

Klucz

Napisać program sprawdzający czy dany klucz pasuje do danego zamka. Dziurka w zamku reprezentowana jest w postaci tablicy zero-jedynkowej i jest spójna. Klucz zakodowany jest jako ciąg par liczb całkowitych, każda reprezentująca punkt należący do klucza jako współrzędne w pewnym układzie kartezjańskim (patrz przykładowe dane). Oś x oraz y tego układu skierowane są odpowiednio w prawo oraz do góry.

Wejście

Dane wejściowe rozpoczynają nieujemne liczby całkowite N oraz M (mniejsze od 2^{16}) określające rozmiar zamka ($N \times M$). Każda z kolejnych M linii zawiera ciąg 0-1 długości N . Następnie podany jest klucz zakodowany w postaci par liczb. Każda kolejna linia zawiera parę liczb całkowitych x , y , współrzędne punktu należącego do klucza, poczynając od lewego-górnego punktu klucza o współrzędnych 0 0.

Wyjście

Na wyjściu należy wypisać TAK jeśli klucz pasuje do podanego zamka. W przeciwnym przypadku należy wypisać NIE.

Przykładowe dane

Wejście:

```
9 7
000000000
000010000
000111000
001111000
000110000
011110000
000011000
0 0
1 1
1 0
1 -1
2 1
2 0
2 -1
2 -2
3 0
3 -1
4 0
4 -1
4 -2
4 -3
5 1
5 0
```

Wyjście:

TAK

Autobusy

W pewnym mieście jest N przystanków autobusowych (ponumerowanych od 1 do N) oraz M (również ponumerowanych) linii autobusowych zakodowanych w postaci tras (ciągów przystanków). Zakładamy, że każdy autobus porusza się w obu kierunkach. Napisać program sprawdzający, czy z przystanku p do przystanku q można dojechać z co najwyżej jedną przesiadką.

Wejście

Dane wejściowe rozpoczynają nieujemne liczby całkowite N oraz M (mniejsze od 2^{16}) oznaczające odpowiednio: liczbę przystanków oraz liczbę linii autobusowych. Następnie, w każdej z kolejnych M linii podane są trasy jako ciągi przystanków. Po podaniu wszystkich tras każda kolejna linia zawiera parę przystanków (dwie liczby od 1 do N) dla których szukamy połączeń. Wejście kończy pusty wiersz.

Wyjście

Dla każdej pary przystanków należy wypisać w osobnej linii TAK jeśli można dojechać z co najwyżej jedną przesiadką, bądź NIE jeśli tego warunku nie spełniono.

Przykładowe dane

Wejście:

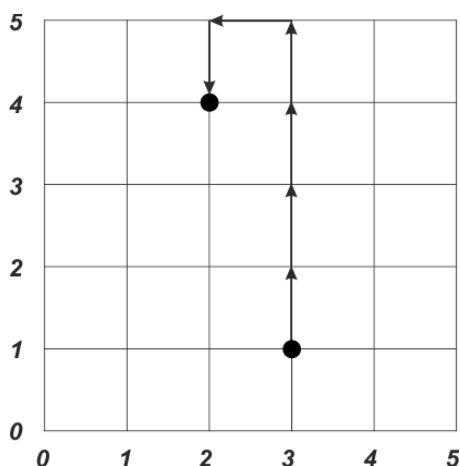
6 3
1 3 5
2 4 5
1 3 6
1 2
4 6
3 2

Wyjście:

TAK
NIE
TAK

Liczba dróg

Dana jest kratownica o rozmiarze $N \times N$ (gdzie $0 < N < 2^{16}$). Krawędzie kratownicy indeksowane są od zera do $N-1$. Dla danych dwóch punktów $P1=(x1,y1)$, $P2=(x2,y2)$ znajdź ile istnieje dróg z $P1$ do $P2$ o długości M . Poruszać można się tylko wzdłuż kratownicy, zaś każdy jej wierzchołek można odwiedzić tylko raz (punkty $P1$ uważa się za odwiedzone, kolejność odwiedzanych wierzchołków ma znaczenie).



Rysunek powyżej przedstawia jedną z możliwych dróg o długości $M=6$ z punktu $(3,1)$ do punktu $(2,4)$ na kratownicy $N=6$.

Wejście

Dane wejściowe rozpoczyna liczba N określająca rozmiar kratownicy. Dwie kolejne linie zawierają współrzędne punktów $P1$ oraz $P2$, każde w postaci dwóch liczb $0 < x, y < N$ oddzielonych spacją. Ostatnia linia zawiera liczbę M określającą długość drogi.

Wyjście

Na wyjściu należy wygenerować liczbę określającą liczbę możliwych dróg o długości M z punktu $P1$ do $P2$ na kratownicy o wymiarze $N \times N$.

Przykładowe dane

Wejście:

4
1 1
3 1
4

Wyjście:

6

Podróż

Podróżujący chce się dostać z punktu A do B w jak najkrótszym czasie. Do dyspozycji ma samochód z określoną ilością paliwa w baku. Samochód może poruszać się tylko z dwiema prędkościami: 90km/h i 120 km/h. Przy prędkości 90km/h spalanie wynosi 9 litrów na 100km, przy prędkości 120km/h 15 litrów na 100km. Zakładamy, że samochód nie potrzebuje czasu na rozpędzenie i hamowanie, a czas poruszania się z daną prędkością musi być wielokrotnością minuty. Należy podać najkrótszy czas (w minutach) po jakim podróżujący dotrze do celu.

Uwaga: Jeśli w baku jest za mało paliwa by przebyć dany dystans to wypisujemy 0. Jeśli czas po którym podróżujący dotrze do celu nie jest wielokrotnością minuty to wynik zaokrąglamy w górę do pełnej minuty (np. 121 min 34 s zaokrąglamy do 122 min).

Wejście

W każdej linii znajduje się liczba kilometrów do przejechania oraz liczba litrów paliwa jaka znajduje się w baku (liczby naturalne mniejsze od 2^{16} rozdzielone spacją). Dane kończą się pustą linią.

Wyjście

Dla każdej linii wejściowej należy w linii wyjściowej podać najkrótszy czas (w minutach), po jakim podróżujący dotrze do celu.

Przykładowe dane

Wejście:

900 80
900 81
1200 180
2100 261
1200 200

Wyjście:

0
600
600
1200
600

Piec

Restauracja Calabria, która częstowała pizzą uczestników III Warmińsko-Mazurskich zawodów w programowaniu, jest wyposażona w automatyczny piec do wypiekania ciast. Piec ma na wejściu taśmę z ułożonymi ciastami (każde ciasto ma swój czas wypiekania). Piec mieści naraz M sztuk ciast. Twoim zadaniem jest tak podzielić ciasta na kolejne wkłady do pieca aby zminimalizować czas wypiekania. Kolejny wkład można umieścić w piecu tylko jak skończą się wypiekać wszystkie ciasta z poprzedniego. Nie możemy zmieniać kolejności ciast na taśmie, możemy za to dowolnie dzielić ciasta na kolejne wkłady (maksymalnie M kolejnych ciast). Trzeba obliczyć i wyświetlić najkrótszy możliwy czas wypiekania.

Przykładowo, piec mieści dwa ciasta ($M=2$). Na taśmie mamy kolejne czasy wypiekania: 2 10 10. Najkrótszy całkowity czas będzie wynosił 12. Pierwszy wkład – pierwsze ciasto, drugi wkład – równocześnie ciasto drugie i trzecie. Gdybyśmy podzielili ciasta inaczej, to czas wypiekania byłby 20.

Wejście

Wejście zaczyna się od liczby całkowitej N ($N < 23$) określającej liczbę przypadków. Każdy przypadek składa się z dwóch linii. W pierwszej linii znajduje się liczba M (liczba ciast mieszczących się w piecu) oraz liczba K , długość taśmy ($K < 1001$). W drugiej linii jest K liczb naturalnych, oznaczających czasy wypiekania poszczególnych ciast na taśmie. Każdy czas wypiekania nie przekracza 100.

Wyjście

Dla każdego przypadku należy obliczyć i wyświetlić w jednej linii liczbę całkowitą, najkrótszy możliwy czas wypiekania podanej kolejki.

Przykładowe dane

Wejście:
2
2 3
2 10 10
2 3
7 5 2

Wyjście:
12
9

Stacje benzynowe

Danych jest N stacji benzynowych ponumerowanych od 1 do N . Z każdej stacji można dojechać do następnej tzn. z 1 do 2, z 2 do 3... z $N-1$ do N oraz z N do 1 (stacje tworzą cykl zamknięty). Posiadasz samochód z pustym bakiem, do którego możesz tankować nieograniczoną ilość paliwa. Niestety, z racji kryzysu na każdej stacji dostępnej benzyny jest bardzo mało. Wiesz, ile paliwa potrzebujesz aby dojechać do następnej stacji. Twoim zadaniem jest znaleźć wszystkie stacje startowe, które pozwalają wykonać pełen cykl tzn. takie, w których zaczynając ruch masz pewność, że wystarczy Ci paliwa aby przejechać całą trasę.

Wejście

Wejście zaczyna się od liczby stacji benzynowych N ($0 < N < 2^8$). W kolejnych dwóch liniach podane jest po N liczb naturalnych (mniejszych od 2^{16}) oddzielonych spacją: w pierwszej są to zapasy paliwa na poszczególnych stacjach, w kolejnej koszty dojazdu do następnej stacji.

Wyjście

Wypisać (osobno w każdej linii) kolejne numery stacji spełniających warunek z zadania. W przypadku braku rozwiązania wypisać 0.

Przykładowe dane

Wejście:

3
21 10 31
31 9 22

Wyjście:

2