

Zaopatrzenie (A)

Limit pamięci: 32 MB

Limit czasu: 1.00 s

Dobra impreza nie może obejść się bez odpowiednich przekąsek! Jasio, jako gospodarz, postanowił zadbać o zaopatrzenie i kupić paluszki, chipsy oraz ciastka. W kieszeni kurtki znalazł banknot o nominale 100 złotych i wyruszył na zakupy.

Jednak stanął przed trudnym wyborem, ponieważ w pobliżu jego domu znajdują się trzy sklepy: *Bitronka*, *Bitl* i *Bajtland*, a ceny tych produktów różnią się w zależności od sklepu. Dodatkowo, każdy ze sklepów posiada płatny parking, za który Jasio musi uiścić opłatę (c_i złotych dla i -tego sklepu), jeśli zdecyduje się skorzystać z jego oferty.

Napisz program, który obliczy, ile maksymalnie reszty może zachować Jasio, kupując po jednej sztuce każdego rodzaju przekąsek i płacąc za parkingi przy wybranych sklepach.

Jeśli Jasiowi nie wystarczy pieniędzy na zakupy, program powinien wypisać pojedyncze słowo NIE. Cenę benzyny na przejazd pomiędzy sklepami należy zignorować, gdyż płacą za nią rodzice Jasia.

Wejście

W pierwszym wierszu wejścia znajdują się trzy liczby naturalne pooddzielane pojedynczymi odstępami, oznaczające ceny parkingu przy kolejnych sklepach, czyli liczby c_1, c_2, c_3 . W kolejnych 3 wierszach znajdują się po 3 liczby naturalne, w i -tym z nich – ceny kolejnych produktów w złotych w i -tym sklepie ($p_{i,1}, p_{i,2}, p_{i,3}$).

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się największa możliwa reszta, jaka zostanie Jasiowi po dokonanych zakupach lub pojedyncze słowo NIE, jeśli Jasiowi nie wystarczy pieniędzy.

Ograniczenia

$$1 \leq c_i, p_{i,j} \leq 100.$$

Przykład

Wejście

```
1 99 99
20 20 20
1 1 1
1 1 1
```

Wyjście

```
39
```

Wyjaśnienie

Jasio może pojechać do Bitronki, zapłacić 1 złocisz za parking, 60 złotych za wszystkie produkty i zostanie jemu łącznie 39 złotych

Wejście

```
15 10 5
15 35 35
35 15 35
35 35 15
```

Wyjście

```
25
```

Wyjaśnienie

Jasio powinien udać się do każdego ze sklepów, w Bitronce kupić paluszki, w Bitlu chipsy, a w Bajtlandzie ciastka. Zapłaci wtedy $15 + 10 + 5 + 15 + 15 + 15 = 75$ złotych, zatem zostanie jemu 25 złotych reszty.

Żabie skoki (B)

Limit pamięci: 64 MB

Limit czasu: 1.00 s

óznym wieczorem goście Jasia znaleźli w oczku wodnym gospodarza żabę. Ich widok wyrwał ją z letargu, a chcąc rozruszać stawy, zaczęła wykonywać skoki – zawsze dokładnie w lewo lub w prawo. Nie miała ochoty obracać się w innych kierunkach. W pierwszej minucie skoczyła o 1 centymetr, w drugiej o 2 centymetry, w trzeciej o 3 centymetry i tak dalej.

Zachowanie żaby wzbudziło duże zainteresowanie wśród zgromadzonych. Zaczęli się zastanawiać, ile najmniej czasu zajęłoby jej dotarcie do punktu oddalonego o K centymetrów w prawo od początkowej pozycji.

Uwaga! Żaba wykonuje skok w każdej minucie – nie może pozostać na miejscu.

Rozwiąż zagadkę kolegów Jasia! Napisz program, który wyznaczy minimalną liczbę minut potrzebną żabie na dotarcie do określonego miejsca.

Wejście

W pierwszym (jedynym) wierszu wejścia znajduje się jedna liczba naturalna K , odległość w centymetrach od jej miejsca startowego.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna znaleźć się jedna liczba oznaczająca minimalną liczbę minut potrzebną żabie na dotarcie do wyznaczonego miejsca. Jeśli żaba nie może dotrzeć do celu, zamiast liczby wypisz pojedyncze słowo NIE.

Ograniczenia

$$1 \leq K \leq 10^{18}.$$

Przykład

Wejście

6

Wyjście

3

Wyjaśnienie

Za każdym razem żaba skacze w prawo, $1 + 2 + 3 = 6$. Dotrze na miejsce po trzech minutach. Można udowodnić że nie jest tego w stanie zrobić w mniejszej liczbie ruchów.

Wejście

8

Wyjście

4

Wyjaśnienie

Pierwszy skok żaba wykonuje w lewo, a każdy następny w prawo, $-1 + 2 + 3 + 4 = 8$. Dotrze na miejsce po czterech minutach. Można udowodnić że nie jest tego w stanie zrobić w mniejszej liczbie ruchów.

Upalny dzień (c)

Limit pamięci: 256 MB

Limit czasu: 2.00 s

Przyjaciel Jasia, Tomek, po udanej imprezie obudził się w samo południe na środku polany. Niepewny wydarzeń minionej nocy, zaczął zastanawiać się, jak wrócić do domu. Jednak żar lejący się z nieba i obawa przed poparzeniami słonecznymi skutecznie zniechęcały go do drogi. Na szczęście na łące rośnie n drzew, a każde z nich rzuca choć odrobinę cienia.

Dla uproszczenia przyjmujemy, że i -te drzewo rzuca cień będącym kołem o promieniu r_i oraz środku w (x_i, y_i) . Cienie drzew mogą na siebie nachodzić lub siebie wzajemnie zawierać. Zakładamy również, że pnie drzew są nieskończenie cienkie i nie utrudniają poruszania się Tomka, który w naszych rozważaniach jest reprezentowany jako punkt.

Tomek może poruszać się w dowolnym kierunku z prędkością 1 jednostki odległości na sekundę i w każdej chwili zmieniać kierunek ruchu. Tomek obudził się w punkcie (x_s, y_s) , a jego dom znajduje się w punkcie (x_e, y_e) .

Napisz program, który wyznaczy minimalny czas (w sekundach), jaki Tomek musi spędzić na słońcu, aby dotrzeć do domu.

Wejście

W pierwszym (jedynym) wierszu wejścia znajdują się cztery liczby pooddzielane pojedynczymi odstępami oznaczające kolejno x_s, y_s, x_e, y_e . Drugi wiersz wejścia zawiera pojedynczą liczbę n .

W następnych n wierszach wejścia znajdują się po 3 liczby całkowite, w i -tym z nich x_i, y_i, r_i oznaczające współrzędne środka oraz promień cienia i -tego drzewa.

Wszystkie liczby na wejściu są całkowite.

Wyjście

W pierwszym (i jedynym) wierszu wyjścia powinna znaleźć się jedna liczba rzeczywista, oznaczająca minimalny czas, jaki Tomek musi spędzić na słońcu, aby dotrzeć do domu.

Twoja odpowiedź zostanie zaakceptowana, jeśli błąd względny lub bezwzględny nie przekroczy 10^{-6} .

Ograniczenia

- $-10^9 \leq x_s, y_s, x_e, y_e \leq 10^9$,
- Punkty (x_s, y_s) i (x_e, y_e) nie pokrywają się,
- $1 \leq n \leq 1000$,
- $-10^9 \leq x_i, y_i \leq 10^9$,
- $1 \leq r_i \leq 10^9$.

Przykład

Wejście

```
7 0 -1 3
3
0 3 1
4 0 1
6 0 1
```

Wyjście

```
3.0000000000
```

Wyjaśnienie

Rysunek poniżej.

Wejście

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

Wyjście

```
1.6568542495
```

Wyjaśnienie

Rysunek poniżej.

Wejście

1 0 6 3

5

1 1 1

2 2 1

3 3 1

4 3 1

5 3 1

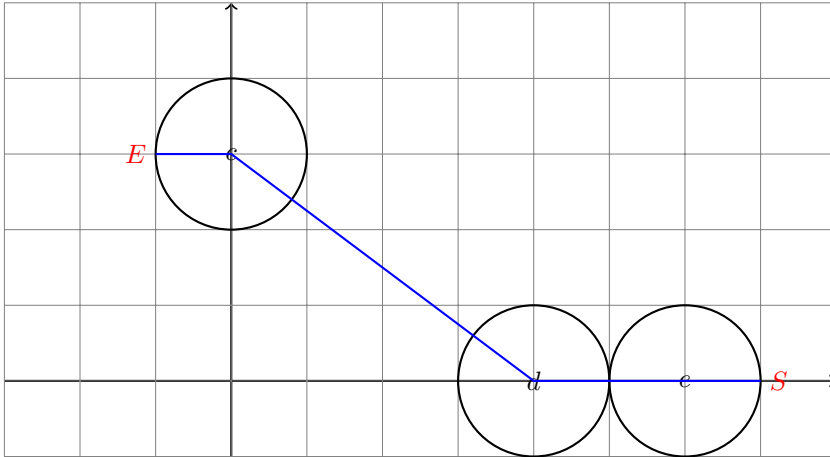
Wyjście

0.0000000000

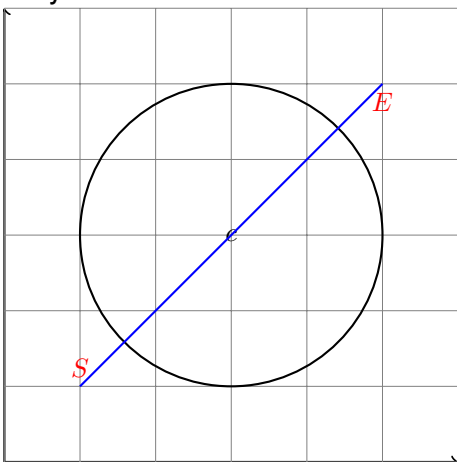
Wyjaśnienie

Rysunek poniżej.

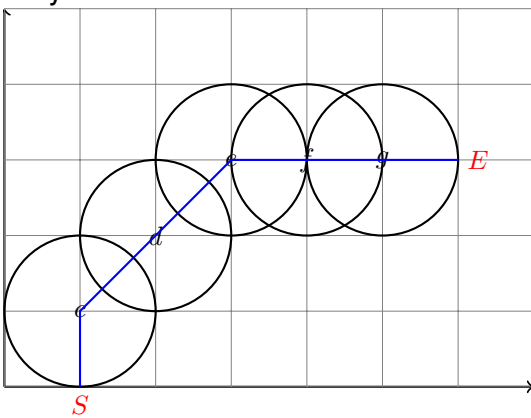
Przykład 1.



Przykład 2.



Przykład 3.



Przepowiednia (D)

Limit pamięci: 32 MB

Limit czasu: 0.50 s

Jedną z atrakcji, jakie Jasio chce przygotować na przyjęcie, jest przepowiadanie przyszłości. Planuje sprawdzić, jak układają się fusy herbaty na dnie filiżanki każdego z gości, zbadać linie życia na ich dłoniach oraz określić ich znaki zodiaku. Choć Jasio dobrze przygotował się z wróżbiarstwa, wciąż ma problem z kalendarzem i odczytywaniem dat.

Pomóż zapracowanemu gospodarzowi i napisz dla niego program, który na podstawie dnia i miesiąca urodzin każdego z gości wypisze jego znak zodiaku, zgodnie z poniższym kalendarium astrologicznym. Każdemu ze znaków przypisany jest domknięty przedział dat:

Baran: 21 marca - 19 kwietnia
Byk: 20 kwietnia - 20 maja
Bliźnięta: 21 maja - 20 czerwca
Rak: 21 czerwca - 22 lipca
Lew: 23 lipca - 22 sierpnia
Panna: 23 sierpnia - 22 września
Waga: 23 września - 22 października
Skorpion: 23 października - 21 listopada
Strzelec: 22 listopada - 21 grudnia
Koziorożec: 22 grudnia - 19 stycznia
Wodnik: 20 stycznia - 18 lutego
Ryby: 19 lutego - 20 marca

Wejście

W pierwszym (jedynym) wierszu wejścia znajduje się dzień i miesiąc urodzin jednego z gości Jasia w formacie: dd.mm

Wyjście

Na wyjściu należy wypisać pojedyncze słowo będące nazwą znaku zodiaku odpowiadającego podanym na wejściu dniowi oraz miesiącowi, zgodnie z powyżej opisanym kalendarium astrologicznym.

Uwaga! Nazwę należy wypisać małymi literami alfabetu angielskiego, tak jak w przykładach.

Ograniczenia

Podany na wejściu dzień i miesiąc odpowiadają dowolnemu istniejącemu dniowi roku.

Przykład

Wejście

16.06

Wyjście

bliznieta

Wejście

01.01

Wyjście

koziorozec

Wejście

01.02

Wyjście

wodnik

Łańcuch ozdobny (E)

Limit pamięci: 64 MB

Limit czasu: 1.00 s

Do dekoracji domu Jasia przydałby się także ładny, ozdobny łańcuch. Jasio nie lubi chaosu i braku ładu, a taki łatwo otrzymać rozwieszając łańcuch z oczkami w wielu kolorach, z których żaden się przebija. Dlatego Jasio chciałby, aby najczęściej występujący kolor oczek w jego łańcuchu, występował na ponad połowie oczek. Gospodarzowi szkoda złoćszy na stworzenie nowego łańcucha, dlatego chciałby wykorzystać stary, znaleziony w piwnicy. Pomóż Jasiowi, napisz dla niego program, który sprawdzi czy ze starego łańcucha da się wyciąć kawałek o pożądanej przez Jasia własności. **Uwaga!** Jedno oczko to żaden łańcuch, dlatego Jasia interesują fragmenty złożone z **co najmniej dwóch** oczek.

Wejście

W pierwszym wierszu wejścia znajduje się pojedyncza liczba naturalna n , oznaczająca liczbę oczek starego łańcucha znalezionego przez Jasia w piwnicy. W drugim wierszu wejścia znajduje się jedno słowo s składające się z n ($s_1, s_2, s_3 \dots s_n$) małych znaków alfabetu angielskiego. Różne litery odpowiadają różnym kolorom oczek łańcucha.

Wyjście

W pierwszym (jedynym) wierszu wyjścia należy wypisać dwie liczby naturalne $l < r$, takie, że fragment starego łańcucha s_l, s_{l+1}, \dots, s_r zawiera co najmniej t wystąpień pewnego koloru, gdzie $t > \lfloor \frac{r-l+1}{2} \rfloor$. Możesz wypisać indeksy odpowiadające dowolnemu z takich fragmentów. Jeśli taki fragment nie istnieje, zamiast tego należy wypisać pojedyncze słowo NIE.

Ograniczenia

$1 \leq n \leq 200000$.

Przykład

Wejście

5
solve

Wyjście

NIE

Wyjaśnienie

Nie istnieje fragment łańcucha o pożądanej własności.

Wejście

6
ananas

Wyjście

1 5

Wyjaśnienie

Fragment anana zawiera 3 literki a, co stanowi ponad połowę jego długości.

Wejście

8
accepted

Wyjście

2 4

Wyjaśnienie

Fragment cce zawiera 2 literki c, co stanowi ponad połowę jego długości.

Dziwna Czekolada (F)

Limit pamięci: 64 MB

Limit czasu: 1.00 s

Jedną z bardziej zaskakujących rzeczy na imprezie Jasia okazała się być jego *dziwna czekolada*. Każdy, kto kiedyś jadł czekoladę wie, że jej poszczególne kawałki zwykle smakują tak samo, albo chociaż bardzo podobnie. Z czekoladą Jasia było jednak inaczej. Jej prostokątny kształt został pocięty poziomo na paski o wysokościach A_1, A_2, \dots, A_k oraz pionowo, na paski o szerokościach B_1, B_2, \dots, B_k , linie podziału przebiegają równoległe do brzegów czekolady. W ten sposób czekolada została pocięta na k^2 prostokątnych kostek. Im większy kawałek tym lepszy. Ponadto smak kawałka zależy w dziwny sposób od jego pozycji na oryginalnej tabliczce czekolady. Smakowitość kawałka powstałego na przecięciu pasków A_i oraz B_j wyraża bowiem się dziwnym wzorem: $A_i \cdot B_j \cdot (i \oplus j)$, gdzie \oplus oznacza operację xor.

Pomóż Jasiowi oraz jego gościom ustalić, jaka jest sumaryczna smakowitość wszystkich kawałków powstałych po pocięciu czekolady.

Wejście

W pierwszym wierszu wejścia znajduje się pojedyncza liczba naturalna k oznaczająca liczbę pasków pionowych i poziomych w podziale czekolady. Drugi wiersz wejścia zawiera k liczb naturalnych, kolejno: A_1, A_2, \dots, A_k podzielanych pojedynczymi odcinkami. Są to wysokości kolejnych poziomych pasków podziału czekolady. Analogicznie trzeci wiersz wejścia zawiera k liczb naturalnych B_1, B_2, \dots, B_k , reprezentujących kolejne szerokości pionowych pasków podziału czekolady.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna znaleźć się suma smakowitości wszystkich kawałków powstałych przez pocięcie czekolady.

Ograniczenia

$1 \leq k \leq 100\,000, 1 \leq A_i, B_i \leq 1\,000\,000$.

Zagwarantowane jest, że sumaryczna smakowitość kawałków czekolady nie przekracza 10^{18} .

Przykład

Wejście

2
1 2
3 4

Wyjście

30

Wyjaśnienie

$(1 \cdot 3 \cdot (1 \oplus 1)) + (2 \cdot 3 \cdot (2 \oplus 1)) =$
 $0 + (2 \cdot 3 \cdot 4) = 18$
 $(1 \cdot 4 \cdot (1 \oplus 2)) + (2 \cdot 4 \cdot (2 \oplus 2)) =$
 $(1 \cdot 4 \cdot 4) + 0 = 12$
 $18 + 12 = 30$

Wejście

3
1 3 2
2 1 3

Wyjście

46

Wyjaśnienie

Kolejne smakowitości (w wierszach)
wynoszą: 0, 18, 8; 3, 0, 2; 6, 9, 0;.
Sumarycznie: 46

Dywan (G)

Limit pamięci: 32 MB

Limit czasu: 2.00 s

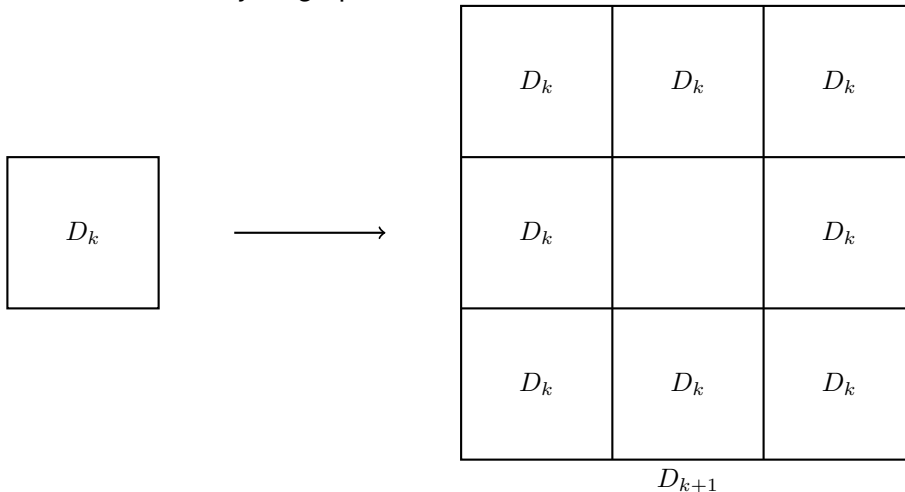
Jasio, przygotowując się na przyjęcie gości, postanowił udekorować swój dom. Przeglądając internet, natrafił na fraktal o ciekawym wyglądzie – dywan Sierpińskiego. Wpadł na pomysł, aby wydrukować go na plakatach i przykleić do ścian. Niestety, jego stara drukarka obsługuje wyłącznie standardowe znaki ASCII.

Pomóż Jasiowi i napisz program, który wczyta stopień fraktalu i wydrukuje go przy użyciu jedynie znaków spacji oraz #.

Dywan stopnia 0 (D_0) to pojedynczy znak #.

Dywan stopnia $k + 1$ (D_{k+1}) otrzymujemy, układając **osiem kopii** dywanu stopnia D_k w układzie **3×3**, pozostawiając środkowy fragment pusty.

Graficzna ilustracja tego procesu:



Puste miejsca w trakcie konstrukcji kolejnych stopni fraktalu należy wypełnić spacjami (patrz przykłady).

Wejście

W pierwszym (jedynym) wierszu wejścia znajduje się pojedyncza liczba całkowita n , oznaczająca poziom fraktalu, jaki należy wypisać.

Wyjście

Na wyjściu powinien znaleźć się dywan sierpińskiego według powyżej opisanych reguł i konwencji.

Ograniczenia

$0 \leq n \leq 7$.

Przykład

Wejście

0

Wyjście

#

Wyjaśnienie

D_0 reprezentujemy poprzez pojedynczy znak #.

Wejście

1

Wyjście

###

Wyjaśnienie

Pojedyncze zagnieżdżenie procedury rysującej dywan da wskazany rezultat.

Wejście

Wyjście

Wyjaśnienie


```
#####  
# # # # #  
#####  
###   ###  
# #   # #  
###   ###  
#####  
# # # # #  
#####
```

Dwukrotne zagnieżdzenie procedury
rysującej dywan da wskazany rezultat.