polega na założeniu, że obraz składa się z linii równoległych do osi X, zatem do określenia położenia i-tej linii wystarczy podać 3 liczby Yi, X1i, X2i, gdzie Yi, X1i oznaczają współrzędne punktu początkowego a Yi, X2i punktu końcowego i-tej linii, X1i ≤ X2i.

Zadanie

Należy podać wartość kąta alfa, o który została obrócona figura w stosunku do położenia "standardowego", z dokładnością do 5 stopni.

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita N, 1 ≤ N ≤ 5, oznaczająca liczbę zestawów danych, zaś w następnych N wierszach opisane są obrazy obróconych figur. W każdym z N wierszy znajdują się trójki liczb naturalnych oddzielonych pojedynczymi spacjami. Każda trójka (postaci *Y*, *X1*, *X2*, $0 \leq Y, X1, X2 \leq 1000$) określa położenie (współrzędne punktów początkowych i końcowych) jednej z linii obrazu. Założono, że oś X jest osią poziomą skierowaną w prawo a oś Y osią pionową skierowaną do góry. Trójek liczb może być więcej niż wynosi wysokość obrazu mierzona w pikselach bo możliwe są np. dwa odcinki w jednej linii (liczba trójek jest z zakresu 100..2000). Koniec wiersza kończy opis jednego obrazu.



Rysunek przedstawia elipsę opisaną w przykładzie, można go powiększyć aby zobaczyć więcej szczegółów.

Wyjście

Na wyjściu dla każdego zestawu w osobnej linii, należy wypisać jedną liczbę całkowitą podzielną przez 5, należącą do przedziału <-85..90>, oznaczającą kąt, o który obrócono skanowaną figurę. Dodatni kąt obrotu pokazany jest na rysunku.

Przykład

Dla danych (2 linie wejścia, elipsa z obrazka):

```
1

3 25 43 4 19 48 5 17 51 6 16 23 6 46 55 7 14 21 7 49 58 8 13 17 8 53 61 9 12 16
9 57 63 10 11 14 10 59 65 11 10 13 11 62 67 12 9 12 12 64 69 13 8 11 13 66 71 14
8 10 14 68 73 15 7 10 15 70 74 16 7 9 16 72 76 17 6 9 17 73 78 18 6 8 18 75 80
19 6 8 19 77 81 20 5 7 20 78 82 21 5 7 21 79 83 22 5 7 22 81 85 23 5 7 23 82 86
24 4 6 24 84 87 25 4 6 25 85 89 26 4 6 26 86 90 27 4 6 27 88 91 28 4 6 28 88 92
29 4 6 29 90 93 30 4 6 30 91 94 31 4 6 31 91 95 32 4 7 32 93 96 33 5 7 33 94 97
34 5 7 34 95 98 35 5 8 35 96 99 36 5 8 36 96 99 37 6 8 37 98 100 38 6 8 38 99
101 39 6 8 39 99 102 40 7 8 40 101 103 41 7 9 41 101 104 42 7 10 42 102 105 43 8
10 43 103 105 44 8 10 44 103 106 45 9 11 45 104 107 46 9 11 46 104 107 47 10 12
47 105 108 48 10 13 48 106 108 49 11 13 49 107 109 50 11 14 50 107 109 51 12 14
51 108 110 52 13 15 52 108 110 53 13 16 53 109 111 54 14 16 54 109 111 55 14 17
55 110 112 56 15 18 56 110 112 57 16 18 57 111 113 58 16 19 58 111 113 59 17 20
59 112 114 60 18 21 60 112 114 61 19 22 61 112 115 62 20 22 62 112 115 63 21 23
63 112 115 64 22 24 64 113 115 65 23 25 65 113 115 66 24 26 66 113 115 67 25 27
67 114 116 68 26 28 68 114 116 69 27 29 69 114 116 70 28 31 70 114 116 71 29 32
71 114 116 72 29 33 72 114 116 73 31 35 73 114 116 74 32 36 74 114 116 75 33 37
75 114 115 76 34 39 76 114 115 77 36 39 77 114 115 78 37 41 78 113 115 79 38 43
79 113 115 80 39 44 80 112 115 81 41 46 81 112 114 82 43 47 82 111 114 83 44 49
83 111 113 84 46 50 84 110 113 85 48 53 85 110 112 86 49 55 86 109 112 87 51 57
87 108 110 88 54 59 88 107 110 89 55 61 89 105 108 90 58 64 90 103 108 91 60 68
```

```
91 101 106 92 63 71 92 99 106 93 66 75 93 96 104 94 69 81 94 92 103 95 74 100 96
77 95 97 82 90
```

prawidłowym rozwiązaniem jest:

```
35
```

Adresy IP

ID:1066

Limit czasu: 2.00 s

Limit pamięci: 4096 kB

Jasio już od dłuższego czasu gromadzi adresy IP (miejmy nadzieję, że Jasio nie jest spamerem). Pewnego dnia postanowił uporządkować swoją kolekcję. Okazuje się, że jest ona dość specyficzna, bowiem w jej skład wchodzą wszystkie możliwości adresów, które zawierają zawsze takie same liczby. Różnią się jedynie kolejnością występowania tych liczb. Wiadomo również, że nie występują w niej dwa takie same adresy.

Adres IP w danej wersji protokołu P składa się z dokładnie P liczb całkowitych z zakresu 0..255, oddzielonych od siebie kropką.

Pomoż Jasiowi w uporządkowaniu jego zbioru adresów IP.

Wejście

W pierwszym wierszu wejścia podana jest liczba P, będąca numerem wersji protokołu IP, $3 \leq P \leq 10$, dla adresów Jasia. Druga linia wejścia składa się z dokładnie P liczb całkowitych, z zakresu 0..255. Są to liczby, z których zbudowane są adresy z kolekcji Jasia.

Wyjście

W oddzielnych liniach wyjścia, należy wypisać w porządku rosnącym adresy IP, pochodzące z kolekcji Jasia, jakie można utworzyć z zadanych P liczb. Jeżeli adresów IP jest więcej niż 10000, należy wypisać pierwsze 10000 adresów.

Przykład

Dla danych wejściowych:

```
3
255 0 1
```

poprawną odpowiedzią jest:

```
0.1.255
0.255.1
1.0.255
1.255.0
255.0.1
255.1.0
```

Dla danych wejściowych:

```
4
127 0 0 1
```

poprawną odpowiedzią jest:

```
0.0.1.127
0.0.127.1
```

```
0.1.0.127
0.1.127.0
0.127.0.1
0.127.1.0
1.0.0.127
1.0.127.0
1.127.0.0
127.0.0.1
127.0.1.0
127.1.0.0
```

Suma cyfr

ID:1067

Limit czasu: 2.00 s

Limit pamięci: 4096 kB

Zadanie

Dla zadanych liczb całkowitych *x*, *y,* a, należy wyznaczyć, ile jest liczb całkowitych pomiędzy *x* a *y* (włącznie), których suma cyfr wynosi *a.*

Wejście

W pierwszej linii wejścia znajduje się liczba *C*, określająca liczbę zestawów danych, $1 \leq C \leq 1000$. W kolejnych wierszach wejścia znajdują się zestawy danych. Każdy z C zestawów danych składa się z wiersza zawierającego liczby całkowite *x, y*, a ($0 \leq x \leq y < 2^{31}$, $0 \leq a \leq 100$), oddzielone pojedyncza spacją.

Wyjście

Dla każdego zestawu danych, w osobnej linii wyjścia, należy wypisać liczbę, określającą ile jest liczb całkowitych pomiędzy *x* a *y*, których suma cyfr jest równa a.

Przykład

Dla danych wejściowych:

```
3
1 100 11
90 2400 14
6502 68020 16
```

poprawną odpowiedzią jest:

```
8
180
2807
```

Szkrable

ID:1068

Limit czasu: 2.00 s

Limit pamięci: 16384 kB

Szkrable jest bardzo popularną i niezwykle skomplikowaną grą słowną polegającą na układaniu słów przez graczy. Jedna z setek zasad tej gry mówi, że słowo jest dopuszczalne do gry jeżeli każdy jego spójny podciąg o długości trzech liter jest wyrazem ze słownika ustalonego przez graczy na początku gry. Wyrazy w słowniku (nazywane 'trójkami') nie powtarzają się. Każda 'trójka' składa się z trzech różnych dużych liter alfabetu angielskiego i ma przyporządkowaną dodatnią liczbę punktów. Za utworzone słowo gracz otrzymuje liczbę punktów równą sumie punktów za każdą 'trójkę' występującą w tym słowie. Jeśli 'trójka' występuje w słowie więcej niż jeden raz, wówczas gracz otrzymuje punkty osobno za każde wystąpienie 'trójki' w słowie. Jedna z zasad szczególnych mówi, że słowo jednoliterowe lub dwuliterowe jest słowem dopuszczalnym jeżeli zawiera się w którymkolwiek z wyrazów ze słownika. Za utworzone słowa jednoliterowe i dwuliterowe gracz nie otrzymuje żadnych punktów.

Zadanie

Pan Henryk od niedawna gra w Szkrable. Pragnie szybko stać się dobrym graczem, dlatego potrzebuje Twojej pomocy. Chciałby wiedzieć, ile może maksymalnie stracić punktów za ułożone przez siebie dopuszczalne do gry słowo, jeżeli będzie do niego dodawał i/lub usuwał litery tak, żeby zmodyfikowane słowo wciąż było dopuszczalne do gry, zaczynało się na tę samą literę na którą zaczyna się oryginalne słowo sprzed modyfikacji i kończyło się na tę samą literę co słowo sprzed modyfikacji. Napisz program, który mu to ułatwi.

Wejście

W pierwszym wierszu wejścia znajduje się liczba n ($1 \le n \le 1000$), określająca liczbę wyrazów w słowniku. W kolejnych n wierszach opisane są 'trójki' ze słownika, jedna 'trójka' w każdym wierszu. Każdy wiersz z opisem 'trójki' składa się z wyrazu T oraz liczby całkowitej K, oddzielonych pojedynczą spacją. Wyraz T jest trzyliterowym wyrazem ze słownika, któremu przyporządkowana jest liczba punktów K ($1 \le K \le 1000$). Następny wiersz wejścia zawiera liczbę q ($1 \le q \le 1000$). W kolejnych q wierszach znajdują się niepuste słowa utworzone przez pana Henryka, po jednym słowie w każdym wierszu. Długość każdego ze słów nie przekracza 1000 liter.

Wyjście

Dla każdego słowa pana Henryka należy w jednym wierszu wyjścia wypisać dwie liczby oddzielone spacją:

- liczbę punktów jaką może on otrzymać za ułożone słowo
- maksymalną liczbę punktów o jaką może pomniejszyć swój wynik modyfikując ułożone słowo tak, żeby dalej było dopuszczalne do gry i żeby zaczynało się i kończyło na te same litery na które zaczyna się i kończy słowo niezmodyfikowane.

Przykład

Dla danych wejściowych:

```
9
DAB 10
```

```
ABC 20
BCD 25
CDZ 100
ADZ 1000
RCB 40
RCD 1000
CBA 30
CDA 50
3
ADZ
RCDABC
RCDZ
```

poprawną odpowiedzią jest:

```
1000 855
1080 1080
1100 0
```

Gugle

ID:1069

Limit czasu: 2.50 s

Limit pamięci: 6144 kB

Grupa programistów OPSS, postanowiła wzbogacić swój system o dobrą wyszukiwarkę tekstową. W trakcie pracy nad głównym algorytmem programiści zdali sobie sprawę, że dobra wyszukiwarka powinna nie tylko działać szybko, lecz również pomagać użytkownikom, gdy pomylą się przy wpisywaniu jakiegoś słowa lub gdy takiego słowa nie ma w słowniku.

Programiści wpadli na pomysł, żeby w momencie wpisania hasła, którego nie znaleziono w słowniku, zwrócić użytkownikowi listę słów podobnych do wpisanej frazy. Lista powinna być posortowana od najbardziej do najmniej podobnych.

Podobieństwo hasła *H1* do *H2* wyznaczamy w ten sposób, że sumujemy minimalną liczbę liter, które należy wstawić, usunąć lub wymienić w słowie *H1* aby uzyskać *H2*.

Na przykład, aby ze słowa 'ALA' otrzymać 'MAREK' musimy wykonać łącznie 4 operacje:

1.Dodajemy na początek literkę 'M', otrzymujemy 'MALA'

2.Wymieniamy literkę 'L' na 'R', otrzymujemy 'MARA'

3.Wymieniamy drugą literkę 'A' na 'E', otrzymujemy 'MARE'

4.Dodajemy na koniec literkę 'K', otrzymujemy 'MAREK'

Dla hasła 'OLA' wartość podobieństwa do 'MAREK' równa jest 5, gdyż usuwamy literę 'O', wymieniamy literę 'L' na 'M' i dodajemy na koniec 3 litery 'R', 'E' i 'K'.

Jak widać słowo 'ALA' jest bardziej podobne do hasła 'MAREK' niż 'OLA'. Dlatego, gdy użytkownik wpisze hasło 'MAREK', którego nie ma w słowniku, system powinien podpowiedzieć mu wypisując posortowaną rosnąco listę - w tym przypadku byłoby to: 'ALA' 'OLA'

Zadanie

Napisz program, który dla danego słownika S i hasła T, posortuje listę słów S niemalejąco względem podobieństwa do hasła T (od najbardziej do najmniej podobnych).

Wejście

W pierwszym wierszu znajduje się liczba zestawów danych D, $1 \leq D \leq 30$. W kolejnych wierszach znajdują się zestawy danych. W pierwszym wierszu zestawu podane jest hasło wzorcowe T. W drugim znajduje się liczba N, $1 \leq N \leq 20$ haseł w liście słownikowej, którą należy posortować. W kolejnych N wierszach znajdują się hasła. Wszystkie słowa składają się z dużych liter alfabetu angielskiego. Długość hasła w słowniku jest nie mniejsza niż 1 znak i nie większa niż 200 znaków.

Wyjście

Dla każdego zestawu danych należy wypisać na wyjściu jedną linię, składającą się z N haseł oddzielonych pojedynczą spacją. Hasła powinny być posortowane względem podobieństwa do wzorca *T* (od najbardziej do najmniej pasującego). W przypadku, gdy dwa słowa są tak samo podobne do wzorca *T*, należy je wypisać w takiej kolejności, w jakiej pojawiły się na wejściu.

Dla danych wejściowych:

```
2
MAREK
4
OLA
HELA
ALA
KERAM
DAREK
5
FOO
ABCDEFGH
DAREEC
FOOBAR
KERAD
```

poprawnym rozwiązaniem jest:

```
ALA KERAM OLA HELA
DAREEC KERAD FOO FOOBAR ABCDEFGH
```

Jedenastka

ID:1070

Limit czasu: 1.00 s

Limit pamięci: 4096 kB

Firma Miracle dbająca o stały i niczym nie zachwiany rozwój swojej bazy danych, wymyśliła nowy, super szybki indeks na kolumnach przechowujących liczby składające się co najwyżej z 20 cyfr. Szczegóły dotyczące zasady działania indeksu, są najściślej strzeżoną tajemnicą firmy. Wiadomo jedynie, że jest związana z liczbą 11.

Grupa testująca otrzymała szereg zapytań o wartości kluczy w tym indeksie. W celu wyznaczenia wskaźników, które pozwolą sklasyfikować badany indeks w rankingu najlepszych indeksów Miracle, naukowcy z firmy muszą wiedzieć ile jest możliwych wartości indeksu dla zadanego wzorca klucza i reszty z dzielenia przez 11. Niestety, sami nie potrafią tego zrobić. Musisz im pomóc.

Wzorzec klucza jest ciągiem składającym się z symboli 'X','0','1','2','3','4','5','6','7','8',9'. Symbol 'X' zastępuje jedną cyfrę ze zbioru '0','1','2','3','4','5','6','7','8','9'. Przykładowe wzorce kluczy w indeksie mogą wyglądać następująco: 'XXX', 'X1X', '2XXX'. Do wzorca można dopasować jedynie liczby - dopasowania nie mogą zawierać zer znaczących. Na przykład, '012' nie jest poprawnym dopasowaniem do wzorca 'X12'.

Zadanie

Wyznacz ilość liczb pasujących do wzorca W, które przy dzieleniu przez 11 dają resztę r.

Wejście

W pierwszym wierszu wejścia znajduje się liczba zestawów danych n, $1 \leq n \leq 100$. W kolejnych wierszach następują zestawy danych. Każdy zestaw składa się z dwóch linii. W pierwszej jest liczba cyfr d, $1 \leq d \leq 20$ we wzorcu i reszta r, $0 \leq r \leq 10$ z dzielenia przez 11. W drugiej linii zestawu znajduje się wzorzec W długości d zawierający znaki 'X','0','1','2','3','4','5','6','7','8','9'.

Wyjście

Dla każdego zestawu danych na wyjściu powinna znaleźć się jedna wartość P określająca ilość liczb pasujących do wzorca W, które przy dzieleniu przez 11 dają resztę r. Możesz założyć, że $0 \leq P < 2^{63}-1$.

Przykład

Dla następujących danych

```
3

2 5

XX

1 8
```

```
X
1 8
7
```

poprawnym rozwiązaniem jest

```
8
1
0
```

Doskonałe trójkąty pitagorejskie

ID:1071

Limit czasu: 3.00 s

Limit pamięci: 16384 kB

Któż nie zna twierdzenia Pitagorasa?! Albo kto nie słyszał o liczbach pierwszych? Z pewnością nie uczestnicy OPSS-esji!, którzy kochają matematykę! Oto zadanie, które zawiera oba te pojęcia...

Trójkąty, których boki (a, b, c) spełniają równanie Pitagorasa: $a^2 + b^2 = c^2$ i jednocześnie a, b, c są liczbami całkowitymi, nazywane są trójkątami pitagorejskimi.

Jest ich nieskończenie wiele, a najmniejszym z nich jest tzw. "trójkąt egipski" o bokach: 3, 4, 5. Jeżeli zwiększymy wymagania i zechcemy, aby dwa z boków były liczbami pierwszymi, otrzymamy trójkąty, które można nazwać "doskonałymi trójkątami pitagorejskimi" (nie jest to nazwa oficjalna), których również jest nieskończenie wiele, ale znacznie mniej niż "zwykłych" trójkątów pitagorejskich. Posortowane według długości najkrótszego boku dają się łatwo zidentyfikować. Wspomniany trójkąt egipski jest pierwszym w ciągu "doskonałych trójkątów pitagorejskich".

Twoim zadaniem jest znalezienie wskazanego "doskonałego trójkąta pitagorejskiego".

Zadanie

Należy podać dwie liczby pierwsze, które są bokami n-tego doskonałego trójkąta pitagorejskiego.

Wejście

W pierwszym wierszu wejścia znajduje się liczba naturalna C, $1 \leq C \leq 1000$, oznaczająca liczbę zestawów danych. W następnych C wierszach podane są liczby całkowite oznaczające numery doskonałych trójkątów. Numery trójkątów są liczbami naturalnymi z przedziału 1..4000.

Wyjście

Na wyjściu, w każdym z C wierszy należy wypisać, oddzielone pojedynczą spacją, dwie liczby pierwsze $m, n, m \leq n,$ równe bokom trójkąta pitagorejskiego o podanym numerze.

Przykład

Dla danych wejściowych:

```
2
1
7
```

prawidłowym rozwiązaniem jest:

```
3 5
61 1861
```

Marsjańskie skały

ID:1072

Limit czasu: 1.50 s

Limit pamięci: 16384 kB

Naukowcy szukający śladów życia na Marsie, postanowili dokonać pewnych badań na próbkach skał pobranych z marsjańskiej gleby. Każdą próbkę można scharakteryzować poprzez podanie jej 10 parametrów. Każdy parametr jest liczbą całkowitą z zakresu 0-100. Uczeni sporządzili tabelę, w której zestawili liczebności poszczególnych rodzajów skał.

Zadanie

Znając tabelę z wynikami badań, pomóż naukowcom określić liczbę skał o zadanych parametrach.

Wejście

W pierwszym wierszu wejścia znajduje się liczba naturalna k ($1 \leq k \leq 10000$), określająca ilość wierszy w tabeli badań. W następnych k liniach znajdują się wiersze tabeli. Każdy wiersz składa się z 11 liczb całkowitych: $x1$, $x2$, .., $x10$, y ($0 \leq xi \leq 100$, $0 \leq y \leq 1000$). Oznacza on, iż w laboratorium znajduje się y skał, których pierwszy parametr ma wartość $x1$, drugi - $x2$, ..., dziesiąty - $x10$. Ponieważ tabelkę tworzyło kilku naukowców, wiersze z takimi samymi parametrami mogą się powtarzać. W kolejnym wierszu znajduje się liczba naturalna m ($1 \leq m \leq 10000$), oznaczająca liczbę pytań zadanych przez naukowców. W m następnych liniach wejścia znajdują się poszczególne pytania. Każde pytanie składa się z warunków, mających postać 10 liczb całkowitych: $a1$, $a2$, .., $a10$ ($-1 \leq ai \leq 100$). Naukowców interesuje liczebność skał spełniających wszystkie 10 warunków. Warunek ai=-1 oznacza, że wartość i-tego parametru może być dowolna. Natomiast ai<>-1 mówi, że i-ty parametr ma mieć wartość ai.

Wyjście

Dla każdego pytania naukowców, w osobnej linii wyjścia, należy wypisać liczbę skał, które spełniają warunki zadane przez naukowców.

Przykład

Dla danych:

```
5
10 1 1 4 5 6 7 8 9 1 100
1 2 4 4 5 5 5 5 5 1 200
2 2 4 4 6 6 6 8 5 2 300
10 2 1 9 5 5 5 5 5 3 400
10 2 1 9 5 5 5 5 5 3 500
3
10 -1 -1 9 -1 -1 -1 -1 -1 -1
-1 2 -1 -1 5 -1 -1 -1 -1 -1
1 2 4 4 -1 5 5 5 5 -1
```

poprawną odpowiedzią jest:

```
900
```

1100

200

Dźwięczne słowa

ID:1073

Limit czasu: 0.50 s

Limit pamięci: 16384 kB

Firma DB-Bit, główny potentat na rynku baz danych w Opsslandii, postanowiła rozszerzyć swój produkt o nowe możliwości wyszukiwania słów w tekście. Firmowi badacze, po przeanalizowaniu różnorodnych zbiorów tekstowych, postawili hipotezę, że najlepszym rozwiązaniem będzie oparcie nowej technologii na słowach, które mają więcej samogłosek (małe lub duże litery: 'a','e','i','o','u','y') niż spółgłosek i zawierają przynajmniej jedną spółgłoskę. Słowa o tej własności nazwali słowami dźwięcznymi. W Opsslandii spółgłoski zapisuje się używając tylko jednej litery.

W celu wykazania słuszności hipotezy, naukowcy muszą sprawdzić częstość występowania dźwięcznych słów w zadanym tekście. Naukowcy nie potrafią programować, a programiści firmy DB-Bit są zbyt obciążeni pracą. Pomóż ambitnym badaczom.

Zadanie

Napisz program, który wyznaczy liczbę dźwięcznych słów w podanej liście.

Wejście

W pierwszym wierszu znajduje się liczba słów N, $1 \le N \le 1000$. W kolejnych N wierszach znajdują się niepuste słowa składające się z co najwyżej 100 liter (dużych i małych) alfabetu angielskiego.

Wyjście

Na wyjściu powinna znaleźć się tylko jedna liczba określająca ilość dźwięcznych słów.

Przykład

Dla danych:

```
3
aBecadLo
OPSS
aaba
```

poprawną odpowiedzią jest:

```
1
```

Pantofelek

ID:1074

Limit czasu: 0.75 s

Limit pamięci: 16384 kB

Pantofelek sprawiedliwy (Paramecium iustus) to bardzo specyficzny gatunek pantofelka. Prowadzi stadny tryb życia i odżywia się bakteriami, które wchłania tylko wtedy, gdy może się nimi podzielić sprawiedliwie z innymi osobnikami w stadzie. Pantofelki sprawiedliwe jeśli zaczynają jeść, to zawsze wchłaniają wszystkie okoliczne bakterie, a następnie przemieszczają się w inne miejsce, w którym jest pokarm. O sprawiedliwym podziale bakterii mówimy wtedy, gdy wszystkie okoliczne bakterie można podzielić po równo pomiędzy wszystkie pantofelki. Wówczas każdy pantofelek wchłonie tę samą liczbę okolicznych bakterii i w okolicy nie będzie już żadnej. Zarówno pantofelki, jak i bakterie, rozmnażają się szybko i w bardzo specyficzny sposób. Co pewien czas, nagle u wszystkich pantofelków dochodzi do podziału komórkowego, w wyniku którego każdy osobnik dzieli się na X osobników potomnych (liczebność populacji wzrasta X razy). Okoliczne bakterie rozmnażają się w taki sam sposób.

Na podstawie historii rozmnożeń jednego pantofelka i jednej bakterii sprawdź, czy aktualny stan ich liczebności pozwala na sprawiedliwy podział. Jeśli taki podział istnieje, oblicz ile bakterii przypada na jednego pantofelka.

Wejście

Pierwsza linia zawiera liczbę zestawów danych D ($1 \le D \le 10$). W następnych liniach znajdują się kolejno po sobie opisy D zestawów danych. Pierwsza linia zestawu danych zawiera liczbę rozmnożeń N ($1 \le N \le 100000$). Każda z kolejnych N linii zestawu zawiera parę symbol-liczba, będącą opisem jednego rozmnożenia. Symbolem jest jedna z małych liter 'b' lub 'p' oznaczająca odpowiednio rozmnożenie się bakterii lub rozmnożenie się pantofelka. Liczba naturalna X w parze symbol-liczba oznacza, że liczebność osobników określonych przez symbol wzrasta X razy ($1 \le X \le 100000$). Symbol w parze jest oddzielony od liczby pojedynczą spacją.

Wyjście

Dla każdego zestawu danych, w osobnych liniach, wypisz w przypadku sprawiedliwego podziału liczbę bakterii przypadających na jednego pantofelka (liczba ta będzie zawsze mniejsza od 2^31) lub -1 jeśli bakterii nie da się podzielić sprawiedliwie.

Przykład

Dla danych wejściowych:

```
3
4
b 12
b 5
p 20
b 7
5
b 8
```

```
p 13
b 5
b 39
p 2
5
b 6
b 48
p 9
b 11
p 3
```

poprawną odpowiedzią jest:

```
21
60
-1
```

Optymalizator

ID:1075

Limit czasu: 2.00 s

Limit pamięci: 16384 kB

Centrum Badań Kosmicznych rozwija nowego rodzaju kompilatory, które mają być wyposażone w Optymalizator Procesów Sekwencyjnie Sumujących (OPSS). Inżynierowie postawili problem jasno - chcą "wyłapać" takie podciągi z ciągu liczb, których wyrazy po zsumowaniu dają określoną liczbę. Tobie oddano do zaprojektowania i zakodowania zadanie związane z procesem sumacyjnym.

Wejście

W pierwszej linii wejścia znajduje się liczba C, określająca liczbę zestawów danych, $1 \leq C \leq 10$. W kolejnych wierszach znajdują się zestawy danych. Każdy z C zestawów danych składa się z 2 linii. Pierwsza linia zawiera dwie liczby całkowite N, S, oddzielone pojedynczą spacją, gdzie N oznacza ile liczb bierze udział w sumowaniu, zaś S to suma jakiej oczekujemy ($1 \leq N \leq 500$, -2^31 < S < 2^31). W drugim wierszu zestawu znajduje się N liczb całkowitych, oddzielonych pojedynczą spacją: $a1$, $a2$, ..., an, $-500 \leq ai \leq 500$, dla i: $1 \leq i \leq N$.

Wyjście

Dla każdego zestawu danych, w osobnej linii wyjścia, należy wypisać słowo TAK, jeśli istnieje niepusty podciąg ciągu liczb $a1$, $a2$, ..., an, który daje zadaną sumę S, w przeciwnym razie należy wypisać słowo NIE.

Przykład

Dla wejścia:

```
2
5 32
1 2 5 10 25
5 -16
1 2 5 -10 -10
```

poprawną odpowiedzią jest:

```
TAK
NIE
```

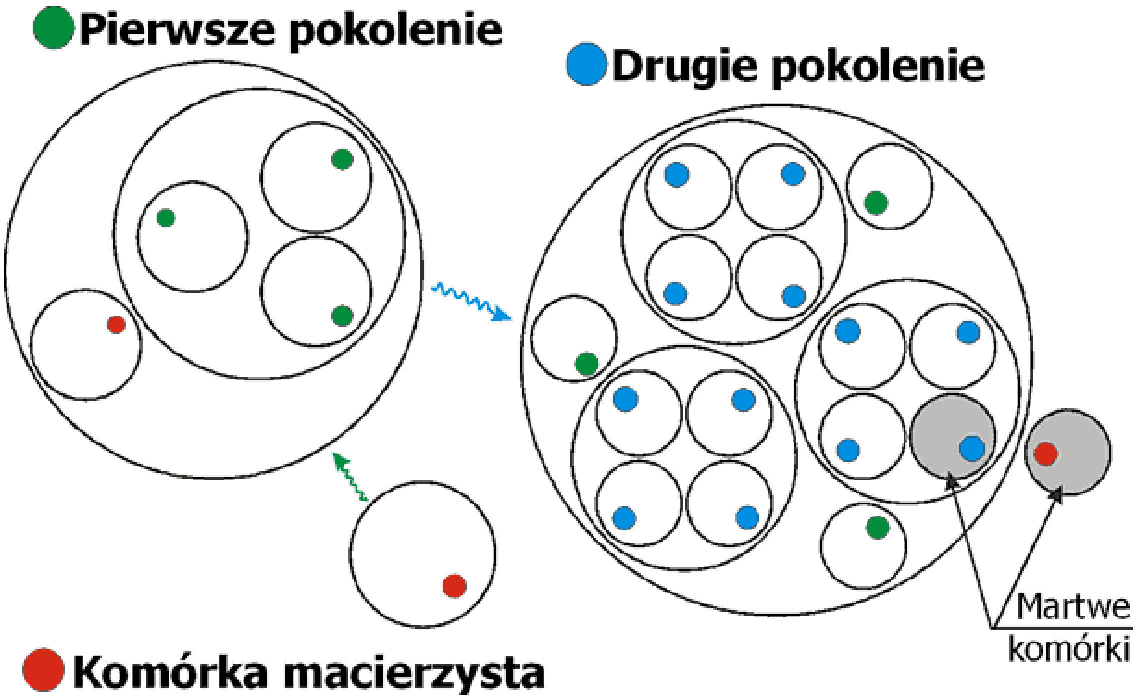
Hodowla alg

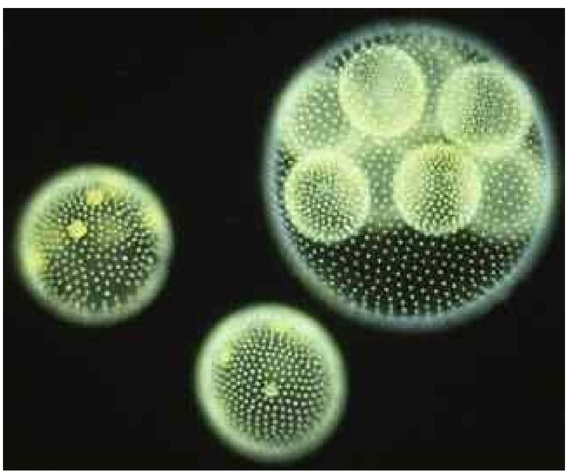
ID:1076

Limit czasu: 1.00 s

Limit pamięci: 16384 kB

Janek, starszy kolega Edka (znanego z hodowli chomików) z Wydziału Biologii, pracuje nad doktoratem. W tym celu musi hodować pewien rodzaj alg i oszacować tempo ich wzrostu. Początek hodowli wydawał się prosty. "Kolonia" alg umieszczona w akwarium po jednej dobie zwiększyła swoją objętość czterokrotnie, to znaczy, że młodych alg było 3 razy więcej niż ich "matek", ale po 5 dniach młodych było "prawie" 600 razy więcej niż na początku, zatem reguła czterokrotnego wzrostu okazała się fałszywa. Dokładniejsze badania wykazały, że komórki alg "matek" w ciągu doby pączkując dają życie czterem nowym komórkom, poprzednie pokolenie alg ("babki") ginąc wydzielają toksynę, która hamuje wzrost młodych w stosunku 1:1, to znaczy ginie tyle samo starych co nowych komórek. Na początku hodowli nie było co prawda starych alg, ale widocznie hodowla dopiero przystosowywała się do nowego środowiska stąd wolniejszy wzrost. Szczegóły rozwoju alg najlepiej prześledzić na rysunku. Ponieważ Janek, jak to biolog, nie jest zbyt sprawny w rachunkach, więc oszacowanie wielkości hodowli (np. po kilku miesiącach) przerasta jego możliwości. Pomóż ambitnemu badaczowi!





Zadanie

Należy oszacować "mnożnik", czyli liczbę komórek młodych alg wywodzących się z jednej komórki macierzystej, po zadanej liczbie dni.

Wejście

W pierwszym wierszu znajduje się liczba N, $0 < N \leq 10000$ oznaczająca liczbę zestawów danych. W każdym z kolejnych N wierszy znajduje się liczba dni hodowli D_i, $0 \leq D_i \leq 200000$.

Wyjście

Na wyjściu, w oddzielnych wierszach, należy wypisać dwie liczby całkowite N, C, oddzielone jedną spacją. N oznacza liczbę cyfr, z których składa się poszukiwany przez Janka mnożnik, a C 10 pierwszych cyfr mnożnika. Gdy $N \leq 10$ to należy wypisać dokładnie N cyfr.

Przykład

Dla danych:

```
3
1
70
16
```

prawidłowym rozwiązaniem jest:

```
1 3
40 8574848899
10 1117014753
```

Normy

ID:1077

Limit czasu: 0.75 s

Limit pamięci: 16384 kB

Naukowcy, znani już z zadania "Marsjańskie Skały", badają zebrane próbki skał z Marsa. Tym razem chcą zbadać każdą próbkę pod kątem spełniania rozmaitych norm zatwierdzonych przez Instytut do Badań Śladów Życia na Obcych Planetach. W tym celu dla każdej skały wyznaczają liczbę cząstek Ni znajdujących się w próbce. Próbka skały spełnia normę, jeśli liczba cząstek Ni w niej zawartych mieści się w przedziale domkniętym określonym przez daną normę.

Zadanie

Dla każdej zadanej normy wyznacz liczbę próbek skał, które ją spełniają.

Wejście

W pierwszym wierszu wejścia znajduje się liczba C, określająca liczbę zestawów danych ($1 \leq C \leq 3$). W kolejnych wierszach znajdują się opisy zestawów danych.

W pierwszej linii opisu zestawu znajduje się liczba N, oznaczająca liczbę posiadanych przez naukowców próbek ($1 \leq N \leq 100000$). W następnej linii zestawu znajduje się N liczb całkowitych: $a1$, $a2$,..., aN, opisujących zawartość cząstek Ni w kolejnych próbkach ($0 \leq ai < 2^{31}$, $1 \leq i \leq N$). W trzeciej linii opisu zestawu znajduje się liczba P, oznaczającą liczbę norm, na podstawie których próbki będą weryfikowane ($1 \leq P \leq 5000$). Kolejne P linii wejścia zawiera opis P norm.

Opis jednej normy składa się z dwóch liczb całkowitych: b_k, c_k ($0 \leq b_k \leq c_k < 2^{31}$, $1 \leq k \leq P$). Próbka skały a_i spełnia k-tą normę jeśli $b_k \leq a_i \leq c_k$.

Wszystkie liczby podane na wejściu w tym samym wierszu oddzielone są od siebie pojedynczą spacją.

Wyjście

Dla każdego zestawu, w osobnych liniach wyjścia, należy wypisać P liczb: $p_1, p_2, \ldots, p_P$, gdzie p_k ($1 \leq k \leq P$) równa jest liczbie próbek spełniających zadaną k-tą normę zestawu.

Przykład

Dla danych:

```
1
5
123 304 200604 12 3000
3
1 300
300 3000
5 5
```

poprawną odpowiedzią jest:

```
2
2
0
```

Wyrażenia

ID:1078

Limit czasu: 0.50 s

Limit pamięci: 16384 kB

Zadanie polega na obliczeniu wartości podanych wyrażeń.

Wejście

Pierwsza linia wejścia zawiera liczbę całkowitą Q ($1 \le Q \le 20$), oznaczającą liczbę zestawów danych. W następnych liniach opisane są kolejno po sobie zestawy danych. Pierwsza linia każdego zestawu zawiera liczbę całkowitą N ($1 \le N \le 10000$), będącą liczbą wyrażeń. W kolejnych N liniach zestawu danych opisanych jest N niepustych wyrażeń, po jednym w każdej linii.

Wyrażeniem jest:

1. pojedyncza liczba bez znaku
2. 2 wyrażenia połączone znakiem dodawania '+', znakiem odejmowania '-' lub znakiem mnożenia '*'
3. wyrażenie ujęte w nawiasy okrągłe
4. wyrażenie poprzedzone minusem, a następnie ujęte w nawiasy okrągłe
5. symbol oznaczający wartość jednego z wyrażeń podanych z zestawie danych ('w1' oznacza wartość pierwszego wyrażenia, 'w2' - wartość drugiego, 'w3' - trzeciego, itd. aż do N)

Wartość wyrażenia obliczana jest z zachowaniem kolejności wykonywania działań (w ogólnie przyjęty w matematyce sposób). Wartość dowolnego wyrażenia (również wchodzącego w skład innego wyrażenia) zawsze zawiera się w przedziale [-10^6, 10^6]. Wyrażenia nie zawierają białych znaków. Długość każdego wyrażenia jest mniejsza niż 256 znaków.

Wszystkie N wyrażeń można ustawić w takiej kolejności, że w każdym wyrażeniu, jeśli wystąpią symbole oznaczające wartości wyrażeń, to będą to wartości wyrażeń poprzednich.

Wyjście

W oddzielnych liniach należy wypisać wartości wszystkich wyrażeń ze wszystkich zestawów danych w tej samej kolejności w jakiej wyrażenia pojawiły się na wejściu.

Przykład

Dla danych wejściowych:

```
2
4
8*4
19
3+17-81+12
(-9)*(-3)
3
```

```
w2-1
5
(-w1*(2+w2))
```

poprawną odpowiedzią jest:

```
32
19
-49
27
4
5
-28
```

Transport

ID:1079

Limit czasu: 1.00 s

Limit pamięci: 16384 kB

Pewna firma świadczy usługi transportowe, dokładniej mówiąc, transportuje kontenery w Opsslandii. Opsslandia to kraj, w którym jest *N* miast, połączonych *K* drogami. Miasta ponumerowane są kolejnymi liczbami całkowitymi od 1 do *N*. Każde dwa różne miasta w tym kraju są połączone co najwyżej jedną drogą. Wszystkie drogi w Opsslandii są dwukierunkowe. Z każdego miasta da się dotrzeć do dowolnego innego miasta, ale zdarzyć się może, że nie będzie istniała pomiędzy nimi droga bezpośrednia i konieczny będzie przejazd przez inne miasta pośrednie. Na każdej drodze obowiązuje dzienne ograniczenie liczby przewożonych kontenerów. Koszt przewiezienia jednego kontenera pomiędzy dwoma bezpośrednio połączonymi miastami, czyli inaczej mówiąc: przejazd jednego kontenera jedną drogą, wynosi 1 opssar (opssar jest oficjalną walutą w Opsslandii). Na każdy kontener nie można wydać więcej niż 1 opssar dziennie (jeden kontener może być przetransportowany co najwyżej przez jedną drogę dziennie).

Twoje zadanie polega na obliczeniu minimalnej liczby dni potrzebnej na przetransportowanie wszystkich kontenerów z miasta nr 1 do miasta nr *N*, mając do dyspozycji określone fundusze przeznaczone na ten cel.

Wejście

Pierwsza linia zawiera liczbę całkowitą D ($1 \leq D \leq 20$), określającą liczbę zestawów danych. W następnych liniach opisane są kolejno po sobie zestawy danych. W pierwszej linii zestawu danych znajdują się dwie liczby całkowite: liczba miast N ($1 \leq N \leq 100$) oraz liczba dróg K. W każdej z kolejnych K linii zestawu danych znajdują się trzy liczby całkowite A, B i C: A i B ($1 \leq A, B \leq N$) są numerami miast, pomiędzy którymi istnieje droga, przez którą dziennie nie można przetransportować więcej niż C kontenerów ($1 \leq C \leq 100$). Ostatnia linia zestawu danych zawiera dwie liczby całkowite: liczbę T będącą liczbą kontenerów, które należy przetransportować z miasta nr 1 do miasta nr N oraz liczbę opssarów F przeznaczoną na transport kontenerów ($1 \leq T \leq$ 10^7; $1 \leq F \leq$ 2^31-1).

Wyjście

W oddzielnych liniach dla każdego zestawu danych należy wypisać minimalną liczbę dni potrzebną do przetransportowania wszystkich kontenerów wykorzystując przy tym nie więcej niż przeznaczoną na ten cel liczbę opssarów. Jeśli przeznaczone fundusze są niewystarczające do przetransportowania wszystkich kontenerów, wówczas należy zamiast minimalnej liczby dni wypisać -1.

Przykład

Dla danych wejściowych:

```
1
4 5
1 2 5
1 3 2
2 3 1
```

```
3 4 10
2 4 3
11 23
```

poprawną odpowiedzią jest:

```
3
```

Dzień Dziecka

ID:1080

Limit czasu: 0.50 s

Limit pamięci: 32768 kB

W szkole Jasia, z okazji Dnia Dziecka, nauczyciele zamiast lekcji postanowili zorganizować różnego rodzaju zabawy edukacyjne.

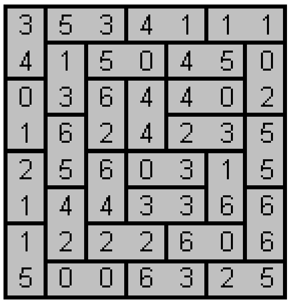
Na zajęcia z matematyki nauczycielka przyniosła domino. Pokazała kilka łamigłówek związanych z tą grą, z których jedna była na tyle trudna, że nawet Jaś, który mógł pochwalić się samodzielnym rozwiązaniem zadania "Domino" z Zawodów Noworocznych, nie potrafił jej rozwiązać. Nauczycielka obiecała, że ten, kto przyniesie rozwiązanie zagadki, otrzyma w nagrodę "szóstkę". Pomóż Jasiowi zdobyć ocenę celującą.

Zadanie

Dana jest plansza o wymiarach 8x7, podzielona na 56 identycznych kwadratów oraz zestaw 28 różnych kamieni domina. Na każdym kwadracie (polu) znajduje się jedna liczba całkowita z przedziału [0..6]. Należy przykryć tę planszę jednym zestawem kamieni domina tak, aby liczby znajdujące się na dwóch sąsiednich, przykrywanych polach były równe liczbom znajdującym się na połówkach kamienia. Kamienie nie mogą na siebie zachodzić, ani leżeć jeden na drugim.

Nie zawsze przykrycie planszy zestawem kamieni domina jest możliwe, często jednak jest wiele możliwości by to zrobić. Twoim zadaniem jest stwierdzenie dla zadanego opisu planszy, na ile sposobów można pokryć planszę kamieniami z jednego zestawu domina oraz podać pierwsze leksykograficznie pokrycie spełniające warunki zadania.

3	5	3	4	1	1	1
4	1	5	0	4	5	0
0	3	6	4	4	0	2
1	6	2	4	2	3	5
2	5	6	0	3	1	5
1	4	4	3	3	6	6
1	2	2	2	6	0	6
5	0	0	6	3	2	5



Przykładowa plansza oraz ułożone na niej kamienie domina

Wejście

Wejście zawiera opis planszy. Opis złożony jest z 8 wierszy, w których występuje po 7 liczb całkowitych (oddzielonych pojedynczą spacją) z przedziału [0..6]. *J*-ta liczba w *i*-tym wierszu wejścia określa liczbę jaka stoi w *i*-tym wierszu, licząc od góry i *j*-tej kolumnie planszy.

Wyjście

W pierwszej linii wyjścia należy wypisać liczbę możliwych ustawień kamieni domina, tak aby pokryć planszę. W następnych liniach wyjścia należy podać opis pierwszego leksykograficznie pokrycia spełniającego warunki zadania (o ile takie istnieje).

Opis pokrycia zawiera 8 wierszy, w których występuje po 7 liczb naturalnych od 1 do 28 (oddzielonych pojedyncza spacja). *J*-ta liczba w i-tym wierszu opisu pokrycia oznacza numer kamienia jaki leży na polu planszy w *i*-tym wierszu i *j*-tej kolumnie w tym pokryciu.

Kamienie numerujemy w następujący sposób:

```
kamień  | numer
0:0     | 1
0:1     | 2
0:2     | 3
...
1:1     | 8
1:2     | 9
...
6:6     | 28
```

Porządek leksykograficzny opisów pokryć oznacza porządek leksykograficzny ciągów uzyskanych przez zapisanie opisu pokrycia wierszami, rozpoczynając od pierwszego górnego wiersza.

Przykład

Dla danych:

```
3 5 3 4 1 1 1
4 1 5 0 4 5 0
0 3 6 4 4 0 2
1 6 2 4 2 3 5
2 5 6 0 3 1 5
1 4 4 3 3 6 6
1 2 2 2 6 0 6
5 0 0 6 3 2 5
```

poprawną odpowiedzią jest:

```
5
20 21 21 11 11 8 8
20 10 6 6 24 24 3
2 10 18 23 5 5 3
2 27 18 23 15 15 26
9 27 25 4 4 13 26
9 16 25 19 19 13 28
12 16 14 14 7 7 28
12 1 1 22 22 17 17
```

Komputerowa telepatia 3D

ID:1081

Limit czasu: 2.00 s

Limit pamięci: 1024 kB

Andrzej i Bartek, bawiąc się wymyślone przez siebie zabawy (Komputerowa telepatia, Kąt obrotu), zdobyli tyle doświadczenia w zgadywaniu zeskanowanych symboli, że robili to za każdym razem bezbłędnie. Postanowili więc po raz kolejny utrudnić sobie zadanie. Ponieważ kupili niedawno skaner do przedmiotów trójwymiarowych, postanowili, że tym razem będą starać się rozpoznać, czy skan przedmiotu przedstawia elipsoidę, prostopadłościan, czy też ostrosłup prawidłowy czworokątny. Skaner ten skanuje wycinek przestrzeni będący prostopadłościanem. Ustalili, że skanowane przedmioty będą ułożone w przestrzeni w specyficzny sposób: osie symetrii elipsoidy oraz krawędzie prostopadłościanu i krawędzie podstawy ostrosłupa będą ułożone tylko prostopadle lub równolegle do krawędzi prostopadłościanu będącego całym zeskanowanym wycinkiem przestrzeni.

Zadanie

Należy zidentyfikować zeskanowany obiekt i wypisać jego numer: 1 - prostopadłościan, 2 - elipsoida, 3 - ostrosłup.

Wejście

W pierwszym wierszu znajduje się liczba C ($1 \leq C \leq 8$), oznaczająca liczbę zestawów danych. Każdy zestaw składa się z dwóch wierszy.

W pierwszym wierszu zestawu znajdują się trzy liczby W, H i D odzielone od siebie pojedynczą spacją. W oznacza liczbę cyfr szesnastkowych przypadającą na szerokość obrazu ($2 \leq W \leq 100$). Szerokość obrazu liczona w pikselach jest równa $4*W$. H oznacza wysokość obrazu liczoną w pikselach ($6 \leq H \leq 100$), a D głębokość obrazu liczoną w pikselach ($6 \leq D \leq 100$).

W drugim wierszu zestawu znajduje się $W*H*D$ znaków ze zbioru [0,1,..9,A,B,C,D,E,F], oznaczających liczby szesnastkowe, którymi została zakodowana figura. Każda cyfra szesnastkowa oznacza 4 punkty obrazu (piksele). W rozwinięciu binarnym 1 oznacza kolor czarny (zeskanowaną bryłę), 0 to białe tło. Pierwsze $W*H$ znaków opisuje płaszczyznę na głębokości 1, kolejne $W*H$ znaków płaszczyznę na głębokości 2 i tak aż do głębokości D. Opis płaszczyzny zaczyna się od lewego dolnego rogu. Figury opisane na płaszczyznach są spójne (nie maja żadnych dziur). Opis skanu jednoznacznie identyfikuje bryłę.

Wyjście

Dla każdego zestawu danych, w osobnym wierszu wyjścia, należy wypisać cyfrę 1, 2 lub 3 identyfikującą zeskanowany obiekt.

Przykład

TEST 0:

```
3
11 11 11
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

```
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

```
11 11 11
```

```
00000000000000000000000000FFE0000000FFFFE000003FFFFF800007FFFFFC00003FFFFF800000
FFFFE0000000FFE00000000000000000000000000000000000000003FFF8000007FFFFFC0001FFFF
FFF0003FFFFFFF8007FFFFFFFC003FFFFFFF8001FFFFFFF00007FFFFFC000003FFF8000000000000
000000FFE0000007FFFFFC0003FFFFFFF800FFFFFFFFE01FFFFFFFFF03FFFFFFFFF81FFFFFFFFF00
FFFFFFFFE003FFFFFFF80007FFFFFC000000FFE0000000FFFFE00001FFFFFFF000FFFFFFFFE03FFF
FFFFFF87FFFFFFFFFC7FFFFFFFFFC7FFFFFFFFFC3FFFFFFFFF80FFFFFFFFE001FFFFFFF00000FFFF
E000003FFFFF80003FFFFFFF801FFFFFFFFF07FFFFFFFFFCFFFFFFFFFFEFFFFFFFFFFEFFFFFFFFFF
E7FFFFFFFFFC1FFFFFFFFF003FFFFFFF80003FFFFF800007FFFFFC0007FFFFFFFC03FFFFFFFFF87F
FFFFFFFFCFFFFFFFFFFEFFFFFFFFFFEFFFFFFFFFFE7FFFFFFFFFC3FFFFFFFFF807FFFFFFFC0007FF
FFFC00003FFFFF80003FFFFFFF801FFFFFFFFF07FFFFFFFFFCFFFFFFFFFFEFFFFFFFFFFEFFFFFFFF
FFE7FFFFFFFFFC1FFFFFFFFF003FFFFFFF80003FFFFF800000FFFFE00001FFFFFFF000FFFFFFFFE0
3FFFFFFFFF87FFFFFFFFFC7FFFFFFFFFC7FFFFFFFFFC3FFFFFFFFF80FFFFFFFFE001FFFFFFF00000
FFFFE0000000FFE0000007FFFFFC0003FFFFFFF800FFFFFFFFE01FFFFFFFFF03FFFFFFFFF81FFFFF
FFFF00FFFFFFFFE003FFFFFFF80007FFFFFC000000FFE0000000000000000003FFF8000007FFFFFC
0001FFFFFFF0003FFFFFFF8007FFFFFFFC003FFFFFFF8001FFFFFFF00007FFFFFC000003FFF80000
000000000000000000000000000000000000FFE0000000FFFFE000003FFFFF800007FFFFFC00003F
FFFF800000FFFFE0000000FFE00000000000000000000000000
```

```
11 11 11
```

```
00000000000000000000000000000000000000000000000000000000000040000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000040000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000E0000000000E0000000000E000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000E0000000000E0000000000E000000000000000000000000000000000000000000000
000000000000000000000000000000000000000001F0000000001F0000000001F0000000001F0000
000001F00000000000000000000000000000000000000000000000000000000000000000000000000
001F0000000001F0000000001F0000000001F0000000001F00000000000000000000000000000000
000000000000000000000000000000003F800000000003F800000000003F800000000003F800000000003F80
00000003F800000000003F800000000000000000000000000000000000000000000000000003F80000
00003F800000000003F800000000003F800000000003F800000000003F800000000003F8000000000000000000
000000000000000000000007FC000000007FC000000007FC000000007FC000000007FC000000007F
C000000007FC000000007FC000000007FC0000000000000000000000000000007FC000000007FC00
0000007FC000000007FC000000007FC000000007FC000000007FC000000007FC000000007FC00000
00000000000000FFE00000000FFE00000000FFE00000000FFE00000000FFE00000000FFE00000000
FFE00000000FFE00000000FFE00000000FFE00000000FFE0000
```

Dla wejścia zawartego w pliku (test0, test0.zip) poprawną odpowiedzią jest:

```
1
```

```
2
```

```
3
```

Zabawa

ID:1082

Limit czasu: 1.50 s

Limit pamięci: 32768 kB

Jeden z bardzo znanych matematyków Opsslandii, Matimus Logaritmus, ma synka Arcusa. Arcus już od małego uwielbia matematyczne gry i łamigłówki.

Ostatnio polubił taką zabawę: wymyśla pewną nieujemną liczbę całkowitą $X1$ i dodaje do niej liczbę powstałą przez zapisanie cyfr liczby $X1$ w odwrotnej kolejności. Otrzymuje w ten sposób liczbę $X2$. Dalej postępuje analogicznie, czyli do liczby $X2$ dodaje liczbę powstałą przez "odwrócenie" cyfr $X2$ otrzymując kolejną liczbę $X3$, itd, itd...

Arcus mógłby bawić się tak w nieskończoność, więc jego tata Matimus postanowił nieco skomplikować chłopcu zabawę. Zapytał go, jaką najmniejszą liczbę należy pomyśleć na samym początku, aby w wyniku zabawy otrzymać w pewnym momencie wskazaną liczbę Z. Arcus myśli już bardzo długo nad odpowiedzią, pomóż mu rozwiązać zagadkę.

Wejście

Pierwsza linia wejścia zawiera liczbę całkowitą N ($1 \leq N \leq 100$), oznaczającą liczbę zestawów danych. W kolejnych N liniach znajdują się zestawy danych, po jednym w każdej linii. Jeden zestaw składa się z jednej liczby całkowitej Z ($0 \leq Z \leq 10$^9).

Wyjście

Dla każdego zestawu danych w oddzielnych liniach, należy wypisać najmniejszą nieujemną liczbę całkowitą X, mniejszą od Z, od której Arcus powinien rozpocząć zabawę tak, aby w trakcie zabawy otrzymać zadaną liczbę Z. W przypadku, gdy taka liczba nie istnieje należy wypisać -1.

Przykład

Dla danych wejściowych:

```
4
121
976071668
5104
187876
```

Poprawnym wynikiem jest:

```
7
1
184
-1
```

Pikoboty

ID:1083

Limit czasu: 1.50 s

Limit pamięci: 32768 kB

Jest rok 3749. Rasa ludzka wymyśliła sztuczny mechanizm obronny przeciwko wirusom infekującym genotyp.

Genotyp to łańcuch genów. Każdy gen jest kodowany w postaci dwuznaku składającego się z liter 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J'. Zdrowy genotyp złożony jest z genów o takim samym kodzie genetycznym P. Na przykład, jeśli P jest genem o kodzie 'AA', to zdrowy genotyp o długości 3 jest postaci 'AA AA AA'. Wirus atakujący genotyp może zmienić kod jednego lub więcej genów zdrowego genotypu na inny kod, różny od P. Chory gen nie może być już ponownie zmieniony przez wirusa. Wszystkie geny ze zmienionym kodem stanowią genotyp wirusa. Przykładowo, zainfekowany genotyp z poprzedniego przykładu może wyglądać następująco 'AB AC AA' (genotyp wirusa 'AB AC'). Każde X kolejnych genów genotypu wirusa tworzy bakterię o długości X. Bakterie o tej samej długości nie mają wspólnych genów. W genotypie wirusa mogą kryć się bakterie różnych długości. Na przykład, genotyp wirusa 'AB AC' zawiera dwie bakterie o długości 1: 'AB','AC' i jedną o długości 2: 'AB AC'.

Jedyną znaną bronią przeciwko wirusowi są pikoboty. Pikobot jest robotem o bardzo małych rozmiarach, którego zadaniem jest przywrócenie chorym genom pierwotnego kodu genetycznego P. Pikobot działa według programu, który ma postać sekwencji kodów genetycznych.

Pikobot jest "wstrzykiwany" do zainfekowanego genotypu i zaczynając od pierwszego genu genotypu wirusa, porównuje ciąg genów genotypu wirusa z sekwencją genów zapisaną w swoim programie. Jeśli oba ciągi są identyczne, wówczas pikobot przywraca genom w ciągu ich pierwotną postać (niszczy bakterię) i przechodzi do kolejnej sekwencji. W przeciwnym razie pikobot przechodzi do kolejnej sekwencji genów w genotypie wirusa (przesuwa się o liczbę genów równą długości swojego programu).

Natychmiast po analizie ostatniego genu pikobot wysyła do głównej bazy danych liczbę zniszczonych przez siebie bakterii. Zniszczenie bakterii następuje wtedy, gdy wszystkie geny bakterii zostaną wyleczone. Pikobot, którego program ma długość X może niszczyć tylko bakterie o długości X.

Zadanie

Znając kod genetyczny genu pierwotnego P, postać genotypu zainfekowanego, dlugości programów pikobota (równe długościom bakterii) i wiedząc, że długość genotypu wirusa nie przekracza 100000, znaleźć program najlepszego pikobota, czyli takiego, który wyleczy największą część "chorego" genotypu. Jeżeli wybór jest niejednoznaczny, należy spośród wcześniej wybranych pikobotów, wybrać tego, który zniszczył najwięcej bakterii. Jeśli to nie wyłoni zwycięzcy, wówczas zamiast programu należy podać 0. Ponadto jeżeli długość zainfekowanego genotypu nie przekracza 10000 lekarze chcieliby wiedzieć, na ile różnych sposobów wirus może zmienić zdrowy genotyp, wykorzystując kody genetyczne wchodzące w skład genotypu wirusa.

Wejście

W pierwszej linii wejścia znajduje się liczba $0< N \leq 1000000$, określająca długość zainfekowanego genotypu oraz po pojedynczej spacji kod genetyczny genu pierwotnego P. W drugim wierszu znajduje się N kodów genetycznych oddzielonych pojedynczą spacją, które stanowią opis zmienionego genotypu. Liczba $0< D\leq 40$ w kolejnym wierszu określa ilość długości programów pikobota (równym długościom bakterii). W ostatnim wierszu wejścia znajdują się liczby $1\leq d_i\leq 40$, gdzie $0< i\leq D$ opisujące długości bakterii.

Wyjście

W pierwszej linii wyjścia powinien zostać wypisany program najlepszego pikobota. Jeżeli długość genotypu nie przekracza 10000 w kolejnym wierszu należy wypisać liczbę różnych sposobów modyfikacji zdrowego genotypu (co najmniej jeden gen musi się różnić) przez wirusa. W przypadku, gdy liczba ta będzie miała więcej niż 50 cyfr wystarczy wypisać 50 ostatnich.

Przykład

Dla danych:

```
7 AA
BC BC FD GF FG FD FD
3
1 2 3
```

poprawnym rozwiązaniem jest:

```
FD
78124
```

Płyty

ID:1084

Limit czasu: 0.50 s

Limit pamięci: 4096 kB

Mieszkańcy Opsslandii planują budowę Instytutu Badań nad Figurami Foremnymi (w skrócie IBFF). Ma to być przestronny budynek o podstawie kwadratowej, zbudowany na specjalnych kwadratowych płytach o rozmiarach a x a metrów, sprowadzonych z odległych zakątków kraju. Płyty mają przylegać do siebie bocznymi krawędziami i formować kwadrat - podobnie jak na szachownicy. Niestety część płyt uległa zniszczeniu podczas transportu i dowieziono tylko k płyt. Mieszkańcy Opsslandii bardzo się tym faktem zmartwili, bo płyt nie wystarczy na zbudowanie Instytutu zakładanej wielkości. Dowiezienie płyt też nie wchodzi w grę, ponieważ transport płyt trwa bardzo długo i jest niezmiernie kosztowny. Postanowili zatem, że zbudują największy budynek z dowiezionych płyt zachowując kwadratową podstawę Instytutu.

Pomóż mieszkańcom Opsslandii ustalić maksymalną możliwą długość ściany budynku, wiedząc że płytek nie można ciąć.

Wejście

W pierwszej linii podana jest jedna liczba całkowita C, oznaczająca liczbę zestawów danych wejściowych ($1 \le C \le 5000$). W linii i+1 (i = 1, 2, ..., C) podane są dwie liczby całkowite: k, oznaczająca ilość płyt ($1 \le k \le 2^{31}$-1) oraz a, oznaczająca długość boku płyty ($1 \le a \le 2^{31}$-1).

Wyjście

Dla każdego zestawu, w osobnych liniach wyjścia, powinna pojawić się jedna nieujemna liczba całkowita, oznaczająca maksymalną możliwą długość ściany budynku, wyrażoną w metrach.

Przykład

Dla danych wejściowych:

```
2
10 3
4 5
```

poprawną odpowiedzią jest:

```
9
10
```

Starożytna maszyna

ID:1085

Limit czasu: 0.50 s

Limit pamięci: 4096 kB

Dawno, dawno temu, w pewnej krainie słynącej z piramid, archeolodzy odkryli ciekawą starożytną maszynę, która potrafiła obliczyć dowolną liczbę Fibonacciego. Dane wprowadzane do maszyny oraz wyniki jej obliczeń, były liczbami zapisanymi w systemie dwójkowym. Znalezisko bardzo przydałoby się mieszkańcom krainy, ale niestety miało jeden mankament: z powodu długowieczności urządzenia ostatnia cyfra wyniku była nieczytelna. Przez tę wadę maszyna była bezużyteczna.

Napisz program symulujący działanie maszyny przy założeniu, że n-ta liczba Fibonacciego **fib(*n*)** zdefiniowana jest następująco:

fib(1)=1,

fib(2)=1,

fib(*n*)=fib(*n*-1)+fib(*n*-2), *n*>2.

Program nie musi drukować wszystkich cyfr liczby Fibonacciego - wystarczy, że poda najmniej znaczącą (ostatnią) jej cyfrę.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita C, określająca liczbę zestawów danych ($1 \leq C \leq 100$). W i+1 wierszu (i = 1, 2, ..., C) znajduje się liczba b zapisana w systemie dwójkowym, składająca się z co najmniej 1 cyfry i co najwyżej 255 cyfr. Pierwszą (najbardziej znaczącą) cyfrą w zapisie liczby b jest zawsze 1.

Wyjście

Dla każdego zestawu danych, program powinien wypisać na wyjściu najmniej znaczącą (ostatnią) cyfrę w zapisie dwójkowym liczby **fib(*b*).**

Przykład

Dla danych wejściowych:

```
2
111
1001
```

poprawną odpowiedzią jest:

```
1
0
```

Licytacja

ID:1086

Limit czasu: 0.50 s

Limit pamięci: 8192 kB

Dwóch matematyków Pi i Sigma spotkało się wieczorem na partyjce pokera. Ponieważ nie mieli pieniędzy, postanowili grać "na zapałki". Pi miał przy sobie tylko niebieskie zapałki, a Sigma tylko zielone. Przed przystąpieniem do licytacji Pi wyłożył a swoich niebieskich zapałek, zaś Sigma dołożył b swoich zielonych. Pi zaczyna licytację. Licytują na zmianę. W każdym kroku licytacji zawodnik dorzuca tyle swoich zapałek, ile aktualnie jest zapałek przeciwnika w puli. Licytacja trwała dość długo i gracze nie mogli się doliczyć aktualnej liczby zapałek w puli. Wiedzieli jednak, ile razy każdy z nich dorzucał zapałki podczas licytacji.

Zadanie

Pomóż matematykom określić, ile zapałek jest w puli po n krokach licytacji. Ponieważ to matematycy, zadowoli ich znajomość reszty z dzielenia tej liczby (liczby zapałek po n krokach) przez ustaloną przez nich "magiczną" liczbę m.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych C ($1 \leq C \leq 1000$). W kolejnych wierszach wejścia znajdują się zestawy danych. Każdy z C zestawów danych składa się z jednego wiersza zawierającego liczby całkowite a, b, n, m, oddzielone pojedynczą spacją ($0 < a, b <$ 2^31, $0 \leq n <$ 2^31, $2 \leq m <$ 2^31). Liczby a, b określają kolejno liczbę niebieskich zapałek Pi oraz zielonych Sigmy znajdujących się w puli przed rozpoczęciem licytacji. Liczba n określa liczbę kroków licytacji. Liczba m to ustalona przez graczy "magiczna" liczba.

Wyjście

Dla każdego zestawu danych, wynikiem jest linia zawierająca jedną liczbę - resztę z dzielenia przez m liczby zapałek po n krokach licytacji, przy założeniu, że przed przystąpieniem do licytacji w puli znajduje się a niebieskich zapałek Pi oraz b zielonych zapałek Sigmy.

Przykład

Dla danych wejściowych:

```
3
1 1 4 1000
5 6 7 1000
4 101 23 999
```

poprawną odpowiedzią jest:

```
13
309
767
```

Multyplikator

ID:1087

Limit czasu: 1.50 s

Limit pamięci: 8192 kB

Pewna firma informatyczna opracowała superszybką metodę mnożenia liczb całkowitych. Przed praktycznym zastosowaniem metoda musi zostać przetestowana. W pierwszym etapie testowania sprawdzana będzie poprawność liczby cyfr w zapisie dziesiętnym iloczynu. Zgodnie z przyjętą polityką firmy, program testujący nie może być dziełem jej programistów. Firma postanowiła zlecić to zadanie Tobie.

Zadanie

Napisz program, który dla zadanej listy czynników, wyznaczy liczbę cyfr w zapisie dziesiętnym iloczynu.

Wejście

W pierwszym wierszu wejścia znajduje się liczba zestawów danych C, $1 \leq C \leq 5000$. Każdy zestaw danych składa się z dwóch wierszy: w pierwszym znajduje się liczba naturalna N, oznaczająca liczbę czynników badanego iloczynu, $1 \leq N < 1000$, natomiast w drugim podanych jest N liczb całkowitych, oddzielonych pojedynczą spacją, będących czynnikami iloczynu. Każdy czynnik jest liczbą składającą się z co najwyżej trzech cyfr.

Wyjście

Dla każdego zestawu danych, w osobnych liniach wyjścia, należy wypisać liczbę cyfr, z których składa się zapis dziesiętny iloczynu.

Przykład

Dla następujących danych:

```
2
2
2 5
5
10 10 10 10 5
```

poprawną odpowiedzią jest:

```
2
5
```

Klocki

ID:1088

Limit czasu: 0.50 s

Limit pamięci: 8192 kB

Mały Jasio dostał ostatnio od rodziców prezent w postaci 66 klocków. Każdy klocek zbudowany jest z dwóch części, z których każda zawiera pewną liczbę oczek, podobnie jak jest to w przypadku kostek domina. Liczba oczek znajdujących się na każdej połówce klocka wynosi co najmniej jeden i co najwyżej jedenaście:

```
1 | 1
1 | 2
..
1 | 11
2 | 2
2 | 3
..
2 | 11
..
11 | 11
```

Bawiąc się klockami, Jasio wymyślił pewną grę polegającą na układaniu klocków w linii prostej według następujących zasad:

- Wybrać pewną początkową liczbę oczek P
- Ułożyć obok siebie w linii prostej (klocki można obracać o 180 stopni), od lewej strony, klocki tak, aby:
    - dla każdego klocka, jego prawy sąsiad (klocek) miał na lewej połowie tyle oczek ile ma ten klocek na swojej prawej połowie
    - każda liczba dostępnych oczek, wystąpiła w tym ciągu klocków dokładnie dwa razy
    - liczba oczek na lewej połowie pierwszego klocka i prawej połowie ostatniego klocka były takie same i wynosiły P

Każdemu klockowi Jasio przyporządkował pewną liczbę punktów i chciałby wiedzieć jaka jest minimalna oraz maksymalna możliwa liczba punktów do zdobycia w grze. Do gry, Jasio może używać 6, 10, 15, 21, 28, 36, 45, 55 lub 66 klocków, wówczas wykorzystuje klocki z maksymalnie odpowiednio 3, 4, 5, 6, 7, 8, 9, 10, 11 oczkami na każdej z połówek.

Przykładowo, zestaw 15 klocków przedstawia się następująco:

```
1 | 1
1 | 2
..
1 | 5
```

```
2 | 2
2 | 3
..
2 | 5
..
5 | 5
```

Zadanie

Znając początkową liczbę oczek P oraz punktację dla kolejnych klocków, wyznacz minimalną i maksymalną liczbę punktów jaką Jasio może uzyskać w swojej grze.

Wejście

W pierwszym wierszu wejścia znajduje się liczba zestawów danych C, $1 \leq C \leq 3$. W kolejnych wierszach wejścia znajdują się zestawy danych. Każdy z C zestawów danych składa się z trzech wierszy. W pierwszej linii zestawu znajduje się liczba N (możliwe wartości: 6, 10, 15, 21, 28, 36, 45, 55, 66), określająca liczbę klocków, które Jasio wykorzystuje do gry. Drugą linię zestawu stanowi N oddzielonych pojedynczą spacją liczb całkowitych z przedziału <-1000; 1000>, określających punktację klocków (klocki w kolejności odpowiednio: 1 | 1, 1 | 2, .. 2 | 2, 2 | 3, ..). W ostatniej linii zestawu znajduje się wybrana przez Jasia początkowa liczba oczek P, $1 \leq P \leq 11$.

Wyjście

Dla każdego zestawu danych, w osobnych liniach wyjścia, należy wypisać minimalną i maksymalną liczbę punktów jaką może uzyskać Jasio. Liczby powinny być oddzielone pojedynczą spacją. W przypadku, gdy nie istnieje ciąg klocków spełniający zasad gry należy przyjąć, że maksimum jest równe minimum i wynosi 0.

Przykład

Dla danych:

```
1
10
1 2 3 4 5 6 7 8 9 10
3
```

poprawnym rozwiązaniem jest:

```
20 21
```

Wzorce bitowe

ID:1089

Limit czasu: 0.50 s

Limit pamięci: 16384 kB

Wzorcem bitowym nazywamy dowolny ciąg zerojedynkowy długości n zawierający k jedynek, $1 \leq k \leq n$. Dla ustalonej długości n i liczby jedynek k, rozpatrzmy wszystkie wzorce bitowe posortowane malejąco w porządku leksykograficznym. Jaką będzie miał postać wzorzec bitowy znajdujący się na pozycji *d?*. Zakładamy, że liczba d jest tak dobrana, że taki wzorzec istnieje.

Wejście

W pierwszym wierszu znajduje się liczba zestawów danych C, $1 \leq C \leq 500$. Każdy zestaw danych składa się z dwóch wierszy. W pierwszym wierszu znajdują się dwie liczby naturalne n i k, $1 \leq k \leq n \leq 100$ oddzielone pojedynczą spacją. Oznaczają odpowiednio długość wzorca i liczbę jedynek. W drugim wierszu podana jest pozycja d, $1 \leq d \leq 2^{31}$-1 szukanego wzorca.

Wyjście

Dla każdego zestawu danych na wyjściu należy wypisać wzorzec bitowy znajdujący się na pozycji d.

Przykład

Dla następujących danych:

```
2
3 3
1
2 1
2
```

poprawną odpowiedzią jest:

```
111
01
```

Generis

ID:1090

Limit czasu: 1.50 s

Limit pamięci: 16384 kB

W dawnych czasach, starożytna cywilizacja Bajteków do przewidywania zjawisk astronomicznych, używała maszyny o nazwie Generis, która wyznaczała mozolnie kolejne wyrazy pewnego rosnącego ciągu liczb. Maszyna była konfigurowalna, a na jej konfigurację składały się dwie liczby całkowite dodatnie, tzw. pierwsza i druga liczba konfiguracji. Pierwszym wyrazem ciągu generowanego przez maszynę była suma liczb z jej konfiguracji, drugim wyrazem, suma pierwszego wyrazu oraz drugiej liczby konfiguracji, natomiast od trzeciego wyrazu ciągu, każdy wyraz był sumą dwóch poprzednio wygenerowanych wyrazów.

Przy odpowiedniej konfiguracji maszyna potrafiła generować ciąg Fibonacciego (co prawda nie od pierwszego elementu) i był to w zasadzie jedyny ciąg, który znalazł u Bajteków praktyczne zastosowanie. Konfigurowalność maszyny nie była zbyt przydatna Bajtekom, a przynajmniej nie mieli pojęcia jak ją dobrze wykorzystać. Potrafili dostrzec wszechobecność ciągu Fibonacciego w przyrodzie i za pomocą maszyny dokonywali m. in. przewidywania pogody. Konfigurowaniem maszyny zajmowali się jedynie naukowcy, próbujący co jakiś czas odnaleźć praktyczne zastosowanie innych ciągów generowanych przez maszynę.

Aby odkryć zastosowania innych ciągów generowanych przez maszynę naukowcy potrzebowali grantów na badania nad nią. Postanowili zorganizować prezentację dla możnych Kraju Bajteków, aby pokazać im działanie maszyny i wyjaśnić jakie można by uzyskać z niej korzyści, gdyby badania nad ciągami generowanymi przez maszynę przyniosły efekty.

Kierujący badaniami profesor Hexos postanowił, że maszyna wygeneruje takie ciągi, w których będą występowały pewne bardzo szczególne dla matematycznej kultury Bajteków liczby. Każdy pokaz będzie polegał na odpowiednim skonfigurowaniu maszyny i wykonywaniu obliczeń, aż do momentu uzyskania pewnej zadanej i podanej publiczności przed konfiguracją liczby. Profesor chce, aby obliczenia były efektowne, oraz aby wyglądały na jak najbardziej skomplikowane (wywarcie takiego wrażenia na publiczności ma ułatwić pozyskanie grantu).

Profesor nie ma czasu aby dla każdej liczby, jaką chce uzyskać w kolejnych pokazach prezentacji, wyznaczyć taką konfigurację maszyny, aby dokonywała ona obliczeń możliwie długo. To zadanie zlecił swojemu asystentowi, czyli Tobie!

Wejście

W pierwszym wierszu występuje jedna liczba całkowita C ($1 \leq C \leq 5000$) określająca ilość pokazów zaplanowanych przez profesora. W i+1 (i = 1, 2, ..., C) wierszu znajduje się liczba N ($1 \leq N \leq 2*10^9$), której wygenerowanie założył sobie profesor.

Wyjście

Dla każdej liczby N wypisz taką konfigurację maszyny, w której liczba N będzie wyznaczona przez maszynę możliwie najpóźniej. Dla każdego zestawu wypisz w osobnej linii pierwszą i drugą liczbę konfiguracji oddzielone pojedyncza spacją. Jeśli istnieje więcej niż jedna konfiguracja, która gwarantuje najdłuższy czas obliczeń, podaj tę, dla której pierwsza liczba konfiguracji jest najmniejsza. W przypadku, gdy konfiguracja maszyny nie istnieje wypisz jedno słowo: **BRAK.** Pamiętaj, że kolejność liczb w konfiguracji ma znaczenie.

Przykład

Dla danych wejściowych:

```
2

3

10
```

poprawną odpowiedzią jest:

```
1 1

2 2
```

Geolog

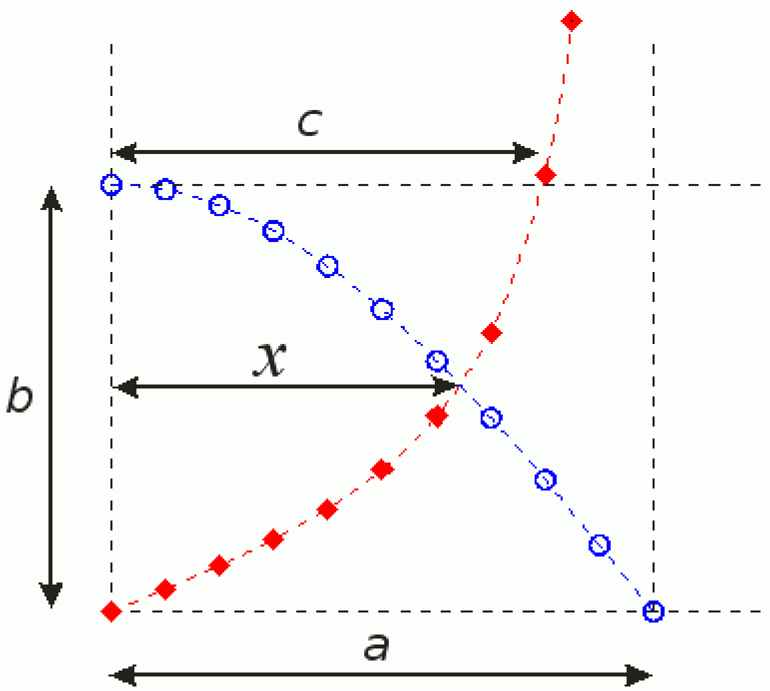
ID:1091

Limit czasu: 0.50 s

Limit pamięci: 16384 kB

Barney - geolog amator - skorzystał z okazji, aby fotografować pustkowia Szkocji "z lotu ptaka". Okazja ta związana była z ćwiczebnym lotem balonem kilku jego przyjaciół - matematyków. W czasie lotu Barney fotografował w podczerwieni i ultrafiolecie tereny nad którymi przelatywali. Już w domu, po wywołaniu filmów, stwierdził dziwne formacje skał (czy też innych obiektów) układające się w krzywoliniowe ścieżki, które wpisywały się w ortogonalną sieć starych dróg, biegnących wzdłuż południków lub równoleżników.

Większość tych ścieżek układała się w figury podobne do tych na rysunku. Kolega matematyk, któremu Barney pokazał wywołane zdjęcie, zasugerował, że krzywe, na których układały się obiekty na zdjęciach, przypominają znane wykresy funkcji "tangens" i "cosinus". Barneya zainteresowały punkty przecięcia ścieżek, które zobaczył na zdjęciach, bo - jak się spodziewał - są to ścieżki wydeptane przez celtyckich kapłanów - druidów, co wróżyło zapewne jakąś tajemnicę. Poprosił matematyka o znalezienie współrzędnych punktów przecięcia ścieżek, aby łatwiej je można było znaleźć w terenie. Ten oczywiście obiecał pomóc, bo to nie wydawało się trudne, ale ponieważ zaplanował już z kolegami lot balonem nad Pacyfikiem (ich przygody opisane są w zadaniu "Dzielni baloniarze"), prosił geologa o cierpliwość. Barney nie może jednak wytrzymać, aż kolega wróci (zresztą powrót z niebezpiecznej wyprawy wcale nie jest pewny) i szuka pomocy. Pomóż niecierpliwemu geologowi!



Niebieska ścieżka widoczna na rysunku jest 1/4 fali cosinusoidy stycznej do drogi równoleżnikowej w punkcie x = 0. Czerwona ścieżka to 1/2 tangensoidy, dla której prawa droga południkowa jest asymptotą pionową. Krzywa ta przecina północną drogę równoleżnikową w odległości *c* od zachodniej drogi południkowej ($c < a$).

Zadanie

Należy wyznaczyć współrzędną x, określającą położenie punktu przecięcia ścieżek druidów w jednostkach imperialnych: mile, jardy, stopy, cale.

Wejście

W pierwszym wierszu wejścia podana jest liczba całkowita $0 < N \le 1000$, równa liczbie zestawów danych. W kolejnych *N* wierszach występują trójki liczb całkowitych $0 < a, b, c \le 1000$, oznaczające odległości podane w milach angielskich (patrz rysunek).

Wyjście

Dla każdego zestawu danych, w jednym wierszu, należy wypisać cztery liczby całkowite: *m, j, s, c* ($0 \le m$, $0 \le j < 1760$, $0 \le s < 3$, $0 \le c < 12$) oddzielone pojedynczymi spacjami określające współrzędną x (patrz rysunek) punktu przecięcia krzywych (zaokrągloną do najbliższej liczby cali). Kolejne liczby oznaczają mile, jardy, stopy i cale. Dla porządku przypominamy że: 1 stopa=12 cali, 1 jard=3 stopy, 1 mila=1760 jardów.

Przykład

```
2
3 4 2
3 5 1
```

prawidłowym rozwiązaniem jest:

```
1 1101 1 10
0 1594 2 5
```

Rezerwacje

ID:1092

Limit czasu: 1.50 s

Limit pamięci: 32768 kB

Projektanci pewnego systemu do logistycznej obsługi magazynu natknęli się na poważny problem. Podczas testów systemu okazało się, że system rezerwacji towarów w magazynie pozwala zarezerwować towary w taki sposób, że w danej chwili magazyn nie będzie w stanie zaspokoić wszystkich potrzeb klientów wyrażonych przez rezerwacje.

Dla uproszczenia przyjmijmy, że zajmujemy się jednym towarem. Klient zamawia określoną ilość konkretnego towaru. Towar w magazynie posiada 3 właściwości, nazwijmy je A, B i C. Wartość każdej własności konkretnej partii towaru w magazynie jest dodatnią liczbą całkowitą.

Są klienci, których nie interesują właściwości towaru, i po prostu zamawiają daną ilość.

Są też tacy, którym zależy na towarze o konkretnych własnościach. Mogą oni określić dokładną wartość jednej, dwóch a nawet trzech własności. Magazyn nie ma prawa wydać klientowi towaru o innych własnościach niż zamówione.

Twoim zadaniem będzie napisanie programu, który stwierdzi czy dla zadanego stanu magazynu (tj. wykazu ilości towaru i jego własności) oraz listy rezerwacji, możliwe jest wydanie towaru realizujące wszystkie rezerwacje.

Wejście

W pierwszym wierszu wejścia znajduje się liczba Z, $1 \leq Z \leq 100$, oznaczająca liczbę zestawów danych. Każdy zestaw danych rozpoczyna się linią zawierająca 2 liczby naturalne: S, R, ($2 \leq S+R \leq 500$; $S, R > 0$), gdzie S oznacza liczbę wierszy na liście definiującej stan magazynu, natomiast R oznacza liczbę rezerwacji.

W kolejnych S wierszach podany jest bieżący stan magazynu. Każdy wiersz stanu zawiera 4 liczby naturalne: I, A, B, C, ($1 \leq I \leq 1000$; $1 \leq A, B, C \leq$ 10^9), oznaczające odpowiednio ilość towaru oraz wartości własności A, B, C towaru.

Ostatnie R wierszy zestawu opisuje zbiór złożonych rezerwacji. Każda rezerwacja to jeden wiersz, zawierający dokładnie 4 liczby całkowite: i, a, b, c, ($1 \leq i \leq 1000$; $0 \leq a, b, c \leq$ 10^9), oznaczające odpowiednio ilość oraz oczekiwane przez klienta własności towaru. Jeśli klient oczekuje własności o wartości 0, oznacza to, że dana własność towaru go nie interesuje. W szczególności, jeśli w rezerwacji wszystkie 3 własności towaru są równe 0, klient zadowoli się dowolnym towarem w żądanej ilości, i niekoniecznie cały towar odpowiadający rezerwacji musi być pod względem własności jednorodny.

Liczby występujące w wierszach wejścia oddzielone są od siebie pojedynczą spacją.

Wyjście

Dla każdego zestawu danych należy wydać na standardowe wyjście linię zawierającą słowo "tak" jeśli magazyn jest w stanie zrealizować wszystkie rezerwacje, a słowo "nie" jeśli jest to niemożliwe.

Przykład

Dla danych:

```
2
2 2
2 1 1 1
2 2 3 1
2 0 0 0
1 2 0 0
1 1
100 1 2 3
101 0 0 0
```

poprawną odpowiedzią jest:

```
tak
nie
```

Samotnik

ID:1093

Limit czasu: 1.50 s

Limit pamięci: 65536 kB

Niedawno cały świat opanowała nowa gra o nazwie "Samotnik". Jej zasady są proste: w grze bierze udział jeden gracz, który ma do dyspozycji szachownicę oraz pewną liczbę pionków rozłożonych na polach szachownicy. Na jednym polu może leżeć co najwyżej 1 pionek.

Pionki są zdejmowane w kolejnych ruchach zgodnie z zasadami, że w jednym ruchu można usunąć jedynie takie pionki, które znajdują się w jednym wierszu bądź w jednej kolumnie szachownicy (w szczególności oznacza to, że dopuszczone jest wykonanie ruchu polegającego na zdjęciu tylko jednego pionka). Wszystkie pionki należy zdjąć, wykonując przy tym najmniejszą liczbę ruchów.

Zadanie

Napisz program, który dla zadanego ustawienia pionków wyznaczy najmniejszą liczbę ruchów potrzebną do zdjęcia wszystkich pionków z szachownicy.

Wejście

Pierwsza linia wejścia zawiera liczbę całkowitą C ($1 \leq C \leq 5$), oznaczającą liczbę zestawów danych. W kolejnych wierszach znajdują się opisy zestawów danych. W pierwszej linii opisu zestawu znajduje się liczba N ($1 \leq N \leq 100000$), oznaczająca liczbę pionków znajdujących się na szachownicy. Kolejne N wierszy zawiera opis położenia pionków. Opis położenia pionka to dwie liczby naturalne, określające współrzędne X i Y pionka oddzielone pojedynczą spacją: x, y, ($1 \leq x \leq 2*10^9$, $1 \leq y \leq 2*10^9$).

Wyjście

Dla każdego zestawu danych należy wypisać w jednej linii minimalną liczbę ruchów potrzebnych do usunięcia wszystkich pionków z planszy w danej grze.

Przykład

Dla danych:

```
1
7
5 12
10 55
10 30
25 12
44 25
5 25
10 1
```

poprawną odpowiedzią jest:

```
3
```

Kosmita

ID:1094

Limit czasu: 5.00 s

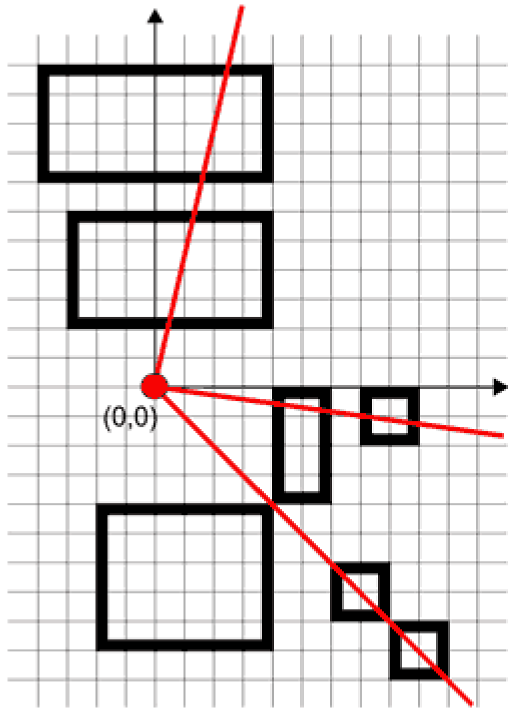
Limit pamięci: 32768 kB

Kosmita wylądował w centrum miasta i chce zniszczyć wszystkie budynki. Tym razem, wyjątkowo, nie będziesz ratował świata, ale pomożesz kosmicie.

Statek kosmity jest wyposażony w potężne działo laserowe dużej mocy. Działo strzela wiązkami laserowymi, które są półprostymi wychodzącymi ze statku kosmity. Wiązka lasera jest tak silna, że przechodzi przez wszystkie budynki znajdujące się na jej drodze. Budynek trafiony promieniem lasera natychmiast zamienia się w obłok pary. Wiązka lasera przemieszcza się tuż nad ziemią - jest w stanie trafić w każdy budynek na swojej drodze niezależnie od jego wysokości. Każdy strzał pozbawia kosmitę zasobów energii, dzięki której może wrócić na macierzystą planetę. Kosmicie zależy na tym, aby zniszczyć wszystkie budynki możliwie najmniejszą liczbą strzałów.

Zadanie

Dysponując dokładnym planem rozmieszczenia budynków, musisz wyznaczyć minimalną liczbę strzałów potrzebną do zniszczenia wszystkich budynków, przy założeniu, że statek nie może zmieniać swojej pozycji.



Wejście

W pierwszym wierszu wejścia znajduje się liczba C, $1 \le C \le 10$, określająca liczbę zestawów danych. W kolejnych wierszach znajdują się zestawy danych. W pierwszym wierszu każdego zestawu danych znajduje się liczba N, $1 \le N \le 10^5$, oznaczająca ilość budynków w mieście. W kolejnych N wierszach znajdują się opisy budynków. W każdym następnym wierszu danego zestawu znajdują się 4 liczby całkowite: xl, yg, xp, yd, $-10^9 \le xl, yg, xp, yd \le 10^9$, $xl < xp$, $yd < yg$, gdzie punkt (xl, yg) jest lewym górnym rogiem budynku, a punkt (xp,yd) prawym dolnym rogiem budynku. Zakładamy, że wszystkie budynki mają ściany prostopadłe do osi układu współrzędnych. Żadne 2 budynki nie zajmują wspólnej powierzchni, jednak mogą "dotykać się" rogami i bokami. Ponieważ niezbędnych pomiarów dokonujemy na statku kosmicznym, więc początek układu współrzędnych został ustalony w miejscu lądowania statku. W tym miejscu nie stoi też żaden budynek.

Wyjście

Dla każdego zestawu danych należy wypisać na standardowe wyjście linię zawierającą jedną liczbę naturalną, oznaczającą ilość strzałów konieczną do zniszczenia wszystkich budynków, przy założeniu, że statek nie zmienia pozycji i znajduje się w punkcie o współrzędnych (0, 0).

Przykład

Dla danych:

```
1
7
4 0 6 -4
7 0 9 -2
6 -6 8 -8
8 -8 10 -10
-2 -4 4 -9
-3 6 4 2
-4 11 4 7
```

poprawną odpowiedzią jest:

```
3
```

Łaty

ID:1095

Limit czasu: 1.50 s

Limit pamięci: 32768 kB

Pewna firma zajmująca się tworzeniem oprogramowania spostrzegła niedawno, że jej system łat jest bardzo skomplikowany. Programy przez nią produkowane składają się z K komponentów ponumerowanych kolejnymi liczbami naturalnymi od 1 do K. Każda łata uaktualnia część komponentów programu do pewnej wersji. Łatę uaktualniającą do wersji v+1 można użyć tylko wtedy, gdy wszystkie komponenty uaktualniane przez tę łatę mają wersję v.

Zadanie

Napisz program, który sprawdzi, czy przy zadanych łatach da się uaktualnić cały program, czyli wszystkie jego komponenty, z wersji 1 do wersji zadanej w pliku wejściowym. Jeśli da się uaktualnić cały program, wówczas należy wypisać numery łat, z których trzeba skorzystać.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych D, $1 \leq D \leq 10$. W następnych liniach znajdują się kolejno po sobie opisy D zestawów danych. Pierwsza linia zestawu danych zawiera trzy liczby naturalne oddzielone pojedynczymi spacjami: K, L oraz V, $1 \leq K \leq 10$, $1 \leq L \leq 10000$, $2 \leq V \leq 1000$. K oznacza liczbę komponentów, z których składa się program, L - liczbę łat, którymi można uaktualniać program, natomiast V - wersję, do której należy uaktualnić wszystkie komponenty programu. W kolejnych liniach zestawu danych znajdują się opisy L łat. Opis łaty składa się z dwóch linii. W pierwszej linii opisu łaty znajdują się dwie liczby naturalne oddzielone spacją: VL, oznaczająca wersję, do której łata uaktualnia komponenty oraz KL, oznaczająca liczbę komponentów uaktualnianych przez łatę ($2 \leq VL \leq V$, $1 \leq KL \leq K$). W drugiej linii opisu łaty znajduje się KL numerów komponentów, które uaktualnia łata. Sąsiednie numery w pliku wejściowym oddzielone są od siebie spacją. Numery uaktualnianych komponentów podane są w kolejności rosnącej. Łaty są ponumerowane kolejnymi liczbami naturalnymi od 1 do L w kolejności pojawiania się ich opisów w pliku wejściowym.

Wyjście

W oddzielnych liniach dla każdego zestawu danych należy wypisać numery łat w kolejności, w jakiej powinny być stosowane. Jeśli istnieje wiele rozwiązań, należy wypisać ciąg numerów łat najmniejszy leksykograficznie. Jeśli nie da się uaktualnić programu do wersji V, wówczas trzeba wypisać liczbę -1.

Przykład

Dla danych wejściowych:

```
2
7 10 4
4 2
1 2
3 4
1 2 3 4
2 2
```

```
4 5
2 3
1 2 3
3 2
1 7
3 3
5 6 7
2 2
6 7
2 3
3 4 5
2 7
1 2 3 4 5 6 7
4 5
3 4 5 6 7
3 4 3
3 2
1 2
3 2
2 3
2 3
1 2 3
3 2
1 3
```

poprawną odpowiedzią jest:

```
3 4 2 1 7 6 10
-1
```

Owce

ID:1096

Limit czasu: 1.50 s

Limit pamięci: 32768 kB

Juhasi Jędruś i Bartuś postanowili urozmaicić sobie wieczorne zaganianie owiec wymyśloną przez siebie zabawą. Każdy z nich z pomocą psa może zagonić do zagrody dowolną liczbę owiec pasących się na hali w kilku grupach. Ponieważ mają jednego psa, to zapędzanie owiec będą robili na zmianę: najpierw Jędruś potem Bartuś i potem znów Jędruś itd. aż do zapędzenia ostatniej. Ten kto zapędza ostatnią owcę przegrywa, bo jego kolega może w tym czasie wziąć jedyną butelkę piwa, która się chłodzi w potoku.

Ponieważ grupy owiec są od siebie oddalone to w czasie jednego zapędzania można zagonić dowolną liczbę owiec (minimum jedną, maksimum wszystkie) pod warunkiem, że należą do jednej grupy. Kolejność wyboru grup jest dowolna.

Po kilku zmianach, Jędruś wziął butelkę piwa z potoku mimo, że jeszcze sporo owiec pasło się na hali, bo jak powiedział i tak będzie ona jemu się należała bez względu na to co zrobi Bartuś. Zastanów się czy jest to możliwe?

Zadanie

Napisz program, który wskaże juhasa, który wygra zabawę przy założeniu, że zaczyna zawsze Jędruś i obaj są dostatecznie sprytni aby wykorzystać wszystkie szanse jakie stwarza im układ owiec pasących się na hali.

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita C, oznaczająca liczbę zestawów danych, $1 \leq C \leq 10$. W następnych C wierszach podane są liczby całkowite oznaczające liczbę grup owiec i ich liczebność. Pierwsza liczba w wierszu oznacza liczbę grup owiec N, $1 \leq N \leq 100000$, a następujące po niej N liczb ai, $1 \leq ai \leq 2^{31}$-1 dla $1 \leq i \leq N$ oznaczają liczebność każdej grupy.

Wyjście

Na wyjściu, w C wierszach należy wypisać pojedyncze duże litery: *J*, gdy wygra Jędruś lub *B*, gdy wygra Bartuś.

Przykład

Dla danych:

```
2
5 7 9 23 11 17
3 1 2 3
```

prawidłowym rozwiązaniem jest:

```
J
B
```

Zależności

ID:1097

Limit czasu: 1.00 s

Limit pamięci: 16384 kB

Prawie każdy, kto instalował oprogramowanie na uniksowych systemach operacyjnych miał do czynienia z zależnościami pomiędzy różnymi pakietami.

Zależność pomiędzy pakietami jest opisana w postaci par liczb (numerów pakietów) oddzielonych od siebie pojedynczą spacją. Para A B (gdzie A, B są różnymi numerami pakietów) oznacza, że pakiet A powinien być zainstalowany przed pakietem B.

Dla przykładu, zależności opisane przez następujące pary:

A B

B C

B D

mogą być instalowane w systemie w następującej kolejności:

A B C D

lub

A B D C.

Wejście

W pierwszej linii wejścia znajduje się liczba zestawów danych D, $1 \leq D \leq 10$. W kolejnych liniach znajdują się zestawy danych. Pierwsza linia zestawu zawiera dwie liczby N, $1 \leq N \leq 100000$, oraz Z, $0 \leq Z \leq 100000$. Liczba N oznacza liczbę pakietów do zainstalowania (pakiety numerowane są kolejnymi liczbami od 1 do N), natomiast liczba Z - liczbę zależności. W kolejnych Z liniach zestawu znajduje się opis zależności w postaci par A, B (dwóch numerów pakietów oddzielonych pojedynczą spacją), $1 \leq A, B \leq N$.

Wszystkie zestawy danych są poprawne, co oznacza, że nie może dojść do zapętleń, np. *A B* (pakiet *B* zależy od *A*), *B C* (pakiet *C* zależy od *B*), *C A* (pakiet *A* zależy od *C*).

Wyjście

Dla każdego zestawu danych program powinien wypisać listę numerów pakietów (oddzielonych pojedynczą spacją) w kolejności w jakiej powinny być instalowane. Jeśli jest więcej niż jedno rozwiązanie, program powinien wypisać pierwsze (najmniejsze) z nich w porządku leksykograficznym.

Przykład

Dla następujących danych:

```
2
5 6
```

```
1 2
1 4
3 1
3 4
4 5
2 5
4 3
1 2
2 3
3 4
```

poprawnym rozwiązaniem jest:

```
3 1 2 4 5
1 2 3 4
```

Samoopisujące się liczby

ID:1098

Limit czasu: 1.00 s

Limit pamięci: 16384 kB

Liczbę całkowitą dodatnią L nazwiemy samoopisującą się liczbą, jeżeli jej pierwsza cyfra w zapisie dziesiętnym (najbardziej znacząca cyfra - pierwsza cyfra od lewej) oznacza liczbę wystąpień cyfry '0' w zapisie tej liczby, druga cyfra oznacza liczbę wystąpień '1' w zapisie tej liczby, itd...

Zadanie

Dla zadanego n należy wyznaczyć największą liczbę samoopisującą nie większą od n.

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita D, $1 \leq D \leq 100$, oznaczająca liczbę zestawów danych. W każdym z D kolejnych wierszy znajduje się liczba całkowita n, $0 \leq n < 1010$, po jednej dla każdego zestawu.

Wyjście

Na wyjściu, w jednym wierszu dla każdego zestawu, należy wypisać największą liczbę samoopisującą nie większą od zadanej liczby n. W przypadku braku rozwiązania należy wypisać -1.

Przykład

Dla danych wejściowych:

```
1
2000
```

poprawną odpowiedzią jest:

```
1210
```

Płytki

ID:1099

Limit czasu: 1.50 s

Limit pamięci: 16384 kB

Jaś postanowił zbudować sobie domek na działce. Chciał rozpocząć od położenia terakoty, więc zakupił dużą ilość kwadratowych płytek i ułożył je na całej działce w szachownicę nie docinając żadnej z płytek. Ściany natomiast Jaś chciał postawić na płytkach w taki sposób, aby całe wnętrze domku było pokryte płytkami. Jeśli Jaś stawiał ścianę na płytce (ale nie na krawędzi płytki), to musiał przyciąć płytkę.

Napisz program, który pomoże Jasiowi znaleźć najmniejszą ilość płytek, które musi pociąć. Zakładamy, że ściana ma grubość zerową.

Wejście

Pierwszy wiersz zawiera dodatnią liczbę całkowitą C ($1 \le C \le 10$) określającą ilość zestawów danych. W kolejnych liniach znajdują się zestawy danych. Pierwszy wiersz zestawu danych zawiera dodatnią liczbę całkowitą D ($1 \le D \le 5$) określającą długość boku płytki. Drugi wiersz zestawu danych zawiera liczbę całkowitą N ($4 \le N < 10000$) określającą liczbę ścian w domku.

Wiersz o numerze $i + 2$ ($i = 1, 2, ..., N$) zestawu danych zawiera dwie liczby całkowite x_i ($0 \le x_i \le 1000000$) oraz y_i ($0 \le y_i \le 1000000$) oddzielone spacjami określające kolejne rogi domku, czyli współrzędne (x_i, y_i) i (x_{i+1}, y_{i+1}) dla i=1, 2, ..., N-1 oraz (x_N, y_N) i (x_1, y_1) określają końce ścian. Ponadto (x_i=x_{i+1}) lub (y_i=y_{i+1}) dla i=1, 2, ..., N-1. Długości ścian są liczbami dodatnimi.

Wyjście

Na wyjściu powinna pojawić się jedna nieujemna liczba całkowita, określająca minimalną ilość płytek jaką trzeba pociąć.

Przykład

Dla danych wejściowych:

```
1
3
4
1 1
9 1
9 10
1 10
```

poprawną odpowiedzią jest:

```
3
```

Rachmistrz

ID:1100

Limit czasu: 1.00 s

Limit pamięci: 16384 kB

Ludzie od wieków, jeszcze w czasach "przedkomputerowych", fascynowali się rachmistrzami. Dotyczyło to w szczególności cudownych, najczęściej autystycznych dzieci, które często nie umiały czytać i pisać, ale liczyły z zadziwiającą sprawnością. W szczególności w XIX wieku zostało opisanych kilku takich wyjątkowych rachmistrzów egzaminowanych przez Francuską Akademię Nauk. Jednym z nich był urodzony w 1826 roku, młody pasterz owiec Henri Mondeux. Zapytany, jakie liczby podniesione do kwadratu maja różnicę równą 133 odpowiedział, że to 66 i 67 a po chwili, że jest także inne rozwiązanie: 6 i 13. Dziś taka fascynacja rachmistrzami już bezpowrotnie minęła, bo w erze powszechnej dostępności kalkulatorów i komputerów trudno by zgromadzić publiczność, która chciałaby przyglądać się takim popisom, ale pozostała nadal ciekawość odkrywania sposobów, dzięki którym takie skomplikowane operacje arytmetyczne można sprawnie wykonać.

Spróbuj postawić się w położeniu małego Henri.

Zadanie

Należy znaleźć rozwiązanie równania $a^2 - b^2 = c$, gdzie c jest zadaną liczbą. Liczby a, b, c są liczbami całkowitymi nieujemnymi.

W przypadku istnienia kilku różnych rozwiązań, tj. par (a, b) spełniających równanie, interesuje nas to rozwiązanie dla którego różnica a-b jest najmniejsza. Jeżeli istnieje kilka rozwiązań dla których różnica a-b jest taka sama, należy wybrać to, dla którego liczba b jest najmniejsza.

Wejście

W pierwszym wierszu wejścia podana jest liczba całkowita L, $1 \le L \le 60000$, oznaczająca liczbę zestawów danych. W kolejnych L wierszach występują wartości c_i, $0 \le c_i \le 5 \cdot 10^6$.

Wyjście

Na wyjściu, w jednym wierszu dla każdej danej ci, należy wypisać jedną liczbę całkowitą bi, spełniającą warunki opisane w zadaniu. Jeżeli rozwiązanie nie istnieje, należy wypisać liczbę -1.

Przykłady

Dla danych:

```
2
133
28900
```

prawidłowym rozwiązaniem jest:

```
66
7224
```

Ciąg

ID:1101

Limit czasu: 3.00 s

Limit pamięci: 16384 kB

Danych jest N dowolnych liczb $a_1, a_2, a_3, .. , a_N$, będących początkowymi wyrazami pewnego ciągu. Kolejne wyrazy ciągu powstają poprzez zsumowanie N bezpośrednio je poprzedzających wyrazów tego ciągu:

$a_{N+1} = a_1 + a_2 + .. + a_N,$

$a_{N+2} = a_2 + a_3 + .. + a_{N+1},$

..

Zadanie

Wyznaczyć pięć ostatnich cyfr liczby, będącej sumą wyrazów powyższego ciągu, począwszy od wyrazu ap, a skończywszy na wyrazie ak.

Wejście

Pierwszą linię wejścia stanowi liczba D, $1 \leq D \leq 20$, wyznaczająca liczbę zestawów danych. W dalszej części wejścia znajduje się D zestawów danych. Zestaw danych składa się z trzech linii. Pierwszą z nich stanowi liczba całkowita N, $2 \leq N \leq 50$, określająca początkową liczbę wyrazów ciągu. W drugiej zaś znajduje się N, oddzielonych pojedynczą spacją, liczb całkowitych z przedziału <0; 10^5>, będących początkowymi wyrazami ciągu. Ostatnia linia zestawu danych ma postać p k, gdzie $0 < p \leq k < 2^{31}$ oraz p oznacza numer wyrazu od którego rozpoczynamy sumowanie, a k numer wyrazu dla którego kończymy sumowanie.

Wyjście

Dla każdego z D zestawów danych, w osobnej linii wyjścia należy wypisać pięć (lub mniej - bez zer wiodących) ostatnich cyfr liczby, będącej sumą wyrazów ciągu, począwszy od wyrazu ap do wyrazu ak włącznie.

Przykład

Dla danych:

```
1
4
2 3 4 5
5 6
```

poprawnym rozwiązaniem jest:

```
40
```

Komórka

ID:1102

Limit czasu: 0.50 s

Limit pamięci: 4096 kB

Numery telefonów w Opsslandii są zapisywane w postaci wyrazów, aby książki telefoniczne były bardziej przystępne dla przeciętnego czytelnika. Każdej literze wyrazu reprezentującego numer telefonu odpowiada jedna cyfra.

```
ABC: 2
DEF: 3
GHI: 4
JKL: 5
MNO: 6
PQRS: 7
TUV: 8
WXYZ: 9
```

Poza Opsslandią numery telefoniczne zapisywane są w postaci ciągu cyfr i osoby spoza Opsslandii mają trudności z rozkodowaniem numerów wyrazowych.

Napisz program tłumaczący wyrazowe numery telefonów na ich odpowiedniki cyfrowe.

Wejście

Pierwszy wiersz zawiera dodatnią liczbę całkowitą C ($1 \leq C \leq 100000$) określającą ilość wyrazowych numerów telefonicznych.

W każdym z kolejnych C wierszy znajduje się wyraz do przetłumaczenia. Każdy wyraz zawiera co najmniej 1 i co najwyżej 100 liter.

Wyjście

Dla każdego wyrazowego numeru telefonu należy, w osobnej lini, wypisać odpowiadający mu numer telefonu w postaci ciągu cyfr.

Przykład

Dla danych wejściowych:

```
1
OPSS
```

poprawną odpowiedzią jest:

```
6777
```

Weselne toasty

ID:1103

Limit czasu: 0.75 s

Limit pamięci: 16384 kB

Młody, zdolny matematyk Karol, został zaproszony na ślub i przyjęcie weselne swojego kolegi. W czasie tego przyjęcia wznoszono wiele toastów a biesiadnicy trącali się kieliszkami, co podobno jest elementem naszej kultury. Aby uniknąć dużego zamieszania, stukanie kieliszkami dotyczyło zwykle najbliższych sąsiadów przy stole (prawy do lewego, lewy do prawego) oraz tych po drugiej stronie stołu.

Karol zaczął zastanawiać się, jak wiele różnych konfiguracji takich stuknięć między dwoma biesiadnikami jest możliwych przy założeniach, że jest parzysta liczba gości, wszystkie stuknięcia odbywają się jednocześnie, każdy z biesiadników ma tylko jeden kieliszek a ręce biesiadników nie mogą się krzyżować. Biesiadnicy siedzący przy okrągłym stole nie opuszczają także swoich miejsc w czasie toastu. Liczby konfiguracji toastów, które Karolowi wyszły w czasie tych rozważań były dosyć dziwne ale to może wina tych wielu toastów? Sprawdź obliczenia Karola.

Zadanie

Należy wyznaczyć liczbę różnych konfiguracji, które powstają przy trącaniu się kieliszkami parzystej liczby osób siedzących przy okrągłym stole. Stuknięcia kieliszkami w jednej konfiguracji odbywają się w tej samej chwili a ręce biesiadników nie mogą się krzyżować.

Wejście

W pierwszym wierszu wejścia podana jest liczba całkowita D ($0<D\leq1000$), oznaczająca liczbę zestawów danych. W kolejnych D wierszach występują wartości N_i ($0<N_i\leq8000$), oznaczające liczby par biesiadników.

Wyjście

Na wyjściu, w jednym wierszu dla każdej danej N_i, należy wypisać dwie liczby całkowite: K_i - liczbę konfiguracji stuknięć kieliszkami i C_i - ilość cyfr liczby K_i. Jeżeli liczba K_i jest większa lub równa 10^9 to należy wypisać tylko 9 jej pierwszych cyfr.

Przykład

Dla danych:

```
2
5
100
```

prawidłowym rozwiązaniem jest:

```
42 2
896519947 57
```

Ułamki

ID:1104

Limit czasu: 1.00 s

Limit pamięci: 16384 kB

Krzyś ma kłopoty z działaniami na ułamkach. Jego tata, chcąc pomóc synkowi w szkolnych problemach zaproponował mu zabawę w dodawanie "nietypowych" ułamków. Nietypowość polega na tym, że sposób (format) zapisywania ułamków jest bardzo odmienny od tego, którego nauczają w szkole.

Przyłącz się do tej edukacyjnej zabawy.

Format zapisywania ułamków to "a", "a/b/c" lub "b/c", gdzie a, b, c są liczbami całkowitymi nieujemnymi. Zapis "a" oznacza liczbę całkowitą nieujemną, zapis "a/b/c" oznacza liczbę będącą tzw. ułamkiem mieszanym (a+b/c), gdzie część ułamkowa jest właściwa ale może nie być skrócona, zaś zapis "b/c" oznacza ułamek właściwy, który może nie być skrócony.

Zadanie

Obliczyć sumę dwóch ułamków podanych w "nietypowym" formacie. Wynik przedstawić również w "nietypowym" formacie ze skróconą częścią ułamkową.

Wejście

W pierwszej linii wejścia znajduje się liczba D, $1 \le D \le 1000$, oznaczająca liczbę zestawów danych. W kolejnych D wierszach występują po dwa ułamki, które należy zsumować. Każdy ułamek zapisany jest w "nietypowym" formacie, tj. "a", "a/b/c" lub "b/c", gdzie a, b, c są liczbami całkowitymi, $0 \le a \le 10000$, $0 < b < c \le 10000$. Ułamki oddzielone są od siebie znakiem "+".

Wyjście

Na wyjściu w osobnej linii dla każdego zestawu należy wypisać sumę dwóch ułamków w "nietypowym" formacie przy czym części ułamkowe muszą być ułamkami nieskracalnymi.

Przykład

Dla danych wejściowych:

```
2
10+7/8
1/2/3+1/1/2
```

poprawną odpowiedzią jest:

```
10/7/8
3/1/6
```

Protest ekologów

ID:1105

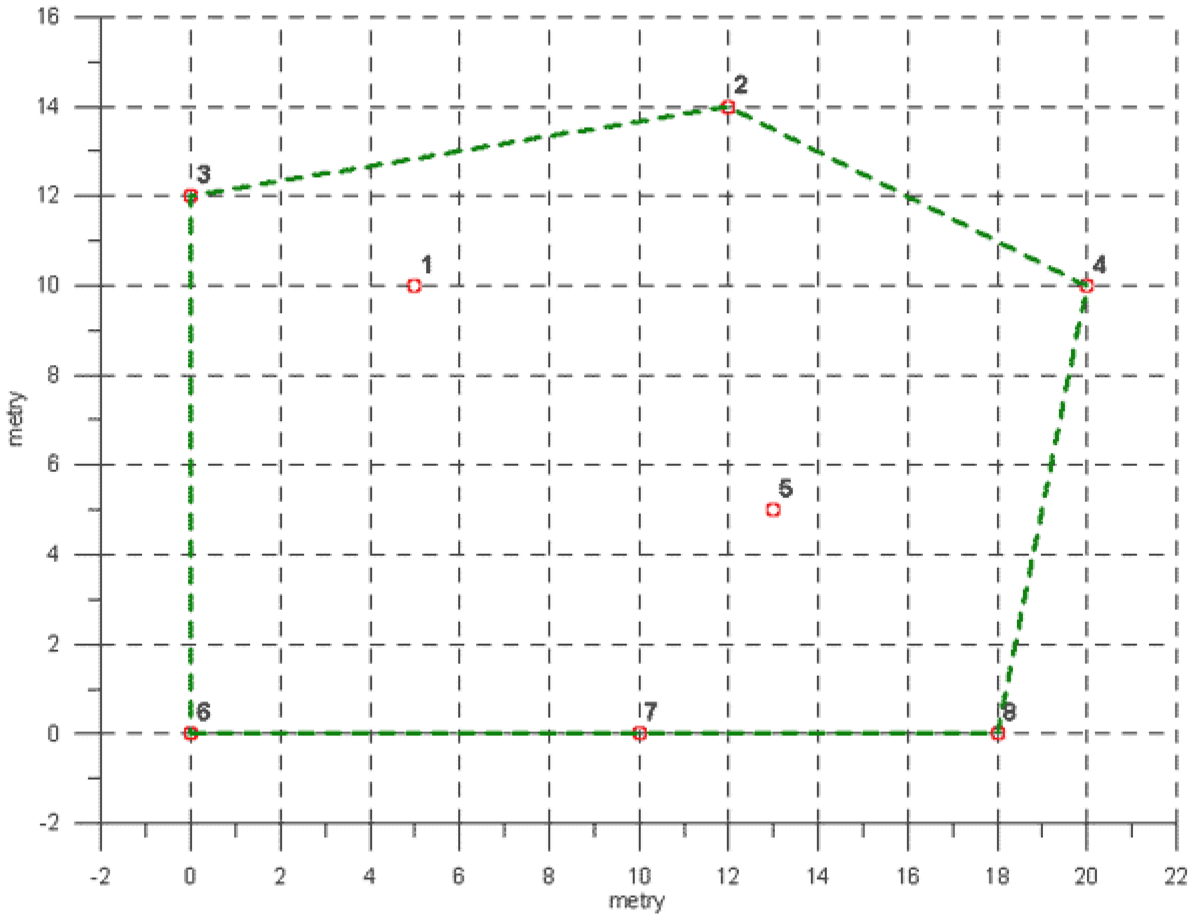
Limit czasu: 2.00 s

Limit pamięci: 16384 kB

Uwaga! Historia jest w całej rozciągłości fikcją literacką i jakiekolwiek podobieństwo do faktów i zdarzeń jest zupełnie przypadkowe.

W pobliżu dosyć dużego, uniwersyteckiego miasta postanowiono rozbudować istniejące lotnisko sportowe. Planowane jest wydłużenie i poszerzenia pasa startowego oraz rozbudowa infrastruktury lotniska. Inwestycja pewnie ruszyłaby już pełną parą gdyby nie problemy ekologiczne. Trawiaste lotnisko sportowe zamieszkuje mianowicie dosyć rzadki gatunek susła, który znalazł tu doskonałe warunki rozwoju. Inwestorzy chcą wyłapać sympatyczne zwierzaki i przenieść je w inne miejsce, ale ekolodzy uważają, że nigdzie nie znajdzie się tak dobrego siedliska dla susłów i chcą bronić ich spokoju. Aby zapobiec wyłapywaniu susłów ekolodzy planują otoczyć ich norki "żywym łańcuchem" biorąc się za ręce.

Znane jest położenie wejść do norek susłów, a problem ekologów polega na określeniu minimalnej liczby ludzi, którzy potrzebni są do utworzenia "żywego łańcucha". Spróbuj rozwiązać problem ekologów!



Zadanie

Należy wyznaczyć minimalną liczbę ekologów, którzy potrzebni są do utworzenia "żywego łańcucha" chroniącego norki susłów przy założeniu, że średnia rozpiętość ramion ekologa (i ekolożki?) to 150cm. Dodatkowo należy też podać numery punktów oznaczających wejścia do norek przez które łańcuch będzie przechodził.

Wejście

W pierwszym wierszu wejścia podana jest liczba całkowita $0<L<1001$, oznaczająca liczbę zestawów danych. W każdym z L kolejnych wierszy występuje liczba N_i ($2< N_i<200000$), oznaczająca liczbę punktów (norek susłów) oraz Ni różnych par liczb całkowitych x_i, y_i ($-100001<x_i, y_i< 100001$) oznaczających współrzędne punktów podane w centymetrach. Wszystkie liczby oddzielone są pojedynczą spacją.

Wyjście

Na wyjściu, w jednym wierszu dla każdego zestawu danych, należy wypisać liczbę ekologów niezbędną do utworzenia łańcucha oraz listę numerów punktów, przez które łańcuch będzie przechodził. Lista punktów powinna rozpoczynać się od punktu o najmniejszej współrzędnej x, przez który będzie przechodził łańcuch a następne punkty na obwodzie łańcucha powinny tworzyć obieg o kierunku przeciwnym do ruchu wskazówek zegara. Jeżeli jest kilka punktów na obwodzie łańcucha, których współrzędna x równa jest xmin to punktem początkowym łańcucha jest ten, który ma najmniejszą współrzędną y. Numeracja punktów jest zgodna z ich kolejnością podaną na wejściu.

Przykłady

```
2
8 500 1000 1200 1400 0 1200 2000 1000 1300 500 0 0 1000 0 1800 0
14 200 1000 1200 1400 500 1000 2000 1000 1300 500 0 0 1000 -200 1800 0 100 500
600 600 1400 200 1400 1000 1600 1200 1900 500
```

prawidłowym rozwiązaniem jest:

```
41 6 7 8 4 2 3
40 6 7 8 14 4 13 2 1 9
```

Grafy animalne

ID:1106

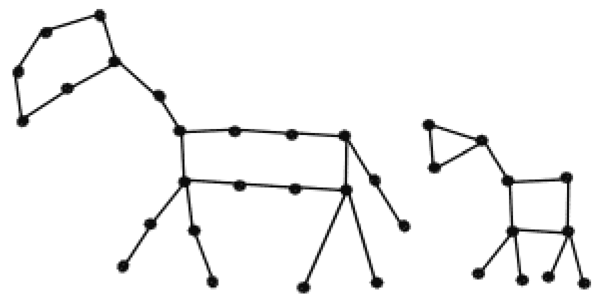
Limit czasu: 3.00 s

Limit pamięci: 16384 kB

Zdefiniujmy pojęcie grafu animalnego. Obrazowo można opisać graf animalny, jako graf który odpowiednio narysowany na płaszczyźnie może przedstawiać schemat czworonoga.

A bardziej formalnie:

- Graf G nazywamy grafem animalnym, jeśli spełnia następujące warunki:
- G jest prosty, spójny, zawiera dokładnie 2 cykle, każdy o długości co najmniej 3.
- Oba cykle są rozłączne, połączone są pojedynczą ścieżką, zawierającą co najmniej 1 krawędź (ścieżka ta będzie szyją czworonoga).
- Jeden z cykli jest krótszy od drugiego. Krótszy cykl nazywać będziemy głową, dłuższy tułowiem.
- Dokładnie 2 wierzchołki tułowia mają stopień 4. Do każdego z tych wierzchołków dołączone są 2 ścieżki równej długości (są to 2 pary nóg naszego czworonoga).
- 1 lub 2 wierzchołki tułowia mają stopień 3. Jeden z nich to wierzchołek łączący tułów z głową, a ewentualny drugi wierzchołek stopnia 3 to miejsce w którym dołączona do tułowia jest jeszcze jedna ścieżka (to ogon naszego czworonoga - nie każdy czworonóg ma ogon).
- Pozostałe wierzchołki tułowia mają stopnie równe 2.
- Wszystkie wierzchołki głowy mają stopień 2, za wyjątkiem jednego, łączącego głowę z tułowiem. Ma on stopień 3.



Rys. Przykłady grafów animalnych.

Najmniejszy graf animalny ma 11 wierzchołków. Twoim zadaniem będzie badanie animalności grafów.

Wejście

W pierwszym wierszu wejścia podana jest liczba grafów do zbadania C, $1 \le C \le 100$. W kolejnych wierszach podane są kolejne grafy. W pierwszym wierszu opisu grafu znajduje się liczba n, $1 \le n \le 100000$ określająca liczbę wierzchołków grafu. W wierszach 2..n, kolejno dla wierzchołków 1..n-1, podani są sąsiedzi każdego wierzchołka o większych numerach (wierzchołki są numerowane od 1 do n). Pierwszą liczbą wiersza i+1 opisu grafu, jest liczba sąsiadów wierzchołka i o numerach większych od i. W dalszej części wiersza i+1 podane są, oddzielone pojedynczymi spacjami, numery sąsiadów wierzchołka i, o numerach większych od i. Wierzchołek o numerze n nie może mieć sąsiadów o numerach większych od n, stąd mamy tylko n-1 wierszy opisujących sąsiedztwa wierzchołków. Każdy graf wejściowy jest grafem prostym.

Wyjście

Wyjście powinno zawierać po jednym wierszu dla każdego grafu. W wierszu i-tym powinno znaleźć się słowo "tak" jeśli i-ty graf wejściowy jest grafem animalnym, a słowo "nie" jeśli nim nie jest.

Przykład

Dla danych:

```
2
19
2 2 3
1 3
1 4
1 5
1 6
2 10 7
3 8 16 18
1 9
2 10 15
2 13 14
1 13
1 14
0
0
0
1 17
0
1 19
11
2 10 11
2 8 9
1 7
1 7
1 8
1 8
2 8 9
0
1 10
1 11
```

Poprawną odpowiedzią jest:

```
tak
tak
```

Rybki

ID:1107

Limit czasu: 0.50 s

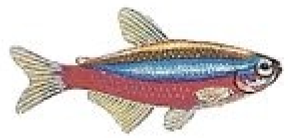
Limit pamięci: 4096 kB

Agatka podjęła decyzję o założeniu domowego akwarium. Przygotowania rozpoczęła od wybrania gatunków rybek, które chciałaby hodować, potem zaś zgromadziła informacje, jakich warunków życiowych potrzebują te gatunki. Okazało się, że każda z wybranych rybek potrzebuje wody o innej temperaturze - Neon Innesa żyje w wodzie o temperaturze od 22 do 26°C, Gupik od 18 do 28°C, Mieczyk Hellera od 20 do 28°C. Agatka dość szybko obliczyła, że aby zapewnić przetrwanie wszystkim wybranym przez nią rybkom, woda w akwarium musi mieć odpowiednią temperaturę - od 22 do 26°C.

Neon Innesa

Paracheirodon innesi

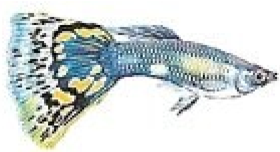
22-26°C



Gupik

Poecilia reticulata

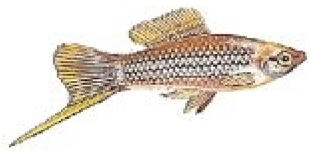
18-28°C



Mieczyk Hellera

Xiphophorus helleri

20-28°C



Zadanie

Napisz program, który pomoże nie tylko Agatce, ale również innym potencjalnym hodowcom rybek akwariowych określić, jaką temperaturę powinna mieć woda w akwarium, aby mogły w niej przeżyć rybki zadanych gatunków.

Wejście

Pierwsza linia wejścia zawiera liczbę naturalną N określającą liczbę gatunków rybek ($1 \leq N \leq 50$). Kolejne N wierszy zawiera informacje dotyczące warunków temperaturowych dla kolejnych gatunków rybek (gatunki rybek ponumerowane są kolejnymi liczbami całkowitymi od 1 do N). Każdy z warunków składa się z wiersza zawierającego dwie liczby naturalne: tmin i tmax, oddzielone pojedynczą spacją, oznaczające, że dany gatunek jest w stanie przetrwać w temperaturze większej lub równej tmin i mniejszej lub równej tmax ($3 \leq t_{min} \leq t_{max} \leq 38$). Kolejny wiersz wejścia zawiera liczbę naturalną K oznaczającą liczbę zapytań ($1 \leq K \leq 1000$). W następnych K wierszach znajdują się kolejne zapytania. Każde zapytanie składa się z jednej linii, w której znajdują się liczby naturalne: m, a_1, ..., a_m, oddzielone pojedynczą spacją ($1 \leq m \leq N$; $1 \leq a_i \leq N$, dla $i = 1, 2, ..., m$). Liczba m oznacza liczbę gatunków, które hodowca chce hodować. Kolejne m różnych od siebie liczb: a_1, ..., a_m, określa numery gatunków rybek, które mają być hodowane.

Wyjście

Dla każdego zapytania, w kolejnych liniach wyjścia, należy wypisać dwie liczby *t1* i *t2* oddzielone pojedynczą spacją, gdzie $t1 \leq t2$, określające końce maksymalnego (różnica $t2 - t1$ jest maksymalna) przedziału domkniętego temperatur, w którym dane gatunki rybek są w stanie przetrwać. Jeśli taki przedział nie istnieje, należy wypisać słowo NIE.

Przykład

Dla danych wejściowych:

```
4
22 26
18 28
20 28
8 20
2
3 1 2 3
2 1 4
```

poprawną odpowiedzią jest:

```
22 26
NIE
```

Serwery

ID:1108

Limit czasu: 0.50 s

Limit pamięci: 4096 kB

W pewnej firmie informatycznej znajdują się serwery gromadzące duże ilości danych. W związku z przebudową infrastruktury technicznej podjęto decyzję o przeniesieniu i umieszczeniu danych na jednym serwerze. Każde dwa serwery połączone są bezpośrednim łączem. Każdy z serwerów w danym momencie może albo wysyłać dane tylko do jednego serwera, albo odbierać dane tylko od jednego serwera. Każdy serwer może pomieścić dane znajdujące się na wszystkich serwerach. Cała operacja przenoszenia danych powinna zostać przeprowadzona tak, aby trwała jak najkrócej. Czas przenoszenia 1MB danych pomiędzy dwoma serwerami jest stały i wynosi 1s.

Zadanie

Napisz program, który dla zadanej konfiguracji serwerów (rozmiar przechowywanych danych) wyznaczy minimalną liczbę sekund, potrzebnych do wykonania tej operacji.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych C ($1 \leq C \leq 100$). W kolejnych wierszach wejścia znajdują się zestawy danych. Każdy z C zestawów danych składa się z dwóch wierszy. Pierwszy wiersz zestawu zawiera liczbę naturalną n określającą liczbę serwerów ($1 \leq n \leq 100$). Drugi wiersz zestawu danych zawiera n liczb całkowitych: $a_1, ..., a_n$, oddzielonych pojedynczą spacją. Liczba a_i ($i = 1, ..., n$; $0 \leq a_i < 2^{31}$) określa rozmiar danych w MB przechowywanych przez i-ty serwer.

Wyjście

Dla każdego zestawu danych, w kolejnych liniach wyjścia, należy wypisać minimalną liczbę sekund potrzebnych do wykonania operacji przenoszenia danych na jeden serwer.

Przykład

Dla danych wejściowych:

```
2
4
4 10 8 3
2
1 1
```

poprawną odpowiedzią jest:

```
15
1
```

OPSSML

ID:1109

Limit czasu: 1.50 s

Limit pamięci: 32768 kB

OPSSML jest bardzo podobny do standardowego XML-a. Plik OPSSML zawiera w pierwszej linii nagłówek oraz bezpośrednio po nim w kolejnych liniach jeden element.

Elementem jest element skrócony albo element pełny.

Element skrócony zawiera kolejno:

- < - znak mniejszości
- *nazwę elementu*
- opcjonalną *listę atrybutów*
- / - znak slash
- > - znak większości
- znak nowej linii

Element pełny o nazwie X zawiera kolejno:

- < - znak mniejszości
- napis X będący nazwą elementu
- opcjonalną *listę atrybutów*
- > - znak większości
- znak nowej linii
- opcjonalną l*istę elementów*
- < - znak mniejszości
- / - znak slash
- napis X będący *nazwą elementu*
- > - znak większości
- znak nowej linii

Atrybut zawiera kolejno:

- spację
- *nazwę atrybutu*
- = - znak równości
- " - znak cudzysłowu
- *wartość atrybutu*
- " - znak cudzysłowu

Lista atrybutów jest to niepusty ciąg następujących po sobie atrybutów, z których każdy ma inną nazwę atrybutu.

Lista elementów jest to niepusty ciąg następujących po sobie elementów.

Nazwy elementów, nazwy atrybutów oraz wartości atrybutów są niepustymi ciągami co najwyżej 20 znaków: małych lub wielkich liter alfabetu angielskiego lub cyfr.

Zadanie

Napisz program, który dla zadanego pliku OPSSML:

posortuje leksykograficznie niemalejąco elementy w obrębie każdej listy elementów względem nazw elementów,

posortuje leksykograficznie niemalejąco atrybuty w obrębie każdej listy atrybutów względem nazw atrybutów.

Porządek w sortowaniu leksykograficznym określa kolejność kodów ASCII.

Wejście

Na wejściu podana jest zawartość pliku OPSSML, którego rozmiar nie przekracza 106 znaków.

Wyjście

Na wyjściu należy wypisać zawartość pliku OPSSML powstałego z pliku, którego zawartość jest podana na wejściu, wyłącznie poprzez leksykograficzne posortowanie elementów w obrębie każdej listy elementów według nazw elementów oraz poprzez leksykograficzne posortowanie atrybutów w obrębie każdej listy atrybutów według nazw atrybutów. Elementy, które w obrębie jednej listy elementów mają takie same nazwy elementów, należy wypisać w takiej kolejności, w jakiej podane są na liście elementów na wejściu. Nie należy zamieniać elementu skróconego na element pełny, ani elementu pełnego na element skrócony.

Przykład

Dla danych wejściowych:

```
<?xml version="11.1" OPSSML ?>
<root b="23" a="fg">
<yaa>
<ghi x="15">
</ghi>
<abc>
</abc>
<def/>
</yaa>
<x attr1a="value" attr1b="value" attr="0">
<tag/>
</x>
<ya>
</ya>
<x/>
```

```
</root>
```

poprawną odpowiedzią jest:

```
<?xml version="11.1" OPSSML ?>
<root a="fg" b="23">
<x attr="0" attr1a="value" attr1b="value">
<tag/>
</x>
<x/>
<ya>
</ya>
<yaa>
<abc>
</abc>
<def/>
<ghi x="15">
</ghi>
</yaa>
</root>
```

Giełda Rzeczy Wartościowych

ID:1110

Limit czasu: 1.00 s

Limit pamięci: 8192 kB

Niegdyś w Opsslandii do handlu służył wydzielony teren na otwartej przestrzeni. Dokonywano tam transakcji kupna i sprzedaży pewnych rzeczy (dóbr) wartościowych. Takie targowisko umożliwiało swobodny, bezpośredni kontakt handlowca z klientem i możliwość negocjowania ceny.

Wraz z upływem czasu oraz rozwojem Opsslandii targowisko zmieniło się w pełni skomputeryzowaną Giełdę Rzeczy Wartościowych.

Rządzi się ona następującymi regułami:

- Na Giełdzie Rzeczy Wartościowych można kupować lub sprzedawać wartościowe rzeczy. Odbywa się to poprzez składanie zleceń kupna bądź sprzedaży, z podanym limitem ceny. Osoba składająca zlecenie musi określić:
    - symbol (identyfikator) rzeczy wartościowej, którą chce kupić/sprzedać
    - rodzaj oferty (kupno lub sprzedaż)
    - liczbę rzeczy wartościowych
    - limit ceny
- Kupujący godzi się na kupno określonej rzeczy tylko po cenie, która jest nie większa od zadanego limitu ceny zlecenia kupna, zaś sprzedający godzi się na sprzedaż tylko po cenie nie mniejszej od zadanego limitu ceny zlecenia sprzedaży.
- Pomiędzy dwoma osobami (z których jedna złożyła zlecenie kupna, a druga sprzedaży) może dojść do realizacji transakcji kupna-sprzedaży po cenie na jaką się godzą.
- Jeśli kupujący kupi od sprzedającego mniejszą liczbę rzeczy od zadeklarowanej w swoim zleceniu, jego zlecenie zostaje zrealizowane częściowo (liczba rzeczy do kupna na zleceniu jest zmniejszana o liczbę rzeczy, które zostały faktycznie kupione). Jeśli kupi dokładnie taką liczbę rzeczy, jaką zadeklarował, jego zlecenie zostaje zrealizowane w całości. Analogicznie jeśli sprzedający nie sprzeda wszystkich rzeczy w jednej transakcji, jego zlecenie sprzedaży zostaje zrealizowane częściowo (liczba rzeczy do sprzedaży na zleceniu jest zmniejszana o liczbę rzeczy, które zostały sprzedane), jeśli sprzeda dokładnie taką liczbę rzeczy, jaką zadeklarował, jego zlecenie zostaje zrealizowane w całości.
- Zlecenia na daną rzecz umieszczane są w arkuszu zleceń dotyczącym tej rzeczy. Arkusz zleceń składa się z dwóch części. Lewa dotyczy zleceń kupna, prawa sprzedaży. Zlecenia ułożone są według kolejności ich ewentualnej realizacji, tzn. po stronie kupna od najwyższego do najniższego limitu ceny, po stronie sprzedaży odwrotnie. W ramach tego samego limitu zlecenia ułożone są według czasu wprowadzenia do systemu giełdowego - zlecenia wprowadzone wcześniej znajdują się powyżej zleceń wprowadzonych później. Dzięki takiemu układowi najwyższy wiersz arkusza prezentuje zawsze zlecenia posiadające najlepszy limit kupna i sprzedaży. Pozycja zlecenia w arkuszu decyduje o kolejności jego realizacji.
- Po złożeniu zlecenia system umieszcza je w arkuszu zleceń i przystępuje do realizacji wszystkich możliwych transakcji (w kolejności wynikającej z arkusza zleceń). W przypadku złożenia zlecenia kupna transakcje wykonywane są po cenie zleceń sprzedaży. Natomiast w przypadku złożenia zlecenia sprzedaży transakcje dokonywane są po cenie zleceń kupna.

Zlecenia zrealizowane w całości są usuwane z arkusza zleceń.

- Wolumen dla danej rzeczy jest to suma ilości rzeczy występująca we wszystkich dokonanych transakcjach.
- Aktualny kurs danej rzeczy jest to cena, po której dokonana została ostatnia transakcja dla tej rzeczy.

Przykład:

Jeśli arkusz zleceń dla rzeczy "A" zawiera zlecenia:

strona kupna		strona sprzedaży	
ilość	limit ceny	ilość	limit ceny
200	10.50	300	11.10
100	10.00	200	11.20

i zostanie złożone zlecenie sprzedaży 250 sztuk rzeczy "A" z limitem ceny 10.00 to zostaną wykonane 2 transakcje: sprzedaż-kupno "A": 200 sztuk po 10.50 oraz 50 sztuk po 10.00 (złożone zlecenie zostanie zrealizowane w całości, zlecenie kupna 200 sztuk po 10.50 zostanie zrealizowane w całości, zlecenie kupna 100 sztuk po 10.00 zostanie zrealizowane częściowo). Po zrealizowaniu transakcji arkusz będzie zawierał zlecenia:

strona kupna		strona sprzedaży	
ilość	limit ceny	ilość	limit ceny
50	10	300	11.10
		200	11.20

Aktualny kurs "A" to 10.00 (kurs ostatniej transakcji), aktualny wolumen to 250 sztuk.

Jeśli teraz zostanie złożone zlecenie kupna 310 sztuk rzeczy "A" z limitem ceny 11.10 zostanie wykonana 1 transakcja - 300 sztuk w cenie 11.10 a arkusz zleceń będzie zawierał:

strona kupna		strona sprzedaży	
ilość	limit ceny	ilość	limit ceny
10	11.10	200	11.20
50	10		

Aktualny kurs "A" to 11.10 (kurs ostatniej transakcji), aktualny wolumen to 550 sztuk.

Zadanie

Twoim zadaniem jest napisanie oprogramowania dla opsslandzkiej Giełdy Rzeczy Wartościowych. Program na podstawie składanych zleceń powinien wyznaczyć aktualny wolumen oraz kurs dla wszystkich rzeczy dostępnych na Giełdzie.

Wejście

Pierwsza linia wejścia zawiera liczbę N określającą liczbę złożonych zleceń ($1 \leq N \leq 100000$). Kolejne N wierszy zawiera opisy kolejno składanych zleceń. Jedno zlecenie składa się z jednego wiersza zawierającego oddzielone od siebie pojedynczą spacją: symbol rzeczy (jedna wielka litera alfabetu angielskiego), typ zlecenia (wielka litera K lub S, oznaczająca odpowiednio zlecenie kupna lub sprzedaży), liczbę naturalną L określającą liczbę rzeczy wartościowych ($1 \leq L \leq 10000$) oraz limit ceny C ($0 < C \leq 10000$). Zapis liczby C zawiera separator dziesiętny (kropkę), co najmniej jedną cyfrę przed separatorem oraz dokładnie dwie cyfry po separatorze.

Wyjście

W pierwszej linii wyjścia należy wypisać liczbę X określającą liczbę rzeczy, dla których doszło do przynajmniej jednej transakcji kupna-sprzedaży na Giełdzie. Dla każdej z tych X rzeczy, w oddzielnych liniach, należy wypisać: literę określającą symbol rzeczy, wolumen oraz aktualny kurs (oddzielone od siebie spacjami). Linie te powinny być wypisane w porządku alfabetycznym względem symbolu rzeczy. Kursy należy wypisać w takim samym formacie, w jakim podane są na wejściu, czyli z kropką, jako separatorem dziesiętnym, co najmniej jedną cyfrą przed separatorem oraz dokładnie dwoma cyframi po separatorze.

Przykład

Dla danych wejściowych:

```
8
A K 100 10.00
A K 200 10.50
A S 300 11.10
A S 200 11.20
A S 250 10.00
A K 310 11.10
B K 100 10.00
B S 50 10.00
```

poprawną odpowiedzią jest:

```
2
A 550 11.10
B 50 10.00
```

Czterysta dwadzieścia dwa

ID:1111

Limit czasu: 1.00 s

Limit pamięci: 8192 kB

Przy głównej ulicy w stolicy Opsslandii po jednej stronie mieści się pałac królewski, a po drugiej stronie, wzdłuż ulicy, w rzędzie, w równych odstępach, na N specjalnie przygotowanych miejscach pomnikowych ponumerowanych kolejnymi liczbami całkowitymi od 1 do N, stoją pomniki wszystkich N dawnych władców kraju. Zgodnie z tradycją każdy rok ma swojego patrona wybieranego co roku spośród dawnych władców Opsslandii. Wraz z nadejściem nowego roku pomnik nowego patrona, w celu uhonorowania patrona, przenoszony jest naprzeciw pałacu królewskiego na miejsce pomnika poprzedniego patrona. Nakłady pracy potrzebne do przeniesienia pomnika zależą od ciężaru pomnika i od odległości, na jaką trzeba go przenieść. Dokładniej: przeniesienie pomnika o ciężarze C z miejsca pomnikowego o numerze A na miejsce pomnikowe o numerze B wymaga C * abs(A - B) jednostek pracy, gdzie abs(x) oznacza wartość bezwzględną liczby x. Miejsce naprzeciwko pałacu musi zostać zwolnione, więc pomnik poprzedniego patrona trzeba przenieść na inne miejsce. Jednak nie zawsze zwykła zamiana miejsc pomników poprzedniego patrona i nowego patrona będzie optymalna pod względem ilości wykonanej pracy. Przeniesienie pomnika poprzedniego patrona na miejsce innego lżejszego pomnika, który stoi dość blisko, a następnie przeniesienie tego lżejszego pomnika na puste miejsce pozostawione przez pomnik nowego patrona może wymagać wykonania mniejszej pracy. Jeszcze bardziej opłacalne może być przeniesienie tego lżejszego pomnika na miejsce innego jeszcze lżejszego pomnika, itd., aż ostatni przenoszony pomnik stanie na pustym miejscu pomnikowym pozostawionym przez pomnik nowego patrona.

Zadanie

Napisz program, który wyznaczy najmniejszą liczbę jednostek pracy, jaką trzeba wykonać, aby przenieść pomniki w taki sposób, żeby pomnik nowego patrona stanął na miejscu pomnikowym znajdującym się naprzeciwko pałacu królewskiego oraz żeby na każdym miejscu pomnikowym stał dokładnie jeden pomnik.

Wejście

Pierwsza linia zawiera liczbę całkowitą D ($1 \le D \le 10$), określającą liczbę zestawów danych. W następnych liniach opisane są kolejno po sobie zestawy danych. W pierwszej linii zestawu danych znajdują się trzy liczby całkowite oddzielone spacjami: N, P oraz K ($1 \le N \le 50000$; $1 \le P, K \le N$) oznaczające odpowiednio: liczbę pomników dawnych władców Opsslandii, numer miejsca pomnikowego, na którym stoi pomnik nowego patrona, oraz numer miejsca pomnikowego znajdującego się naprzeciwko pałacu królewskiego. Druga linia zestawu danych zawiera N oddzielonych od siebie spacjami całkowitych liczb dodatnich mniejszych od 1000000, gdzie i-ta liczba w tej linii oznacza ciężar pomnika stojącego na i-tym miejscu pomnikowym.

Wyjście

W kolejnych liniach dla każdego zestawu danych należy wypisać jedną liczbę całkowitą oznaczającą najmniejszą liczbę jednostek pracy, jaką trzeba wykonać, aby przenieść pomniki w taki sposób, żeby pomnik nowego patrona stanął na miejscu pomnikowym znajdującym się naprzeciwko pałacu królewskiego oraz żeby na każdym miejscu pomnikowym stał dokładnie jeden pomnik.

Przykład

Dla danych wejściowych:

```
2
10 2 7
8 3 6 12 9 4 8 7 1 5
10 9 5
8 3 6 12 9 4 8 7 1 5
```

poprawną odpowiedzią jest:

```
37
25
```

Drogowcy

ID:1112

Limit czasu: 1.50 s

Limit pamięci: 16384 kB

Opsslandia miejscami porośnięta jest bujnymi lasami. Przez jeden z nich drogowcy muszą poprowadzić drogę. Las nie jest w każdym miejscu tak samo gęsty, dlatego wybór przebiegu drogi wpływa na liczbę wyciętych drzew podczas jej budowy. Ponieważ wszyscy mieszkańcy Opsslandii cenią sobie lasy, istotne jest, żeby wyciąć jak najmniej drzew. W celu ułatwienia zadania, drogowcy sporządzili kwadratowy plan tej części lasu, przez którą może przebiegać droga. Następnie przy użyciu siatki podzielili sporządzony plan na N * N równych kwadratowych obszarów. Jeden koniec przyszłej drogi ma być w obszarze znajdującym się w północno-zachodnim rogu siatki, a drugi koniec w obszarze znajdującym się południowo-wschodnim rogu siatki. Z pewnych względów przez dany obszar drogę można poprowadzić albo na wprost (łącząc obszar sąsiadujący od północy z obszarem sąsiadującym od południa lub łącząc obszar sąsiadujący od zachodu z obszarem sąsiadującym od wschodu), albo wytyczyć zakręt pod kątem prostym (na przykład łącząc obszar sąsiadujący od południa z obszarem sąsiadującym od zachodu lub łącząc obszar sąsiadujący od wschodu z obszarem sąsiadującym od północy). Kolejnym krokiem drogowców było wyznaczenie dla każdego obszaru liczby drzew, jaką muszą ściąć, gdyby chcieli poprowadzić przez ten obszar drogę (wyznaczona liczba drzew do ścięcia jest zawsze taka sama, niezależnie czy jest to zakręt, droga na wprost, czy jeden z końców drogi). Jednak nie tylko liczba ściętych drzew decyduje o przebiegu drogi. Najważniejsze jest, żeby droga przebiegała przez minimalną liczbę obszarów, co zdaniem drogowców spowoduje, że będzie najkrótsza. Ponadto każdy zakręt musi być dobrze oznakowany, a drogowcy dysponują znakami, które wystarczą na oznakowanie tylko K zakrętów (końce przyszłej drogi nie są zakrętami).

Zadanie

Napisz program, który wyznaczy najmniejszą liczbę drzew, jakie drogowcy muszą ściąć budując drogę, która będzie przebiegać przez minimalną liczbę obszarów i będzie zawierać nie więcej niż K zakrętów.

Wejście

Pierwsza linia zawiera liczbę całkowitą D ($1 \le D \le 10$) określającą liczbę zestawów danych. W następnych liniach opisane są kolejno po sobie zestawy danych. W pierwszej linii zestawu danych znajdują się dwie liczby całkowite rozdzielone spacją: liczba N ($2 \le N \le 100$) oraz liczba K ($1 \le K < N * N$). W każdej z kolejnych N linii zestawu danych znajduje się opis kolejnego wiersza obszarów sporządzonego planu. Wiersze obszarów planu podane są w kolejności od najbardziej północnego do najbardziej południowego. Opis pojedynczego wiersza planu składa się z N liczb całkowitych nieujemnych mniejszych od 1000 oddzielonych od siebie pojedynczymi spacjami. Kolejne liczby w opisie wiersza przyporządkowane są kolejnym obszarom sporządzonego planu (w kolejności od obszaru najbardziej zachodniego do najbardziej wschodniego). Liczba przyporządkowana obszarowi oznacza liczbę drzew, jaką muszą ściąć drogowcy, gdyby chcieli poprowadzić drogę przez ten obszar.

Wyjście

W oddzielnych liniach dla każdego zestawu danych należy wypisać jedną liczbę całkowitą oznaczającą najmniejszą liczbę drzew, jakie drogowcy muszą ściąć budując drogę, która będzie przebiegać przez minimalną liczbę obszarów i będzie zawierać nie więcej niż K zakrętów.

Przykład

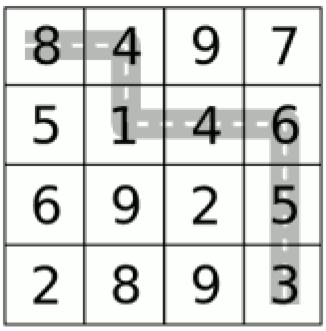
Dla danych wejściowych:

```
3
4 3
8 4 9 7
5 1 4 6
6 9 2 5
2 8 9 3
4 1
8 4 9 7
5 1 4 6
6 9 2 5
2 8 9 3
4 6
8 4 9 7
5 1 4 6
6 9 2 5
2 8 9 3
```

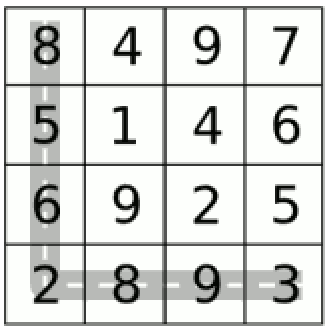
poprawną odpowiedzią jest:

```
31
41
27
```

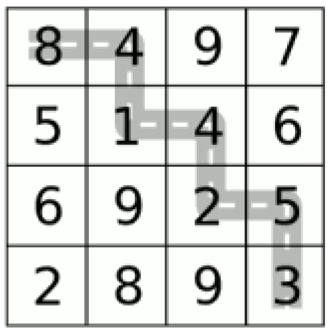
Ilustracja przykładu:



Pierwszy zestaw danych



Drugi zestaw danych



Trzeci zestaw danych

Grupy krwi

ID:1113
Limit czasu: 0.50 s

Limit pamięci: 16384 kB

Każdego lata szpitale w Opsslandii narzekają na niewystarczającą liczbę osób oddających krew. Często też brakuje jednostek krwi dla przebywających aktualnie w szpitalu pacjentów potrzebujących krwi. Problem jest jednak bardziej skomplikowany. Każdy mieszkaniec Opsslandii posiada jedną z N grup krwi numerowanych od 1 do N. Mówimy, że grupa krwi g_1 jest zgodna z grupą krwi g_2 wtedy i tylko wtedy, gdy osoba z grupą krwi g_2 może przyjąć krew grupy g_1. Każda grupa krwi jest zgodna sama ze sobą. Fakt zgodności grupy g_1 z grupą g_2 nie musi oznaczać, że grupa g_2 jest zgodna z grupą g_1.

Napisz program, który stwierdzi czy jednostki krwi występujące w szpitalu wystarczą dla wszystkich potrzebujących krwi pacjentów uwzględniając zgodność grup krwi.

Wejście

Pierwsza linia zawiera liczbę całkowitą C ($1 \leq C \leq 10$) określającą liczbę zestawów danych. W następnych liniach opisane są kolejno po sobie zestawy danych. W pierwszej linii zestawu danych znajduje się jedna liczba całkowita N ($1 \leq N \leq 100$) określająca liczbę grup krwi. W drugiej linii zestawu danych znajduje się N liczb całkowitych d_i ($0 \leq d_i \leq 1000$). Liczba i-ta w tej linii ($1 \leq i \leq N$) określa liczbę dostępnych jednostek krwi grupy o numerze i. W trzeciej linii zestawu danych znajduje się N liczb całkowitych bi ($0 \leq b_i \leq 1000$). Liczba i-ta w tej linii ($1 \leq i \leq N$) określa liczbę jednostek krwi jakie są potrzebne dla pacjentów z grupą krwi o numerze i. W linii o numerze i+3 ($1 \leq i \leq N$) zestawu danych znajduje się liczba całkowita C_i ($1 \leq C_i \leq N$), po której następuje C_i liczb całkowitych określających numery grup, z którymi zgodna jest grupa o numerze i.

Wyjście

W oddzielnych liniach dla każdego zestawu danych należy wypisać słowo TAK jeśli potrzeby pacjentów na krew da się zaspokoić przez posiadane jednostki krwi lub NIE w przeciwnym przypadku.

Przykład

Dla danych wejściowych:

```
2
4
6 2 2 4
5 3 1 5
4 1 2 3 4
3 2 3 4
2 3 4
1 4
2
1 5
2 3
```

```
2 1 2
1 2
```

poprawną odpowiedzią jest:

```
TAK
NIE
```

Suma cyfr

ID:1114

Limit czasu: 0.50 s

Limit pamięci: 16384 kB

Dla liczb całkowitych N, B ($0 \le N$; $2 \le B \le 36$), przez $R_B(N)$ oznaczmy liczbę zdefiniowaną następująco:

$$R_B(N)=\begin{cases} N, & \text{dla } 0 \le N \le B-1 \\ R_B(\text{suma cyfr liczby N w systemie o podstawie B}), & \text{dla } N \ge B \end{cases}$$

Zadanie

Dla zadanych liczb całkowitych N, B ($1 \le N \le 2^{31}-1$; $2 \le B \le 36$) obliczyć wartość $R_B(1+2+3+...+N)$.

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę całkowitą L ($1 \le L \le 1000$) - jest to liczba zestawów danych. W kolejnych L wierszach znajdują się zestawy danych. Każdy zestaw danych składa się z dwóch liczb całkowitych N, B oddzielonych pojedynczą spacją ($1 \le N \le 2^{31}-1$; $2 \le B \le 36$).

Wyjście

Dla każdego zestawu danych na wyjściu należy wypisać wartość $R_B(1+2+3+...+N)$ (w systemie o podstawie B). Cyfry większe niż 9 powinny być wypisywane dużymi literami alfabetu angielskiego ('A' zamiast 10, 'B' zamiast 11, ..., 'Z' zamiast 35).

Przykład

Dla danych wejściowych:

```
2
7 36
8 36
```

poprawną odpowiedzią jest:

```
S
1
```

Chodnik

ID:1115

Limit czasu: 0.50 s

Limit pamięci: 16384 kB

Projekty budynków opsslandzkich architektów sięgają granic możliwości ich wykonania. Skomplikowanie brył budynków stwarza liczne problemy, na które natrafiają wykonawcy projektów. Robotnicy właśnie wybudowali jeden z takich budynków, ale mają problem jego z wykończeniem. Budynek musi zostać z każdej strony otoczony przylegającym do ścian chodnikiem o szerokości jednego metra, a żeby złożyć zamówienie na materiały potrzebne do wyłożenia chodnika, trzeba znać jego powierzchnię. Zewnętrzne ściany budynku zawsze są idealnie proste, tworzą między sobą kąt prosty i mają od zewnętrznej strony całkowitą liczbę metrów długości. Dzięki temu robotnicy bez trudu naszkicowali plan budynku (tylko ściany zewnętrzne) na kartce w kratkę przyjmując skalę, w której jedna kratka na kartce odpowiada w rzeczywistości kwadratowi o powierzchni jednego metra kwadratowego. Ściany na planie tworzą łamaną zamkniętą, nie mają punktów wspólnych poza końcami ścian i pokrywają się z bokami kratek. Koniec ściany (róg budynku) jest punktem wspólnym dokładnie dwóch ścian. Pozostało im teraz policzyć wszystkie kratki na planie, na których ma znaleźć się chodnik, czyli te kratki, które są na zewnątrz budynku i sąsiadują bokiem lub rogiem z kratką, która jest wewnątrz budynku. Napisz program, który je policzy.

Wejście

Pierwsza linia zawiera liczbę całkowitą D ($1 \le D \le 10$), określającą liczbę zestawów danych. W następnych liniach opisane są kolejno po sobie zestawy danych. Jeden zestaw danych opisuje sposób, w jaki robotnicy narysowali na kartce plan budynku bez odrywania ołówka od kartki. W pierwszej linii zestawu danych znajduje się liczba całkowita N ($4 \le N \le 20000$) oznaczająca liczbę ścian budynku. Każda z kolejnych N linii zestawu danych zawiera opis kolejnego ruchu ołówka po kartce (bez odrywania go od kartki), czyli oddzielone od siebie spacją: wielką literę (kierunek rysowania ściany na planie) oraz całkowitą liczbę dodatnią (długość ściany liczoną w kratkach). Literą definiującą kierunek jest 'G', 'L', 'D' lub 'P', co oznacza odpowiednio ruch ołówka w górę, w lewo, w dół lub w prawo. Plan budynku razem z otaczającym go chodnikiem w całości mieści się na kwadratowej kartce w kratkę o rozmiarze 100000 kratek na 100000 kratek.

Wyjście

W kolejnych liniach dla każdego zestawu danych należy wypisać jedną liczbę całkowitą oznaczającą liczbę kratek na planie, na których ma znaleźć się chodnik.

Przykład

Dla danych wejściowych:

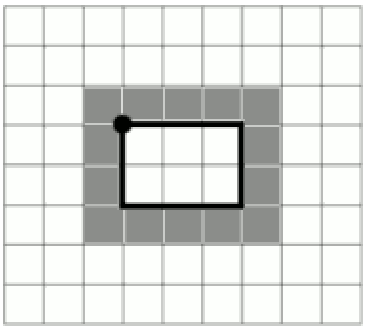
```
2
4
P 3
D 2
L 3
G 2
10
L 4
```

```
G 3
P 3
G 3
L 7
D 9
P 11
G 5
L 3
D 2
```

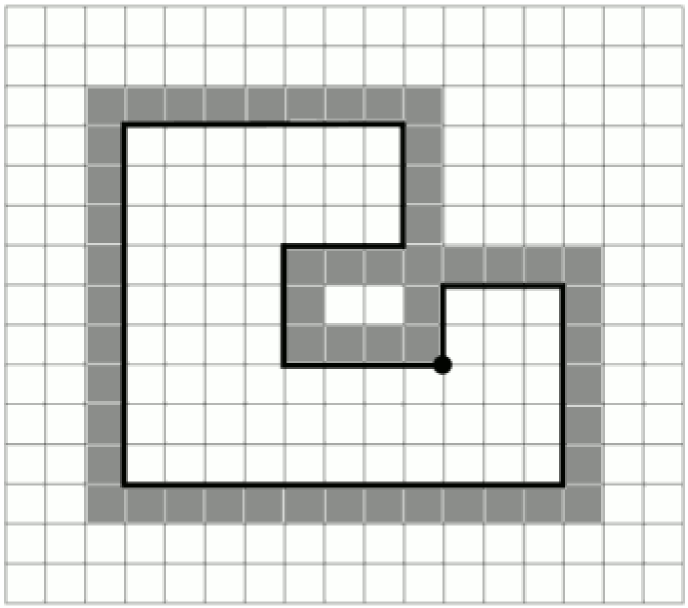
poprawną odpowiedzią jest:

```
14
53
```

Ilustracja przykładu (kropka oznacza początek rysowania planu; kratki zamalowane na szaro oznaczają chodnik):



Pierwszy zestaw danych



Drugi zestaw danych

Iloczyny

ID:1116

Limit czasu: 2.00 s

Limit pamięci: 65536 kB

Dana jest liczba naturalna n. Ile jest rożnych wyników iloczynu dwóch liczb naturalnych a i b, gdzie a, b ≤ n?

Wejście

W pierwszym wierszu wejścia znajduje się liczba C określająca liczbę zestawów danych (1 ≤ C ≤ 100). W kolejnych C wierszach wejścia znajdują się zestawy danych. Każdy z C zestawów danych składa się z jednej liczby n (1 ≤ n ≤ 100000).

Wyjście

Dla każdego zestawu danych, w osobnych liniach wyjścia, należy wypisać liczbę różnych wyników iloczynu liczb naturalnych a i b, gdzie 1 ≤ a, b ≤ n.

Przykład

Dla danych wejściowych:

```
2
5
8
```

poprawną odpowiedzią jest:

```
14
30
```

Partie

ID:1117

Limit czasu: 1.50 s

Limit pamięci: 16384 kB

W Opsslandii zbliżają się wybory parlamentarne. Każdy z tamtejszych polityków jest bardzo stanowczy i zdecydowany: albo popiera drugiego polityka, albo jest jemu przeciwny. W dodatku każdy polityk zawsze odwzajemnia swoje poparcie: popiera tylko tych polityków, którzy go popierają, oraz jest przeciwny tylko tym politykom, którzy są jemu przeciwni. Wielu polityków nie należy jeszcze do żadnej partii, więc myślą o założeniu nowej. W tym celu spotykają się z innymi politykami i wymieniając swoje poglądy rozpatrują ewentualny skład nowej partii. Prawo opsslandzkie wymaga, aby do partii należały co najmniej dwie osoby. Jednak wielkim zagrożeniem dla istnienia partii jest sytuacja rozłamowa, czyli taka, w której polityk spoza partii jest niektórym jej członkom przeciwny, a niektórych członków popiera. Sytuacja rozłamowa grozi rozpadem partii i jest powszechnie niepożądana. W celu zlikwidowania sytuacji rozłamowej należy przyjąć do partii polityków, którzy tę sytuację powodują. Wówczas, dzięki przynależności do tej samej partii, osobiste spory jej członków przestają być istotne i sytuacja rozłamowa przestaje istnieć. Życie w Opsslandii pokazuje, że im partia jest większa, tym ma większe trudności w stanowczym działaniu, więc politycy są bardziej skłonni tworzyć partie mniej liczne. Niektórzy częściowo kierują się również rankingiem zaufania do polityków i chcą wiedzieć, jaką najwyższą i jaką najniższą pozycję w rankingu zajmują politycy należący do ich partii (w rankingu zaufania znajdują się wszyscy politycy Opsslandii i nigdy nie ma w nim pozycji ex aequo). Wszystkie te kryteria powodują, że podczas spotkania dwóch polityków, którzy chcą razem utworzyć nową partię, niezwykle trudno jest im wyznaczyć jej skład.

Zadanie

Napisz program, który na podstawie danych o wzajemnym poparciu polityków oraz ich pozycjach w rankingu zaufania wyznaczy dla danych dwóch polityków najmniejszą możliwą liczbę członków partii, do której należeliby obaj politycy i która nie byłaby w sytuacji rozłamowej, a następnie poda najwyższą i najniższą pozycję w rankingu zaufania, jaką zajmowaliby politycy tej partii.

Wejście

Pierwsza linia zawiera liczbę całkowitą N ($2 \le N \le 800$) oznaczającą liczbę polityków w Opsslandii. Druga linia zawiera liczbę całkowitą K ($0 \le K \le N*(N-1)/2$). W każdej z kolejnych K linii znajdują się dwie różne oddzielone spacją liczby całkowite A i B ($1 \le A, B \le N$) oznaczające, że politycy znajdujący się w rankingu zaufania na pozycjach A i B wzajemnie się popierają. Każda z K par liczb A i B określa inną parę polityków. W następnej linii znajduje się liczba całkowita S ($1 \le S \le N*(N-1)/2$) oznaczająca liczbę spotkań. Każda z kolejnych S linii zawiera dwie różne oddzielone spacją liczby całkowite: X i Y ($1 \le X, Y \le N$) oznaczające spotkanie dwóch polityków znajdujących się w rankingu zaufania na pozycjach X i Y.

Wyjście

Dla każdego spotkania należy wypisać w jednej linii trzy liczby rozdzielone spacjami: najmniejszą możliwą liczbę członków partii, do której należeliby obaj spotykający się politycy i która nie byłaby w sytuacji rozłamowej, oraz najwyższą i najniższą pozycję w rankingu zaufania, jaką zajmowaliby politycy tej partii.

Przykład

Dla danych wejściowych:

```
11
13
1 2
2 3
3 4
4 5
5 6
3 5
4 6
7 8
7 9
7 10
8 10
9 10
10 11
4
4 5
3 6
9 11
6 7
```

poprawną odpowiedzią jest:

```
2 4 5
6 1 6
4 7 11
11 1 11
```

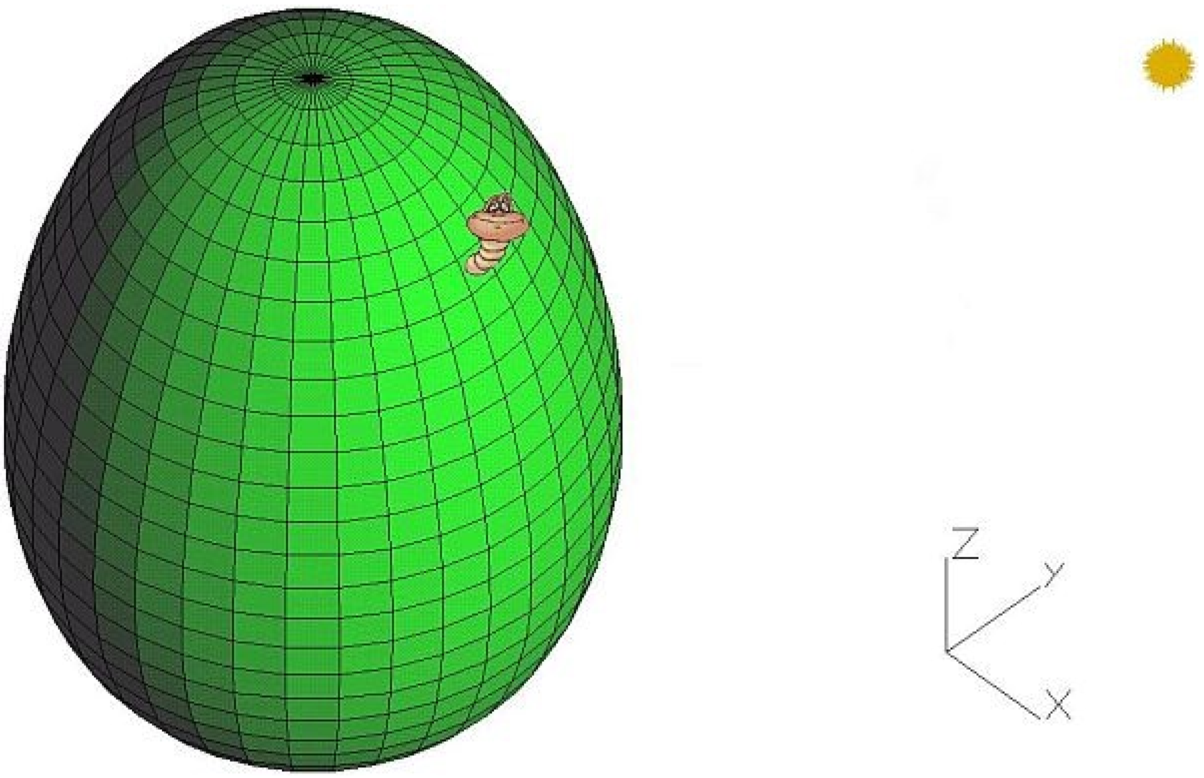
Robaczek

ID:1118

Limit czasu: 1.00 s

Limit pamięci: 16384 kB

Janek jest studentem ogrodnictwa. W czasie zajęć opiekuje się egzotycznymi drzewkami owocowymi mango. Zauważył, że szkodniki, które uszkadzają owoce mango często uszkadzają skórkę owocu w miejscach najbardziej nasłonecznionych (lub oświetlonych żarówką). Swoje obserwacje chce przekazać w czasie seminarium Koła Naukowego Ogrodników. Janek chciałby, aby ta prezentacja wyglądała bardzo efektownie - powinna być poparta jakąś "symulacja komputerową". Ponieważ nie jest on zbyt biegły w technice cyfrowej, więc poszukuje kogoś, kto potrafi wykonać taką symulację. Najistotniejszym elementem tego pokazu musi być wyznaczenie miejsca uszkodzenia owocu w zależności od jego kształtu, kierunku padania promieni i odległości od sztucznego źródła światła.



Powierzchnia owocu została opisana za pomocą małych płatów - figur płaskich (trójkąty, czworokąty) tworzących wypukły wielościan. Należy podać numer tego płata, który jest najlepiej oświetlony, czyli mówiąc ściśle, na jego powierzchni jest największe natężenie oświetlenia. Natężenie oświetlenia (fizycy mierzą je w luksach) płatu zależy od nachylenia jego powierzchni względem promieni padających. Maksymalne oświetlenie dostajemy wtedy, gdy promienie są prostopadłe do powierzchni, zerowe oświetlenie wtedy gdy promienie są równoległe lub nie docierają bezpośrednio do powierzchni, w innych przypadkach dostajemy wartości pośrednie. Natężenie oświetlenia jest odwrotnie proporcjonalne do kwadratu odległości płatu od źródła światła. W tym zadaniu należy przyjąć, dla uproszczenia, że dla wszystkich punktów płatu odległość jest taka sama i równa odległości źródła światła od środka płatu (wyznaczonego przez średnią arytmetyczną współrzędnych wierzchołków płatu). Można również założyć, że jeśli natężenie oświetlenia najlepiej oświetlonego płatu jest równe E, to natężenie oświetlenia każdego innego płatu jest mniejsze od $(1 - 10^{-11}) * E$.

Wejście

W pierwszym wierszu wejścia jest podana liczba zestawów danych L, ($1 \leq L \leq 5$). Po niej następują zestawy danych. W pierwszym wierszu zestawu danych podana jest para liczb naturalnych, oddzielona jedną spacją: N W, gdzie N jest liczbą płatów powierzchni ($4 \leq N \leq 15000$), a W liczbą punktów, które są wierzchołkami wielokątów opisujących powierzchnię owocu ($4 \leq W \leq 15000$). Drugi wiersz zestawu zawiera 3 liczby rzeczywiste oddzielone pojedynczą spacją ($-10^{10} < x,y,z < 10^{10}$), które są współrzędnymi źródła światła. Można założyć, że źródło światła jest zawsze umieszczone na zewnątrz powierzchni owocu. Następne W wierszy zestawu zawiera po 3 liczby rzeczywiste ($-10^{10} < x,y,z < 10^{10}$) oddzielone pojedynczą spacją, które są współrzędnymi wierzchołków wielościanu. Wierzchołki ponumerowane są kolejnymi liczbami naturalnymi od 1 do W w kolejności podawania ich współrzędnych. Następne N wierszy zestawu danych zawiera po 4 liczby całkowite (pierwsze trzy z nich są dodatnie, czwarta jest nieujemna), które są numerami wierzchołków należących do płatów powierzchni. Gdy czwarty numer wierzchołka jest równy 0, to płat powierzchni jest trójkątem. Gdy wszystkie 4 numery są większe od 0, to płat jest czworokątem. Numery wierzchołków są również rozdzielone pojedynczą spacją.

Wyjście

Na wyjściu, dla każdego zestawu danych, w jednym wierszu należy wypisać jedną liczbę naturalną, która jest numerem najlepiej oświetlonego płata powierzchni.

Przykład

Dla następujących danych:

```
1
4 4
6.100000E+00 5.100000E+00 4.300000E+00
1.619953E-01 8.379142E-02 7.330799E-03
1.923704E-01 2.691684E-01 2.354919E-02
3.671053E-01 3.717102E-01 3.252042E-02
7.274067E-02 3.063127E-01 2.044082E-01
2 1 3 0
1 2 4 0
3 1 4 0
3 2 4 0
```

prawidłowym rozwiązaniem jest:

```
3
```