



Wujek Janusz zamierza dać Jasiowi na urodziny cztery odcinki o całkowitych długościach. Spośród wszystkich figur geometrycznych Jaś najbardziej lubi równoległoboki i będzie bardzo zawiedziony, jeśli z otrzymanych odcinków nie uda mu się żadnego zbudować. Niestety, Janusz dowiedział się o tym już po zapakowaniu prezentu. Napisz program, który sprawdzi, czy przygotowany przez wujka prezent ucieszy siostrzeńca.

WEJŚCIE

W jednym wierszu wejścia znajdują się cztery liczby naturalne A, B, C, D , pooddzielane pojedynczymi odstępami, oznaczające długości odcinków w prezencie przygotowanym przez Janusza.

WYJŚCIE

Należy wypisać jeden wiersz zawierający słowo TAK, jeżeli istnieje równoległobok, który ma boki o długościach podanych na wejściu (ich kolejność nie ma znaczenia), lub NIE w przeciwnym wypadku.

OGRANICZENIA

$1 \leq A, B, C, D \leq 100$.

PRZYKŁADY

Wejście

4 4 4 4

Wyjście

TAK

Z odcinków o takich długościach można zbudować kwadrat.

Wejście

6 5 4 3

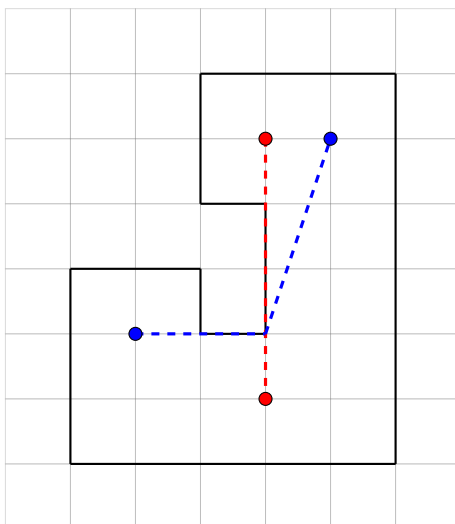
Wyjście

NIE



Jasio od małego fascynował się grami komputerowymi. Nic dziwnego, że kiedy tylko nauczył się programować, zaczął tworzyć swoją własną grę. Gra Jasia składa się z wielu poziomów. Celem w każdym z nich jest przejście z punktu początkowego do punktu końcowego, pokonując po drodze przeciwników. Obszar, po którym może poruszać się gracz jest wielokątem prostym, czyli wielokątem bez dziur, którego boki się nie przecinają. Dodatkowo, każdy bok wielokąta jest prostopadły do osi OX lub OY.

Jasio planuje przypisać każdemu poziomowi stopień trudności, który w dużej mierze zależy od tego, jak długą drogę musi przebyć gracz. Konkretniej, do wyznaczenia stopnia trudności potrzebna jest znajomość minimalnego dystansu jaki należy pokonać, żeby dostać się z punktu początkowego do punktu końcowego, nie wychodząc poza dozwolony obszar. Jasio nie jest zbyt dobry z algorytmiki, więc zwrócił się do Ciebie o pomoc w obliczeniu potrzebnych danych. Co więcej, nie jest jeszcze pewien gdzie umieści punkty początkowe i końcowe, więc potrzebne będzie obliczenie dystansów dla kilku możliwych scenariuszy. Poniżej znajduje się ilustracja drugiego testu przykładowego.



Napisz program, który wczyta opis poziomu i zapytania w postaci pary punktów, dla każdego zapytania wyznaczy minimalny dystans między punktem początkowym a końcowym i wypisze go na standardowe wyjście.

WEJŚCIE

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N , oznaczająca liczbę wierzchołków w wielokącie prostym, który przedstawia dozwolony obszar. W następnych N wierszach znajdują się współrzędne wierzchołków wielokąta, w postaci pary liczb całkowitych X_i, Y_i , oddzielonych pojedynczym znakiem odstępu. Boki wielokąta to odcinki pomiędzy każdymi dwoma kolejnymi punktami, a także między pierwszym a ostatnim. Możesz założyć, że żadne trzy kolejne punkty nie są współliniowe.

W kolejnym wierszu znajduje się jedna liczba naturalna Q , oznaczająca liczbę zapytań. W każdym z następnych Q wierszy znajdują się cztery liczby całkowite Sx_i, Sy_i, Ex_i, Ey_i , porozdzielane pojedynczymi znakami odstępu, oznaczające współrzędne punktu początkowego i końcowego. W każdym zapytaniu punkt początkowy jest różny od punktu końcowego oraz żaden z nich nie leży na boku wielokąta.

WYJŚCIE

Na standardowe wyjście należy wypisać Q wierszy. W i -tym wierszu należy wypisać odpowiedź na i -te zapytanie – liczbę rzeczywistą oznaczającą minimalny dystans między podanymi punktami. Żeby odpowiedź została uznana za poprawną, jej błąd względny lub bezwzględny nie może przekroczyć 10^{-6} .

OGRANICZENIA

$1 \leq N, Q \leq 200, |X_i|, |Y_i|, |Sx_i|, |Sy_i|, |Ex_i|, |Ey_i| \leq 1\,000.$

PRZYKŁAD

Wejście	Wyjście
8	280.0108692442
0 0	171.0119109505
100 0	136.0548925101
100 100	10.0000000000
0 100	139.0164287382
0 98	144.0153179986
98 98	290.0103090044
98 2	
0 2	
7	
6 1 6 99	
24 99 99 1	
84 1 72 99	
99 50 99 60	
99 53 4 99	
99 52 4 1	
1 1 1 99	
Wejście	Wyjście
10	5.1622776602
2 2	4.0000000000
1 2	
1 4	
4 4	
4 -2	
-1 -2	
-1 1	
1 1	
1 0	
2 0	
2	
0 0 3 3	
2 3 2 -1	



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

Jasio traktuje bezpieczeństwo w sieci bardzo poważnie. Kiedy potrzebuje nowego hasła, zapisuje na kartce niepusty ciąg liczb. Następnie dopisuje na końcu ciągu sumę kontrolną – sumę wszystkich wcześniejszych elementów. Oczywiście Jasio nie ma szans zapamiętać tak wygenerowanego hasła, dlatego zawsze przepisuje je z kartki.

Dzisiaj rano brat Jasia znalazł kartkę z hasłem i dla żartów dopisał trochę liczb na początku i na końcu ciągu. Teraz biedny Jasio nie może zalogować się na `solve.edu.pl`, a turniej programistyczny już się zaczął! Czy pomożesz mu odzyskać hasło? Wystarczy, że podasz liczbę spójnych podciągów ciągu z kartki, które mogły być jego hasłem. Jeżeli identyczne hasło wystąpiło wielokrotnie w różnych miejscach ciągu, to każde takie wystąpienie zliczamy osobno.

Napisz program, który wczyta ciąg liczb znajdujący się na kartce, wyznaczy liczbę spójnych podciągów spełniających warunki bycia hasłem Jasia i wypisze wynik na standardowe wyjście.

WEJŚCIE

W pierwszym wierszu wejścia znajduje się liczba N , oznaczająca liczbę elementów ciągu na kartce. W drugim wierszu wejścia znajduje się N liczb całkowitych A_1, A_2, \dots, A_n , poroździelanych pojedynczymi znakami odstępu – ciąg z kartki Jasia.

WYJŚCIE

Na standardowe wyjście należy wypisać jedną liczbę całkowitą – liczbę spójnych podciągów, które mogły być hasłem wygenerowanym przez Jasia.

OGRANICZENIA

$$1 \leq N \leq 100\,000, |A_i| \leq 10^9.$$

PRZYKŁAD

Wejście
5
-1 1 0 0 0

Wyjście
6

Hasłem mogły być podciągi $(-1, 1, 0)$, $(-1, 1, 0, 0)$, $(-1, 1, 0, 0, 0)$, $(0, 0, 0)$ oraz $(0, 0)$ występujący dwa razy.

Wejście
4
2 4 4 10

Wyjście
2



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

W Instytucie Informatyki UWr jest duży wybór kursów związanych z algorytmiką. W tym semestrze student Adam zapisał się na jeden z nich. Na pierwszych zajęciach prowadzący dał im następujące zadanie:

„Dany jest las o N wierzchołkach etykietowanych liczbami od 1 do N składający się z drzew ukorzenionych. W każdym kroku wybieramy jeden liść i usuwamy go z grafu. To znaczy, możemy wybrać dowolny wierzchołek, który ma stopień 1 i nie jest korzeniem lub ma stopień 0. Wykonujemy N takich kroków zapisując etykiety kolejno usuwanych wierzchołków. Wypisz p -tą najmłodszą leksykograficznie permutację, którą można uzyskać w ten sposób.”

Na następnych zajęciach K studentów dostarczyło prowadzącemu rozwiązania. Po uruchomieniu ich na pewnym teście otrzymał K permutacji. Nie wszystkie rozwiązania były poprawne, więc permutacje niekoniecznie były identyczne. Niestety, prowadzący zapomniał jaki był to test i próbuje go sobie przypomnieć. Na szczęście ma permutacje wygenerowane przez programy studentów. Wiadomo, że każda z nich jest możliwa do uzyskania algorytmem z zadania, lecz niekoniecznie jest p -ta w kolejności leksykograficznej. Pomóż prowadzącemu. Napisz program, który wyznaczy, ile co najmniej było drzew w tym teście.

WEJŚCIE

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne N i K oznaczające odpowiednio wielkość testu oraz liczbę studentów. W następnych K wierszach znajdują się N -elementowe permutacje wyznaczone przez programy studentów.

WYJŚCIE

Należy wypisać jedną liczbą naturalną, minimalną liczbę drzew ukorzenionych, z których mógł się składać las, dla którego można wygenerować permutacje podane na wejściu.

OGRANICZENIA

$1 \leq N \leq 300, 1 \leq K \leq 20$.

PRZYKŁAD

Wejście

4 2
1 2 3 4
2 1 4 3

Wyjście

2

W przykładowym lesie składającym się z dwóch drzew wierzchołki 4 i 3 byłyby korzeniami, a 2 i 1 synami 4.

Wejście

3 2
1 2 3
3 2 1

Wyjście

3

Istnieje tylko jeden las, dla którego mogły być wygenerowane te permutacje.



Ministerstwo
Cyfryzacji

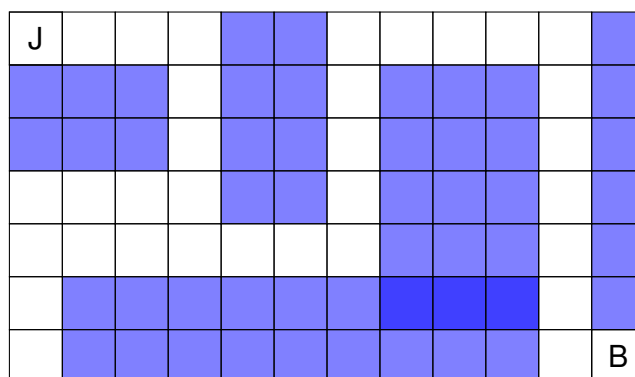


Uniwersytet
Wrocławski

Jasio po wielu latach uczenia się informatyki stwierdził, że to nie dla niego, po czym wstąpił do armii i został saperem. Przed chwilą został zrzucony z helikoptera na teren wroga, w celu rozbrojenia zakopanych tutaj min. Zaraz, zaraz, coś tu nie gra. O zgrozo, Jasio zostawił w bazie swój wykrywacz metalu! Spokojnie, to jeszcze nie czas na panikę. Wprawdzie teren jest zaminowany, ale nie w całości. W plecaku Jasia znajduje się mapa, na której zaznaczone są zaminowane obszary. Przemieszczanie się poza tymi obszarami jest całkowicie bezpieczne i Jasio planuje w ten sposób wrócić do bazy. Pytanie, czy taka bezpieczna trasa w ogóle istnieje.

Mapa dzieli teren na $X \cdot Y$ sektorów w kształcie jednostkowych kwadratów, ułożonych w X kolumn i Y wierszy. Kolumny ponumerowane są liczbami od 1 do X , od zachodu do wschodu, a wiersze liczbami od 1 do Y , od północy do południa. Sektor położony w i -tej kolumnie oraz j -tym wierszu oznaczamy przez (i, j) .

Jasio znajduje się początkowo w sektorze $(1, 1)$, a baza w sektorze (X, Y) . Każdy zaminowany obszar jest prostokątem złożonym z zaminowanych sektorów. Możesz założyć, że sektory $(1, 1)$ i (X, Y) są bezpieczne, czyli nie znajdują się w żadnym takim prostokącie. Z sektora (i, j) Jasio może przedostać się do sektorów $(i + 1, j)$, $(i - 1, j)$, $(i, j + 1)$, $(i, j - 1)$, oczywiście pod warunkiem, że są bezpieczne. Teren leżący poza mapą nie jest bezpieczny. Poniżej znajduje się rysunek, przedstawiający mapę z pierwszego testu przykładowego.



Napisz program, który ustali, czy Jasio ma szansę dostać się do bazy. W tym celu wczyta wymiary mapy i zaminowane obszary, sprawdzi czy istnieje bezpieczna trasa między pozycją Jasia a bazą i wypisze odpowiedź na standardowe wyjście.

WEJŚCIE

W pierwszym wierszu wejścia znajdują się trzy liczby naturalne X , Y , N , porozdzielane pojedynczymi znakami odstępu, oznaczające kolejno liczbę kolumn, wierszy i zaminowanych obszarów. W następnych N wierszach opisane są zaminowane obszary. Każdy z nich składa się z czterech liczb naturalnych x_i, y_i, x'_i, y'_i , porozdzielanych pojedynczymi znakami odstępu. Wszystkie sektory (i, j) dla $x_i \leq i \leq x'_i$ i $y_i \leq j \leq y'_i$ są zaminowane.

WYJŚCIE

Na standardowe wyjście należy wypisać słowo TAK, jeżeli istnieje bezpieczna trasa między pozycją Jasia a bazą. W przeciwnym wypadku należy wypisać słowo NIE.

OGRANICZENIA

$2 \leq X, Y \leq 10^9$, $1 \leq N \leq 2\,000$.

PRZYKŁAD

Wejście

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

Wyjście

TAK

Wejście

3 3 1
1 2 3 2

Wyjście

NIE



Nikt nie lubi rozwiązywać zadań z geometrii, a zwłaszcza na zawodach programistycznych. Tegoroczne obozy informatyczne dla uczniów organizowane przez Uniwersytet Wrocławski mają to zmienić. Aby przyszli studenci nabrali większej wprawy w tej dziedzinie, codziennie rozwiązują przynajmniej jedno zadanie z geometrii.

Jasio miał dzisiaj do rozwiązania bardzo skomplikowane zadanie. Napisał trzy wersje rozwiązania, jedno dla trójkąta ostrokątnego, drugie dla prostokątnego, a trzecie dla rozwartokątnego. Pomóż mu i napisz program, który znając długości boków trójkąta, przyporządkuje go do jednego z powyższych zbiorów.

WEJŚCIE

W jednym wierszu wejścia są trzy liczby naturalne A, B, C , długości boków pewnego trójkąta.

WYJŚCIE

Należy wypisać jeden wiersz zawierający literę O, P lub R, jeżeli trójkąt o długościach boków A, B, C jest odpowiednio ostro-, prosto- lub rozwartokątny.

OGRANICZENIA

$1 \leq A, B, C \leq 100$.

PRZYKŁAD

Wejście

3 3 3

Wyjście

O

Wejście

4 5 3

Wyjście

P

Wejście

3 3 5

Wyjście

R



Jasio gra w *Dziel i zwyciężaj*. Zasady są bardzo proste. Na początku przeciwnik dysponuje armią N żołnierzy. W każdej turze Jasio może wykonać jedną z dwóch operacji: zaatakować zmniejszając liczebność armii przeciwnika o jeden lub podzielić liczebność armii przeciwnika przez dowolny właściwy dzielnik.

Na przykład po podzieleniu 12 przez 4 przeciwnikowi zostaje 3 żołnierzy. Poza tym można podzielić 12 przez 3 lub 2, ale nie można przez 12 (ten dzielnik nie jest właściwy). Gdy przeciwnik straci wszystkich żołnierzy, gra kończy się, a Jasio wygrywa. Jasio bardzo lubi zwyciężać, ale jeszcze niezbyt dobrze umie dzielić. Pomóż mu i napisz program, który obliczy, w jakiej najmniejszej liczbie ruchów może osiągnąć upragnione zwycięstwo.

WEJŚCIE

W jedynym wierszu wejścia znajduje się jedna liczba naturalna N , liczba żołnierzy w armii przeciwnika.

WYJŚCIE

Wypisz jedną liczbę naturalną, minimalną liczbę ruchów potrzebną do wyeliminowania wszystkich żołnierzy przeciwnika.

OGRANICZENIA

$$1 \leq N \leq 10^9$$

PRZYKŁAD

Wejście

8

Wyjście

3

W pierwszym ruchu należy podzielić 8 przez 4, a w dwóch następnych odejmować jeden (atakować).

Wejście

9

Wyjście

4

Wystarczy podzielić 9 przez 3, a następnie trzykrotnie eliminować po jednym żołnierzu.