

# Bond

---

Limit pamięci: 32MB

Poniżej widzisz kod programu, który na ekranie wypisuje komunikat: "HELLO WORLD!". Zmodyfikuj treść programu tak, aby wypisywał w pierwszej linii komunikat "My name is Bond.", zaś w drugiej "James Bond."

```
#include<iostream>
using namespace std;
int main()
{
    cout << "HELLO WORLD!";
    return 0;
}
```

## Wejście

Twój program nie powinien oczekiwać żadnych danych.

## Wyjście

W pierwszej linii wypisz komunikat: "My name is Bond.", zaś w drugiej "James Bond."

# Klasy

---

Dostępna pamięć: 256MB

W liceum w Bajtomiu przyjęto nowych uczniów do trzech klas pierwszych. Zapamiętaj liczby uczniów w każdej klasie, a później je wypisz.

Wejście

W pierwszej linii wejścia znajdują się trzy liczby całkowite  $a$ ,  $b$  oraz  $c$  ( $1 \leq a, b, c \leq 50$ ), odpowiednio liczba uczniów w klasie  $a$ ,  $b$  i  $c$ .

Wyjście

W pierwszym wierszu wyjścia wypisz liczby uczniów w klasach  $a$ ,  $b$  i  $c$ . W kolejnych trzech liniach wypisz nazwy klas (mała litera) oraz (po odstępach) liczbę uczniów w każdej z klas.

Przykład

<b>Wejście</b> 12 34 23	<b>Wyjście</b> 12 34 23 a 12 b 34 c 23
----------------------------	--

## Obwód trójkąta

---

Dostępna pamięć: 32MB

Dla danych długości boków trójkąta oblicz jego obwód.

Wejście

W pierwszym wierszu wejścia znajduje się trzy liczby całkowite  $a$ ,  $b$  i  $c$  ( $1 \leq a, b, c \leq 10^9$ ) – długości boków trójkąta. Możesz założyć, że z podanych długości boków zawsze będzie można zbudować trójkąt.

Wyjście

Długość obwodu trójkąta.

Przykład

Wejście 4 3 2	Wyjście 9
------------------	--------------

# Stopnie

---

Limit pamięci: 64MB

Napisz program, który dla podanej temperatury w stopniach Fahrenheita wypisze temperaturę w stopniach Celsjusza.

Możesz wykorzystać wzór:  $^{\circ}\text{C} = 5 / 9 ( ^{\circ}\text{F} - 32 )$

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę całkowitą  $f$  ( $-459 \leq f \leq 10^9$ ) – temperaturę w stopniach Fahrenheita.

Wyjście

Na wyjściu wypisz temperaturę w stopniach Celsjusza zaokrągloną do 2 miejsc po przecinku.

Przykład

Wejście 32	Wyjście 0.00
---------------	-----------------



# Szachy – Liczby pół

Limit pamięci: 64MB

Wiele gier korzysta z planszy mającej kształt kwadratu i złożonej z równej ilości wierszy i kolumn. Pola są malowane naprzemiennie w kolorach jasnym i ciemnym. Plansza taka nazywana jest szachownicą. Czy zastanawiałeś się kiedyś, ile pół na takiej szachownicy jest jasnych, a ile ciemnych?

Wejście

Na standardowym wejściu znajduje się jedna liczba całkowita  $n$  określająca rozmiar szachownicy ( $1 \leq n \leq 10^3$ ).

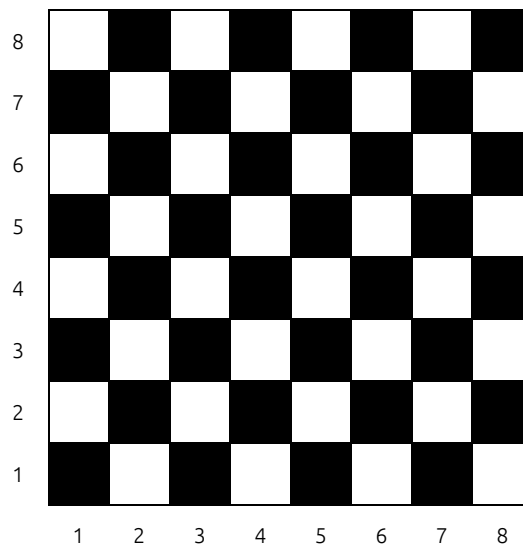
Wyjście

Na wyjściu w pierwszym wierszu wypisz liczbę pół jasnych, w drugim – ciemnych. Zakładamy, że pole o współrzędnych  $[1, 1]$  jest polem ciemnym.

Przykład

Wejście 8	Wyjście 32 32
--------------	---------------------

Ilustracja przykładu:



# Łamanie czekolady

---

Limit pamięci: 64MB

Pan Integer kupił swoją ulubioną czekoladę z nadzieniem toffi. Czekolada ma kształt prostokąta o rozmiarze  $n$  na  $m$  kawałków. Pan Integer chciałby teraz odłamać jednym ruchem dokładnie  $k$  kawałków. Czy jest to możliwe?

Wejście

Pierwszy wiersz wejścia zawiera dwie liczby całkowite  $n$  oraz  $m$  – rozmiar czekolady ( $1 \leq n, m \leq 10^6$ ). W kolejnej linii znajduje się całkowita liczba  $k$  kawałków czekolady, które chce odłamać pan Integer ( $1 \leq k \leq 10^6$ ).

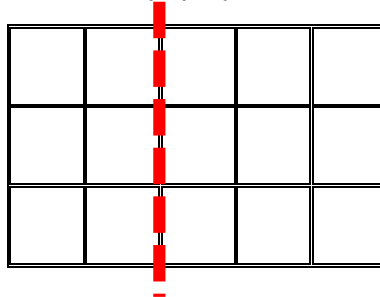
Wyjście

Na wyjściu wypisz odpowiedź na pytanie, czy pan Integer może jednym przełamaniem oderwać  $k$  kawałków czekolady (TAK lub NIE).

Przykład

Wejście 3 5 6 Wyjście TAK	Wejście 4 8 6 Wyjście NIE
------------------------------------	------------------------------------

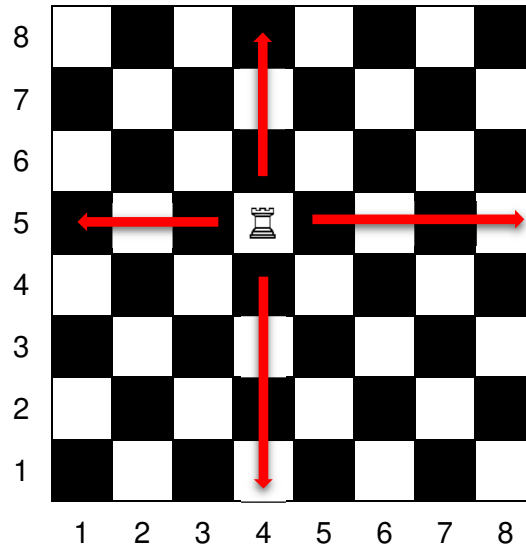
Ilustracja przykładu 1:



## Szachy – C. Wieża

Limit pamięci: 64MB

Wieża po szachownicy przesuwa się pionowo lub poziomo. Znając początkowe położenie wieży określ, czy w jednym ruchu można przenieść ją na wybrane pole.



Wejście

Pierwszy wiersz wejścia zawiera dwie oddzielone spacją liczby naturalne  $x_1$  oraz  $y_1$  ( $1 \leq x_1, y_1 \leq 8$ ) – współrzędne pola, na którym stoi wieża. Drugi wiersz zawiera dwie liczby naturalne  $x_2$  oraz  $y_2$  ( $1 \leq x_2, y_2 \leq 8$ ) – współrzędne pola, na które chcemy przesunąć wieżę.

Wyjście

Wypisz informację, czy z pola  $[x_1, y_1]$  można przesunąć wieżę na pole  $[x_2, y_2]$  w jednym ruchu: TAK lub NIE.

Przykład

Wejście 4 5 5 4	Wyjście NIE
-----------------------	----------------

## Dwójki

---

Limit pamięci: 64MB

Dana jest liczba naturalna  $x$  ( $x < 10^{18}$ ). Dla podanego  $x$  wypisz liczbę potęg 2 mniejszych bądź równych  $x$ .

Przykład

Wejście 10	Wejście 64
Wyjście 4	Wyjście 7

## Litery w liczbie

---

Limit pamięci: 64MB

Janek bada właściwości liczb i ostatnio szuka takich liczb, które zapisane w postaci szesnastkowej zawierają co najmniej jedną literę.

Wejście

Pierwszy wiersz danych zawiera liczbę naturalną mniejszej od trylionu.

Wyjście

Wypisz komunikat TAK, jeśli w postaci szesnastkowej występują litery lub NIE – jeśli ich brak.

Wejście	Wejście
54	255
Wyjście	Wyjście
NIE	TAK

## Małpki na wybiegu

---

Dostępna pamięć: 32MB

Mały Bitek trenuje małpki. Umie już je ustawić w jednym rzędzie. Planuje teraz kolejne sztuczki. Na noc jednak Bitek musi schować małpki na wybiegu. Pomóż mu!

Wejście

W pierwszym i jedynym wierszu wejścia znajduje się jedna liczba całkowita  $n$  ( $1 \leq n \leq 10^3$ ) – liczba małpek.

Wyjście

Wypisz  $n$  znaków @ obok siebie otoczonych płotkiem ze znaków #.

Przykład

Wejście	Wyjście
5	##### #@#@#@#@# #####

# Szachownica

---

Dostępna pamięć: 32MB

Napisz program, który dla podanej na standardowym wejściu liczby całkowitej  $n$ , narysuje szachownicę z cyfr 0 i 1 o boku  $n$ .

Wejście

Jedyny wiersz danych zawiera liczbę całkowitą  $n$  ( $1 \leq n \leq 200$ ).

Wyjście

Program powinien wypisać szachownicę o wielkości  $n$ .

Przykład

Wejście	Wyjście
5	01010 10101 01010 10101 01010

## Najbliższy palindrom

---

Dostępna pamięć: 32MB

Mały Bitek przeczytał na Wikipedii: Palindrom (gr. palindromeo – biec z powrotem) – wyrażenie brzmiące tak samo czytane od lewej do prawej i od prawej do lewej. Janek szybko ułożył sobie jeden palindrom (biorąc pod uwagę tylko litery): „U Izydy żądze na wyrku co noc ukrywane, zdąży Dyziu?”. Potem zaczął wymyślać całą masę następnych. Niestety – dużo gorzej mu idzie z liczbami. Chciałby szybko zamienić dowolną liczbę (jeśli nie jest ona palindromem) na najbliższy większy od niej palindrom. Czy mu pomożesz?

Wejście

Pierwszy wiersz danych zawiera liczbę całkowitą  $k$  ( $10 \leq k \leq 1\,000\,000$ ) do zamiany na palindrom.

Wyjście

Program powinien wypisać w jedną liczbę całkowitą – najmniejszą możliwą liczbę nieujemną, którą należy dodać do danej liczby, aby otrzymać palindrom.

Przykład

Wejście 150 Wyjście 1	Wejście 55555 Wyjście 0	Wejście 142 Wyjście 9
--------------------------------	----------------------------------	--------------------------------





Zadaniem Twojego programu będzie wypisanie wszystkich naturalnych dzielników zadanej liczby.

### Zadanie

Napisz program, który:

- wczyta ze standardowego wejścia liczbę naturalną  $n$ ,
- wypisze na standardowe wyjście wszystkie dzielniki liczby  $n$  uporządkowane rosnąco.

### Wejście

Jedna liczba naturalna  $1 \leq n \leq 10^9$ .

### Wyjście

W  $i$ -tym wierszu wyjścia należy wypisać  $i$ -ty z kolei dzielnik liczby  $n$ .

### Przykład

Dla danych wejściowych:

12

poprawną odpowiedzią jest:

1  
2  
3  
4  
6  
12

## Podzielne

---

Dostępna pamięć: 32MB

Dane są liczby naturalne  $a$  i  $b$ . Wypisz, ile liczb w przedziale jest podzielnych przez 3 lub przez 5.

Wejście

W pierwszej linii wejścia znajdują się dwie liczby całkowite  $a$  i  $b$  ( $1 \leq a \leq b \leq 10^{15}$ ).

Wyjście

Na standardowym wyjściu należy wypisać jedną liczbę całkowitą.

Przykład

Wejście 25 75	Wyjście 24
------------------	---------------

# Przenoszenie

---

Dostępna pamięć: 32MB

Jasio uczy się dodawać wielocyfrowe liczby od prawej do lewej, po jednej cyfrze. Dla Jasia operacja przeniesienia, podczas której jedynka jest przenoszona z jednej pozycji do następnej, stanowi poważne wyzwanie. Twoim zadaniem jest policzenie, ile operacji przeniesienia wystąpi w każdym z dodawań w danym zestawie. Pomoże to Jasiowi w oszacowaniu trudności zadań.

## Wejście

W pierwszej linii wejścia znajduje się liczba  $n$  ( $1 \leq n \leq 10^6$ ) – liczba zestawów testowych. W każdym z  $n$  następujących wierszy znajdują się po dwie liczby całkowite bez znaku, każda z nich ma mniej niż 18 cyfr.

## Wyjście

Dla każdego z  $n$  zestawów liczb wypisz liczbę operacji przeniesienia występujących podczas dodawania dwóch liczb.

## Przykład

Wejście	Wyjście
3	0
234 342	3
654 456	1
191 111	

# Ścieżka Sterna-Brocota

Limit pamięci: 32MB

Drzewo Sterna-Brocota to drzewo binarne zawierające wszystkie dodatnie ułamki nieskracalne. Struktura ta posiada wiele ciekawych właściwości. Jeśli liczby  $a$  oraz  $b$  są względnie pierwsze, to ułamek  $\frac{a}{b}$  występuje w drzewie dokładnie jeden raz. Ponadto każdą liczbę rzeczywistą dodatnią możemy zapisać jako ciąg symboli L oraz P tak, że początkowe fragmenty tego ciągu symbolizują liczby wymierne przybliżające tę liczbę. Na przykład liczbę  $\frac{5}{7}$  opiszemy jako LPPL.

Zaczynamy od  $\frac{0}{1}$  symbolizującego zero i  $\frac{1}{0}$  symbolizującego nieskończoność. Następnie na kolejnych piętrach drzewa wpisujemy „pomiędzy” wartości  $\frac{a}{b}$  oraz  $\frac{c}{d}$  wartość  $\frac{a+c}{b+d}$ .

Naszymi wartościami startowymi jest  $\frac{0}{1}, \frac{1}{1}, \frac{1}{0}$ .

Zatem w pierwszym kroku mamy:

$\frac{0}{1}$		$\frac{1}{1}$		$\frac{1}{0}$
---------------	--	---------------	--	---------------

W drugim kroku:

$\frac{0}{1}$		$\frac{1}{2}$		$\frac{1}{1}$		$\frac{2}{1}$		$\frac{1}{0}$
---------------	--	---------------	--	---------------	--	---------------	--	---------------

W trzecim:

$\frac{0}{1}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{1}$	$\frac{3}{2}$	$\frac{2}{1}$	$\frac{3}{1}$	$\frac{1}{0}$
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

Zaś w czwartym:

$\frac{0}{1}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{3}{5}$	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{1}{1}$	$\frac{4}{3}$	$\frac{3}{2}$	$\frac{5}{3}$	$\frac{2}{1}$	$\frac{5}{2}$	$\frac{3}{1}$	$\frac{4}{1}$	$\frac{1}{0}$
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

Napisz program, który czyta dwie liczby  $m$  oraz  $n$  i wypisuje ścieżkę Sterna-Brocota.

Wejście

Jedyny wiersz danych zawiera dwie *względnie pierwsze* liczby całkowite naturalne  $m$  i  $n$  ( $1 \leq m, n \leq 10^5$ ).

Wyjście

Program powinien wypisać ścieżkę z drzewa Sterna-Brocota.

Przykład

Wejście 5 3	Wejście 7 2
Wyjście PLP	Wyjście PPPL

# Temperatury

---

Dostępna pamięć: 256MB

Bajtek na lekcji przyrody uczy się określać różne zjawiska pogodowe. Jednym z zadań, które otrzymał, jest codzienne notowanie temperatury. Bajtek rzetelnie zapisywał liczby na karteczce każdego dnia. Teraz przygląda się uzyskanym wynikom i zastanawia, ile razy i w które dni uzyskał określoną temperaturę. Pomożesz mu?

## Wejście

W pierwszej linii wejścia znajduje się jedna liczba całkowita  $n$  ( $1 \leq n \leq 10^3$ ), oznaczająca liczbę dni pomiarowych. W drugiej linii wejścia znajduje się  $n$  liczb całkowitych - zanotowanych w kolejnych dniach temperatur  $t_i$  ( $-50 \leq t_i \leq 100$ ). W trzeciej linii znajduje się jedna liczba  $x$  ( $-50 \leq x \leq 100$ ) - szukana temperatura.

## Wyjście

Na wyjściu w jednej linii powinna znaleźć się liczba dni  $k$ , w które została zmierzona temperatura  $x$ , oraz  $k$  liczb oznaczających numery dni, w które zmierzono temperaturę  $x$ . Liczby należy podać w kolejności rosnącej i oddzielić pojedynczymi spacjami.

## Przykład

Wejście	Wyjście
5	2 1 4
-2 0 1 -2 3	
-2	

## Najlepsze sumy

---

Dostępna pamięć: 32MB. Na podstawie zadania cke.gov.pl

Najlepszą sumą ciągu liczb  $a_1, a_2, \dots, a_n$  nazywamy największą wartość wśród sum złożonych z sąsiednich elementów tego ciągu. Na przykład dla ciągu: 1, 2, -5, 7 mamy następujące sumy:

1  
 $1+2 = 3$   
 $1+2+(-5) = -2$   
 $1+2+(-5)+7 = 5$   
2  
 $2+(-5) = -3$   
 $2+(-5)+7 = 4$   
-5  
 $-5+7 = 2$   
7

Zatem najlepszą sumą jest 7 (zwróć uwagę, że jeden element też uznajemy za sumę).

Zaproponuj algorytm wyznaczania najlepszej sumy dla dowolnego ciągu liczb całkowitych. Na jego podstawie napisz program do obliczenia najlepszych sum ciągów liczb.

Wejście

W kolejnych liniach znajduje się ciąg liczb całkowitych zakończony liczbą 0. Liczb jest nie więcej niż 500000, ich wartość mieści się w przedziale  $[-1000, 1000]$ .

Wyjście

Największa suma złożona z kolejnych elementów ciągu.

Przykład

Wejście	Wyjście
1	7
2	
-5	
7	
0	

# Bankiet

---

Dostępna pamięć: 32MB. Źródło: I OIG (dostępne w serwisie szkopol.edu.pl)

W restauracji Utalentowany Miś zaplanowano bankiet dla finalistów OIG. Goście zasiądą przy okrągłych stołach w ściśle określony sposób. Kierownik sali otrzymał listę gości wraz z informacją, kto ma siedzieć z lewej strony każdego z nich. Ile stołów musi przygotować na bankiet?

## Zadanie

Opracuj program, który:

- wczyta ze standardowego wejścia informacje o rozmieszczeniu gości,
- obliczy ile stołów trzeba przygotować,
- wypisze wynik na standardowe wyjście.

## Wejście

W pierwszym wierszu zapisano liczbę gości  $N$  ( $1 \leq N \leq 30\,000$ ). Goście są ponumerowani kolejnymi liczbami naturalnymi od 1 do  $N$ . W drugim wierszu zapisano numer gościa siedzącego po lewej stronie pierwszego gościa. W trzecim wierszu zapisano numer gościa siedzącego po lewej stronie drugiego gościa itd. W  $i$ -tym wierszu zapisano numer gościa siedzącego po lewej stronie  $(i-1)$ -tego gościa. W  $N+1$ -szym wierszu zapisano numer gościa siedzącego po lewej stronie  $N$ -tego gościa.

## Wyjście

W pierwszym wierszu wypisz liczbę stolików potrzebnych do usadzenia wszystkich gości.

## Przykład

Wejście	Wyjście
12	4
4	
10	
7	
3	
2	
6	
1	
5	
11	
8	
12	
9	

# Pociąg

Dostępna pamięć: 64MB

„Stoi na stacji lokomotywa,  
Ciężka, ogromna i pot z niej spływa -  
Tłusta oliwa.  
[...]  
A tych wagonów jest ze czterdzieści,  
Sam nie wiem, co się w nich jeszcze mieści.  
Lecz choćby przyszło tysiąc atletów  
I każdy zjadłby tysiąc kotletów,  
I każdy nie wiem jak się natężał,  
To nie udźwigną - taki to ciężar!”

*Julian Tuwim, Fragment wiersza 'Lokomotywa'*

Jaś pracuje na rampie kolejowej i przeładowuje wagony. Wiedząc, jak ciężkie mogą być wagony, postanowił zakupić specjalny dźwig. Może on podnieść dowolną liczbę wagonów jednocześnie, ważne, żeby stały one obok siebie. Niestety, nawet taki super-sprzęt posiada maksymalny udźwig. Jak duży ciężar musi podnieść dźwig Jasia?

## Wejście

W pierwszym wierszu wejścia znajduje się liczba wagonów  $n$  pewnego pociągu ( $2 \leq n \leq 10^6$ ).  
W drugiej linii znajduje się  $n$  liczb całkowitych  $w_i$  – ciężar każdego z wagonów ( $1 \leq w_i \leq 10^6$ ).  
W trzeciej linii znajduje się jedna liczba całkowita  $k$  – liczba zestawów wagonów ( $1 \leq k \leq 10^6$ ).  
W kolejnych  $k$  liniach znajdują się informacje o zestawach wagonów, które należy podnieść dźwigiem, odpowiednio numer pierwszego  $w_p$  i ostatniego wagonu  $w_k$  do podniesienia (należy jednocześnie podnieść wszystkie wagony od  $w_p$  do  $w_k$  włącznie).

## Wyjście

Twój program powinien w  $k$  liniach ciężar każdego z zestawów wagonów.

## Przykład

Wejście	Wyjście
5	9
3 5 4 5 4	13
2	
2 3	
3 5	



# Bitib, Bajtjab i Palindrom

---

Dostępna pamięć: 64MB.

Bitib i Bajtjab lubią takie ciągi liczb, które obaj mogą czytać jednocześnie - Bitib od pierwszej do ostatniej, zaś Bajtjab od ostatniej do pierwszej, a mimo to czytają to samo. Czy liczby na kartce, którą właśnie znaleźli, można tak poprzestawiać, aby otrzymać ulubiony przez chłopców sposób ich ustawienia?

## Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna  $n$  nie większa niż milion. W następnej linii znajduje się  $n$  liczb naturalnych nie większych niż milion.

## Wyjście

Jedyny wiersz wyjścia zawiera słowo „TAK” – jeśli z elementów ciągu można zbudować palindrom, lub „NIE” – jeśli nie jest to możliwe.

## Przykład

<p>Wejście</p> <p>5</p> <p>15 15 81 18 81</p> <p>Wyjście</p> <p>TAK</p> <p>Wyjaśnienie: Przykładowy możliwy do uzyskania palindrom: 81 15 18 15 81</p>	<p>Wejście</p> <p>6</p> <p>15 15 15 15 18 81</p> <p>Wyjście</p> <p>NIE</p>
--	--

# Odciski palca

---

Dostępna pamięć: 32MB

Pan Integer pracuje nad programem do odczytywania odcisków palców. Najwięcej uwagi poświęca modułowi odpowiedzialnemu za sprawdzanie, czy pewien kwadratowy czteroelementowy fragment odnalezionego obrazu (zdjęty z przedmiotu odcisk) może być jakąś częścią dużego obrazu (pełnego wzoru). Zastanawia się przy tym, ile razy fragment pojawił się w jakiegokolwiek formie na dużym obrazie. Pomóż mu to obliczyć!

## Wejście

W pierwszej i drugiej linii wejścia znajdują się po dwie dwucyfrowe liczby pierwsze – odnaleziony fragment obrazu. W drugiej linii wejścia znajdują się dwie liczby całkowite  $h$  oraz  $w$  ( $1 \leq h, w \leq 1000$ ), odpowiednio wysokość i szerokość dużego obrazu.

W kolejnych  $h$  liniach znajduje się po  $w$  rozdzielonych spacją dwucyfrowych liczb pierwszych.

## Wyjście

Liczba wystąpień dowolnej permutacji fragmentu obrazu we wzorze.

## Przykład

<b>Wejście</b> 11 13 17 19 2 3 23 11 13 29 17 19 <b>Wyjście</b> 1	<b>Wejście</b> 11 13 17 19 3 4 23 11 13 17 29 17 19 11 11 17 17 13 <b>Wyjście</b> 3	<b>Wejście</b> 11 13 17 19 1 4 11 13 17 19 <b>Wyjście</b> 0
--	---	---

# Brakująca cyfra

---

Dostępna pamięć: 64MB

Mamusia ciągle powtarzała Jankowi: 'Odrabiając pracę domową z matematyki nie pij nad zeszytem mleka!'. Janek oczywiście nie słuchał mamy i mleko nad zeszytem pił. Aż do czasu, kiedy mleko rozlało się na zadanie domowe! Całe szczęście, że rozmyła się tylko jedna cyfra w całej liczbie! Janek pamięta, że każda liczba na kartce była podzielna przez 9. Twoim zadaniem jest odnalezienie brakującej cyfry. Jeśli warunek spełnia kilka cyfr, pomóż odnaleźć Jankowi najmniejszą z nich.

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę  $k$  z pracy domowej ( $1 \leq k \leq 10^{1000}$ ). Zagubiona cyfra zastąpiona jest znakiem 'x'. Możesz przyjąć, że dla 40% testów zachodzi warunek  $k \leq 10^{18}$ .

Wyjście

Na wyjściu wypisz brakującą cyfrę.

Przykład

Wejście 23X54	Wyjście 4
------------------	--------------

# Ciąg monotoniczny

---

Dostępna pamięć: 32MB

Ciąg liczbowy nazywamy monotonicznym jeżeli jest rosnący, albo malejący, albo stały.

W naszym zadaniu dany jest ciąg liter. Jakie litery należy w nim zmienić, aby nasz ciąg był stały, zastępując przy tym jak najmniej znaków? Jeżeli jest kilka takich znaków, które można pozostawić, nie zmieniaj alfabetycznie największej z nich. Możesz założyć, że zawsze będzie co najmniej jedna litera do zmiany.

Wejście

W pierwszej linii wejścia znajduje się ciąg co najmniej dwóch liter (wyłącznie małe litery alfabetu łacińskiego) nie dłuższy niż  $10^6$  elementów.

Wyjście

Na wyjściu w pierwszym wierszu wypisz alfabetycznie wszystkie litery do zmiany.

Przykład

Wejście kajak	Wyjście aaj
------------------	----------------

Wyjaśnienie: Należy pozostawić litery 'k'.

## System 36

---

Dostępna pamięć: 32MB

Bitek sprawdza przydatność obliczeń w różnych systemach obliczeń. Stwierdził, że korzystając z cyfr i dużych liter alfabetu łacińskiego może korzystać nawet z systemu trzydziestoszóstkowego! Próbuje teraz napisać program, który szybko przeliczałby wartości pomiędzy dowolnymi systemami liczbowymi.

Wejście

Pierwszy wiersz zawiera trzy liczby:  $X$ ,  $Y$  i  $Z$ , gdzie  $X$  jest liczbą zapisaną w systemie o podstawie  $Y$  (wartość  $X$  nie przekracza  $10^{15}$  w systemie dziesiętnym).  $Z$  jest podstawą systemu, na który należy zamienić liczbę  $X$  ( $1 < Y, Z < 37$ ).

Wyjście

Jedna liczba całkowita - zapis  $X$  w systemie o podstawie  $Z$ .

Przykład

Wejście 37826876 10 36	Wyjście MIREK
---------------------------	------------------

# James i potęgi dwójki

Dostępna pamięć: 64MB

To był już prawie koniec misji. James Blond rozejrzał się uważnie po pokoju. Pomiedzy porozrzucanymi kartkami znajdowała się TA JEDNA, poszukiwana przez wszystkich. „Jak mogli jej nie zauważyć?”- zapytał sam siebie. Schylił się i wszystko zrozumiał. Na przedartej stronie widać było krótkie ciągi liczb. Ich końcowe cyfry musiały znajdować się na brakującej większej części. W panice zaczął przegarniać papiery. „Nie ma! Nie ma!” – krzyczał do siebie bezgłośnie. Usiadł bezsilnie wpatrując się ciągi znaków. I wtedy... przypomniał sobie ostatnie słowa Profesora: „Tylko potęgi dwójki!”. A więc o to mu chodziło! Każda z liczb to potęga dwójki! Wystarczy znaleźć najmniejszą potęgę dwójki, której początkowe cyfry widział na kartce! James wyjął swój smartphone, włączył aplikację kalkulatora w widoku programisty i... Nie mógł uwierzyć! Android się zawiesił! Czy coś mu jeszcze może pomóc?

## Zadanie

Napisz program, który dla zadanej dodatniej liczby całkowitej  $n$  wyznaczy najmniejszy wykładnik  $X$  taki, że pierwsze cyfry liczby  $2^X$  zgadzają się z zadaną liczbą. Pamiętaj, że oderwano większą część kartki, więc na pewno brakuje ponad połowy cyfr.

## Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita bez znaku  $n$  ( $1 \leq n < 10^{50}$ ).

## Wyjście

Dla podanej  $n$  wyznacz wykładnik  $X$  taki, że pierwsze cyfry liczby  $2^X$  zgadzają się z zadaną liczbą. Możesz założyć, że wynik zawsze będzie istniał i  $2^X < 10^{100}$ . Możesz założyć, że dla 40% testów zachodzi warunek  $2^X < 10^{18}$ .

## Przykład

Wejście 1	Wejście 5	Wejście 1 6
Wyjście 7	Wyjście 9	Wyjście 1 4

# Suma ułamków

---

Limit pamięci: 64MB

Napisz program, który czyta dwa ułamki zwykłe, a następnie wypisuje ułamek będący ich sumą.

**Wejście**

Dwa ułamki zapisane w postaci czterech liczb oddzielonych pojedynczym odstępem (odpowiednio licznik i mianownik pierwszej oraz licznik i mianownik drugiej liczby). Liczniki i mianowniki są liczbami naturalnymi nieprzekraczającymi miliarda.

**Wyjście**

Wypisywany ułamek (suma) w postaci licznika i mianownika przedzielonego znakiem /. Ułamek powinien być zapisany za pomocą liczb względnie pierwszych.

**Przykład**

<b>Wejście</b> 2 3 5 7	<b>Wejście</b> 1 4 7 4
<b>Wyjście</b> 29/21	<b>Wyjście</b> 2/1

# Liczby super-B-pierwsze

---

Limit pamięci: 64MB. Źródło: cke.gov.pl

Liczba „super-B-pierwsza” to taka liczba naturalna, która spełnia następujące warunki:

- jest liczbą pierwszą,
- suma cyfr tej liczby jest liczbą pierwszą,
- suma cyfr w jej zapisie binarnym jest liczbą pierwszą.

Oblicz, ile jest liczb „super-B-pierwszych” w przedziale  $[a, b]$ .

Wejście

Pierwszy wiersz danych zawiera dwie liczby całkowite  $a$  i  $b$  ( $2 \leq a < b \leq 1\,000\,000$ ).

Wyjście

Program powinien wypisać ilość liczb „super-B-pierwszych” w zadanym przedziale.

Przykład

Wejście 2 100	Wejście 8
------------------	--------------



# Trzy liczby rosnąco

---

Limit pamięci: 64MB.

Napisz program, który czyta trzy liczby całkowite, a następnie wypisuje je w kolejności niemalejącej.

Wejście

Dane wejściowe zawierają trzy liczby oddzielone pojedynczym odstępem. Dane są dodatnimi liczbami całkowitymi nieprzekraczającymi miliona.

Wyjście

Trzy liczby.

Przykład

Wejście 7 5 3	Wejście 3 5 7
------------------	------------------

# Struś pędziwiatr

---

Limit pamięci: 64MB

Struś Pędziwiatr porusza się z niesamowitą prędkością. Jest w stanie całe miasto przemierzyć w mniej niż jedną sekundę. Na przeszkodzie stoją mu jednak sygnalizatory świetlne. Nie są ze sobą zsynchronizowane, przez co Struś wciąż rusza i wciąż się zatrzymuje. Każde skrzyżowanie w mieście działa w ten sam sposób: przez  $x_i$  sekund na  $i$ -tym skrzyżowaniu zapalone jest czerwone, a następnie przez jedną sekundę zielone światło. Struś zauważył, że zdarzają się takie momenty, kiedy wszystkie światła w mieście świecą na zielono. Ile sekund w najgorszym przypadku musi czekać na taki moment Struś?

## Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita  $n$  ( $1 < n < 10$ ) – liczba skrzyżowań w mieście. W kolejnych  $n$  liniach znajduje się po jednej liczbie całkowitej  $x_i$  – czas trwania zmiany czerwonych świateł na  $i$ -tym skrzyżowaniu ( $1 < x_i < 50$ ). Możesz założyć, że każde ze skrzyżowań działa z inną częstotliwością zmian świateł.

## Wyjście

Jedna liczba całkowita – najmniejszy czas, jaki upłynie pomiędzy dwoma momentami, kiedy zielone światła zapalą się jednocześnie na wszystkich skrzyżowaniach.

## Przykład

Wejście	Wejście
2	11
3	
5	

# Liczby parzystocyfrowe

---

Limit pamięci: 256MB. Źródło: OIJ (szkopuł.edu.pl)

Dodatnią liczbę całkowitą nazywamy parzystocyfrową, jeśli wszystkie jej cyfry są parzyste. Na przykład: liczby 6, 42, 2020 są parzystocyfrowe, zaś 7, 34, 2019 lub 13 579 nie są. Gdyby wszystkie liczby parzystocyfrowe ustawić w kolejności rosnącej, która liczba byłaby N-ta w tym porządku?

Napisz program, który: wczyta liczbę naturalną N, wyznaczy N-tą liczbę parzystocyfrową i wypisze wynik na standardowe wyjście.

## Wejście

W pierwszym (jedynym) wierszu wejścia znajduje się jedna liczba naturalna N. Liczba będzie równa co najmniej 1 i co najwyżej  $10^{18}$ .

## Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna liczba naturalna – N-ta liczba parzystocyfrowa.

## Przykład

Wejście 12	Wejście 7921
Wejście 44	Wejście 446282

# Drzewo Sterna-Brocota

Limit pamięci: 128MB

Drzewo Sterna-Brocota to drzewo binarne zawierające wszystkie dodatnie ułamki nieskracalne. Własności (według Wikipedii):

- W drzewie występują wszystkie dodatnie liczby wymierne zapisane jako ułamki nieskracalne.
- Jeśli liczby  $a$  oraz  $b$  są względnie pierwsze, to ułamek  $\frac{a}{b}$  występuje w drzewie dokładnie raz.

Zaczynamy od  $\frac{0}{1}$  - symbolizującego zero i  $\frac{1}{0}$  symbolizującego nieskończoność. Następnie na kolejnych piętrach drzewa wpisujemy „pomiędzy” wartości  $\frac{a}{b}$  oraz  $\frac{c}{d}$  wartość  $\frac{a+c}{b+d}$ .

Zatem w pierwszym kroku mamy:

$$\frac{0}{1} \quad \frac{1}{1} \quad \frac{1}{0}$$

W drugim kroku:

$$\frac{0}{1} \quad \frac{1}{2} \quad \frac{1}{1} \quad \frac{2}{1} \quad \frac{1}{0}$$

Zaś w trzecim:

$$\frac{0}{1} \quad \frac{1}{3} \quad \frac{1}{2} \quad \frac{2}{3} \quad \frac{1}{1} \quad \frac{3}{2} \quad \frac{2}{1} \quad \frac{3}{1} \quad \frac{1}{0}$$

Napisz program, który czyta liczbę naturalną  $n$  i wypisuje sekwencję ułamków odpowiadającą temu numerowi iteracji.

## Wejście

Jedyny wiersz danych zawiera liczbę całkowitą  $n$  ( $0 \leq n \leq 20$ ) – numer iteracji.

## Wyjście

Program powinien wypisać wiersz tekstu zawierający sekwencję ułamków (zapisanych z użyciem znaku „/”) oddzielonych pojedynczymi odstępami.

## Przykład

Wejście

4

Wyjście

0/1 1/4 1/3 2/5 1/2 3/5 2/3 3/4 1/1 4/3 3/2 5/3 2/1 5/2 3/1 4/1 1/0

## Cyfry 2356

---

Dostępna pamięć: 32MB

Pan Integer znalazł w swoim pokoju pudełko z cyframi. W pudełku jest  $k_2$  cyfr 2,  $k_3$  cyfr 3,  $k_5$  cyfr 5 i  $k_6$  cyfr 6.

Ulubionymi liczbami całkowitymi pana Integera są 32 i 256. Postanowił poskładać te liczby z cyfr znajdujących się w pudełku. Każda cyfra może być użyta nie więcej niż jeden raz, tzn. wszystkie liczby powinny zawierać nie więcej niż  $k_2$  cyfr 2,  $k_3$  cyfr 3 i tak dalej. Chce przy tym, aby suma uzyskanych liczb była jak największa. Cyfry nieużywane nie są doliczane do sumy. Pomóż mu rozwiązać to zadanie!

### Wejście

Jedyna linia wejścia zawiera cztery liczby całkowite  $k_2, k_3, k_5$  i  $k_6$  - odpowiednio liczbę cyfr 2, 3, 5 i 6 ( $0 \leq k_2, k_3, k_5, k_6 \leq 5 \cdot 10^6$ ).

### Wyjście

Wypisz jedną liczbę całkowitą - maksymalną możliwą sumę ulubionych liczb całkowitych pana Integera, które można uzyskać za pomocą cyfr z pudełka.

### Przykład

Wejście 5 1 3 4 Wejście 800	Wejście 1 1 1 1 Wejście 256
--------------------------------------	--------------------------------------

Wyjaśnienie do przykładu pierwszego: Mamy pięć cyfr 2, jedną 3, trzy 5 i cztery 6. Pan Integer może złożyć 3 liczby 256 i jedną 32, co daje  $3 \cdot 256 + 32 = 800$ .

## Podatki

---

Dostępna pamięć: 64MB

Pan Longint zamieszkał w Bitolandii. Bardzo lubi tamtejsze widoki. Bitolandia to kraj o bardzo dziwnym systemie podatkowym. Kwota podatku, jaką musi zapłacić pan Longint od umowy opiewającej na  $n$  bitów (miejscowa waluta), jest równa maksymalnemu dzielnikowi właściwemu  $n$ . Na przykład dla trzech umów odpowiednio na 25, 6 i 2 bity należy zapłacić 9 bitów (5 bitów dla umowy na 25 bitów, 3 dla 6 i 1 dla 2 bitów).

Pan Longint dostrzegł jednak pewną lukę w przepisach. Otóż może on podzielić umowę na kilka różnych (mniejszych) umów i (być może) zapłacić mniejszy podatek opodatkowując każdą umowę oddzielnie! Czy mu się to opłaca?

Oblicz, jaki łączny (możliwie najmniejszy) podatek dla podanej kwoty zapłaci pan Longint!

Wejście

Pierwszy i jedyny wiersz wejścia zawiera jedną liczbę  $n$  ( $2 \leq n \leq 10^{10}$ ) – kwota, na jaką opiewa umowa.

Wyjście

Wypisz jedną liczbę całkowitą – najniższy podatek, jaki może zapłacić Longint.

Przykład

Wejście	Wejście
4	27
Wejście	Wejście
2	3

# Pogotowie lotnicze

---

Dostępna pamięć: 64MB

Bajtazar, król Bitolandii, chce podwyższyć poziom bezpieczeństwa w państwie. Postanowił powołać do życia lotnicze pogotowie ratunkowe. Ma nadzieję, że w ten sposób lekarze będą mogli śmigłowcami szybko dotrzeć do każdego mieszkańca. To na pewno podniesie poczucie bezpieczeństwa ludności.

Bitolandia to ciekawy kraj. Całe państwo przecina jedna prosta droga, a wszystkie miasta położone są wzdłuż tej drogi. Król Bajtazar chciałby umieścić bazę lotniczego pogotowia w takim mieście, żeby w razie nagłej potrzeby można było śmigłowcem jak najszybciej dolecieć w dowolne miejsce w kraju.

Pomóż mu i wskaż wszystkie miasta, które się do tego nadają!

## Wejście

W pierwszym wierszu wejścia znajduje się liczba miast w Bitolandii  $n$  ( $1 \leq n \leq 10^6$ ). W drugiej linii wejścia znajduje się  $n-1$  liczb całkowitych  $x_i$  ( $1 \leq x_i \leq 10^6$ ), gdzie  $x_i$  oznacza odległość między miastami o numerze  $i$  oraz  $i+1$ .

## Wyjście

W jednej linii wypisz w kolejności rosnącej wszystkie miasta, w których można umieścić bazę lotniczego pogotowia ratunkowego.

## Przykład

Wejście 7 2 3 2 2 4 1	Wejście 4
-----------------------------	--------------

# Kolorowe żarówki

---

Dostępna pamięć: 64MB

W Bitocji mieszkańcy lubią ozdabiać domy kolorowymi żaróweczkami. Tworzą z nich różne wzory, ale zawsze dbają o to, by żarówki znajdowały się w jednej linii obok siebie. Czasem trzeba dokonywać napraw i wówczas ważne jest, by żarówki w tym samym kolorze znajdowały się blisko siebie. Każdy mieszkaniec chce więc wiedzieć, w jakiej odległości żarówki tego samego koloru są najbliższej oraz najdalej siebie.

## Wejście

W pierwszym wierszu wejścia znajduje się liczba  $n$  ( $1 \leq n \leq 10^6$ ). W drugiej linii znajduje się  $n$  liczb całkowitych  $k_i$  – kolory żarówek ( $1 \leq k_i \leq n$ ).

## Wyjście

Twój program powinien zapisać dwie liczby całkowite oznaczające odpowiednio najmniejszą odległość pomiędzy dwoma żaróweczkami dowolnego ale tego samego koloru oraz największą możliwą odległość pomiędzy dwoma żaróweczkami dowolnego, ale tego samego koloru. Jeżeli nie jest możliwe wyznaczenie któregoś z odległości, wypisz 0.

## Przykład

Wejście 7 1 2 3 3 1 1 3	Wejście 1 5
-------------------------------	----------------

Wyjaśnienie: najbliższe żarówki w kolorze 3 znajdują się w odległości 1 (1 2 3 3 1 1 3), najdalsze żarówki w kolorze 1 znajdują się w odległości 5 (1 2 3 3 1 1 3).



# Roztargniony profesor

---

Dostępna pamięć: 64MB

Pewien profesor, aby dostać się do swego gabinetu, musi pokonać  $N$  schodów. Profesor pokonuje zazwyczaj jeden lub dwa schody na raz. Oblicz, na ile różnych sposobów profesor może pokonać schody. Podaj te sposoby.

Wejście

Jedna liczba całkowita  $N$  ( $1 \leq N \leq 25$ ) - liczba schodów.

Wyjście

Wyjście zawiera w pierwszym wierszu liczbę możliwych kombinacji, w kolejnych wierszach wszystkie możliwe kombinacje w porządku leksykograficznym.

Przykład

Wejście	Wyjście
3	3 1 1 1 1 2 2 1

# Mysz w labiryncie 1

---

Dostępna pamięć: 32MB

Do labiryntu trafiła mysz. Pomóż jej odnaleźć wyjście!

## Wejście

W pierwszej linii wejścia znajdują się dwie liczby całkowite  $n$  i  $m$  ( $2 \leq n, m \leq 1000$ ) oznaczające rozmiar labiryntu:  $n$  kolumn i  $m$  wierszy. W kolejnych liniach znajdują się znaki oznaczające kolejno:

- - - możliwość przejścia (korytarz);
- x - brak przejścia (ściana);
- o - początkowe położenie myszy;
- w - wyjście (końcowe położenie myszy).

## Wyjście

Pierwszy i jedyny wiersz wyjścia powinien wypisać „**TAK**” – jeśli istnieje wyjście z labiryntu, lub „**NIE**” – jeżeli warunek nie będzie spełniony.

## Przykład

<pre>Wejście 8 8 w--x-x-- xx-x-x-- ---x---- --xxxx-- --x--x-- -----x-o xxxx---- -----x-</pre>	<pre>Wyjście TAK</pre>
---	------------------------

## Problem 8 hetmanów

---

Dostępna pamięć: 64MB

Janek uczy się grać w szachy. Poznaje sposób poruszania się różnych figur. Najśmieszniejszy wydaje mu się styl skoków konika. Zastanawia się również, co byłoby, gdyby na szachownicy znalazło się kilka figur hetmanów. Analizuje, czy na planszy o wymiarach  $n \times n$  możliwe jest takie rozstawienie  $n$  tych figur, aby żaden z nich się nie szachował, a jeśli tak, to na ile sposobów można to zrobić?

Wejście

$n$  - rozmiar szachownicy, liczba naturalna mniejsza niż 15.

Wyjście

Liczba sposobów, na jakie można rozmieścić figury hetmanów na szachownicy tak, by się wzajemnie nie szachowały.

Przykład

Wejście 8	Wyjście 92
--------------	---------------

# Latarnie

---

Dostępna pamięć: 32MB

*Samotne długie noce i ostre noże w kieszeniach  
Wskazówki zegarka świecą w ciemnościach  
nie, nie, nie  
Kroki na schodach, w bezruchu zamierasz  
Tu czai się strach, tak ciemno tu teraz  
nie, nie, nie  
Właśnie tak, tak wygląda moje miasto nocą,  
tak wygląda nocą świat  
nie, nie, nie  
tak, tak wygląda moje miasto nocą,  
tak wygląda nocą świat*

De Mono: Moje miasto nocą

Nie każdy lubi chodzić nocą ulicami, na których latarnie dają zbyt mało światła. Ciemności boi się również pan Integer. Wzdłuż ulicy, na której mieszka, postawiono  $k$  latarni. Pan Integer uważa, że jest ich stanowczo za mało. Albo za słabo świecą. Ponieważ zaplanowano właśnie wymianę wszystkich żarówek na nowe (energooszczędne), pan Integer postanowił zwrócić się do miejscowego zarządu dróg z nietypową prośbą. Chciałby, aby każda latarnia świeciła światłem z odpowiednią jasnością i co najmniej o promieniu  $r$  tak, aby cała ulica była oświetlona. Nie chce przy tym narażać miasta na dodatkowe koszty, dlatego chce wyznaczyć  $r$  tak, by było jak najmniejsze. Pomożesz mu?

## Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite  $d$  oraz  $k$  – odpowiednio długość ulicy, na której mieszka pan Integer oraz liczba latarni ( $1 \leq k \leq d \leq 10^6$ ). W kolejnej linii znajduje się  $k$  liczb całkowitych  $x_i$  – odległości  $i$ -tej latarni od początku ulicy ( $0 \leq x_i \leq d$ ). Latarnie mogą stać w dowolnym miejscu ulicy. W jednym miejscu znajduje się tylko jedna latarnia.

## Wyjście

Minimalny promień światła  $r$ , dla którego każdy punkt ulicy zostanie rozświetlony. Wynik wypisz z dokładnością do jednego miejsca po przecinku.

## Przykład

Wejście 10 3 1 5 9	Wyjście 4.0
--------------------------	----------------

# Najczęściej występujący

---

Dostępna pamięć: 32MB

Wyznacz najpopularniejszy element ciągu (to znaczy taki, który występuje w nim największą liczbę razy). W przypadku, gdy w ciągu jest więcej niż jeden najpopularniejszy element, podaj największy z nich.

Wejście

W jednej linii znajduje się ciąg liczb całkowitych zakończony liczbą 0. Możesz założyć, że liczb jest co najmniej dwie i nie więcej niż 1000000, zaś ich wartość mieści się w przedziale  $[-1018, 1018]$ .

Wyjście

Najpopularniejszy element ciągu.

Przykład

Wejście 1 2 -5 7 1 3 3 7 3 1 0	Wyjście 3
-----------------------------------	--------------

# Ciąg Farey'a

---

Dostępna pamięć: 32MB

Ciąg Farey'a dla liczby naturalnej  $N$  ( $N > 1$ ) to ciąg ułamków, których licznik i mianownik są liczbami naturalnymi nieprzekraczającymi liczby  $N$  oraz licznik nie jest większy od mianownika. Ułamki powinny być skrócone, posortowane rosnąco i nie mogą się powtarzać. Oto przykład ciągu Farey'a dla  $N = 4$ :

$$\begin{array}{cccccc} 0 & 1 & 1 & 1 & 2 & 3 & 1 \\ \hline 1 & 4 & 3 & 2 & 3 & 4 & 1 \end{array}$$

Napisz program, który czyta liczbę  $N$  i wypisuje odpowiadający jej ciąg Farey'a.

Wejście

Pierwszy i jedyny wiersz danych zawiera jedną liczbę naturalną  $N$  ( $3 \leq N \leq 1000$ ).

Wyjście

Program powinien wypisać w jednym wierszu ciąg ułamków zapisanych przy użyciu znaku „/” (np. „1/2”) oddzielonych pojedynczymi odstępami, stanowiącymi ciąg Farey'a dla liczby  $N$ .

Przykład

Wejście	Wyjście
4	0/1 1/4 1/3 1/2 2/3 3/4 1/1

# Ojciec Wirgiliusz

---

Limit pamięci: 128MB

*Ojciec Wirgiliusz,  
uczył dzieci swoje,  
a miał ich wszystkich  
prawdziwe roje,  
takie malutkie,  
i takie wielkie,  
takie grubiotkie,  
i takie cienkie.*

*Piosenka tradycyjna*

Wszystkie dzieci Wirgiliusza postanowiły zamieszkać w Bajtomiu w pięknych domkach na nowo wybudowanej ulicy Bajtockiej. Razem z nimi zamieszka również Wirgiliusz. Ponieważ ojciec zamierza często odwiedzać wszystkie swoje dzieci, chciałby znaleźć taki dom jednego z nich, aby mieć możliwie blisko do nich wszystkich.

Ojciec Wirgiliusz chce zminimalizować sumaryczną odległość do wszystkich swoich dzieci i poprosił Ciebie, abys napisał stosowny program.

## Wejście

Pierwszy wiersz zawiera liczbę całkowitą  $r$  ( $1 \leq r \leq 10^6$ ) oznaczającą liczbę dzieci Wirgiliusza. Drugi wiersz zawiera  $r$  całkowitych numerów domów  $s_1, s_2, \dots, s_i, \dots, s_r$ , w których one mieszkają ( $1 \leq s_i \leq 1\,000\,000\,000$ ). Zauważ, że różne dzieci mogą mieszkać w tych samych domach.

## Wyjście

W pierwszym i jedynym wierszu wyjścia Twój program musi wypisać minimalną sumę odległości optymalnie położonego domu Wirgiliusza od wszystkich dzieci. Odległość między dwoma domami o numerach  $s_i$  i  $s_j$  wynosi  $d_{ij} = |s_i - s_j|$ .

## Przykład

Wejście	Wyjście
3 2 4 6	4

# Układanie kart

---

Dostępna pamięć: 64MB

Mały Bitek dostał od rodziców talię kart. Ze zdziwieniem jednak zauważył, że karty zamiast tradycyjnych oznaczeń typu *król pik* są oznaczone kolejnymi liczbami całkowitymi. Szybko jednak bardzo mu się to spodobało.

Bitek grając w karty zawsze ustawia je od najmniejszej do największej. Stosuje przy tym popularną metodę:

- zakłada, że pierwsza karta jest już na swoim miejscu,
- każdą kolejną kartę przesuwa w lewo tak długo, aż znajdzie się ona na właściwej pozycji.

Bitek zauważył, że w trakcie porządkowania  $i$ -ta karta miją zawsze  $k_i$  innych kart. Bardzo go to zaintrygowało i teraz chciałby wiedzieć, ile łącznie takich „minięć” zostanie wykonanych dla całej potasowanej talii.

Wejście

W pierwszym wierszu znajduje się jedna liczba całkowita  $n$  – liczba kart ( $1 \leq n \leq 10^6$ ). W kolejnej linii znajduje się  $n$  różnych liczb całkowitych  $x_i$  – wartości kolejnych kart ( $1 \leq x_i \leq n$ ).

Wyjście

Jedna liczba całkowita oznaczająca łączną liczbę miniętych w trakcie układania kart.

Przykład

Wejście	Wyjście
4	5
4 2 3 1	



# Równoważne teksty

---

Dostępna pamięć: 256MB

Dwa teksty  $a$  oraz  $b$  nazwiemy równoważnymi, jeżeli spełniają jeden z dwóch warunków:

- są identyczne,
- jeżeli tekst  $a$  podzielimy na dwie równej długości części  $a_1$  i  $a_2$ , a tekst  $b$  podzielimy podobnie na teksty  $b_1$  i  $b_2$ , to jeden z warunków będzie poprawny:
  - $a_1$  i  $b_1$  oraz  $a_2$  i  $b_2$  będą równoważne  
*lub*
  - $a_1$  i  $b_2$  oraz  $a_2$  i  $b_1$  będą równoważne.

Sprawdź, czy podane dwa teksty są równoważne według powyższej definicji!

## Wejście

W dwóch wierszach wejścia znajduje się po jednym tekście. Teksty mają długość od 1 do  $2 \cdot 10^5$  znaków i składają się wyłącznie z małych liter alfabetu łacińskiego. Długości tekstów są równe.

## Wyjście

Wypisz jedno słowo: TAK lub NIE, odpowiedź na pytanie, czy podane teksty są równoważne.

## Przykład

Wejście abcd cdba	Wyjście TAK
Wejście abcd acbd	Wyjście NIE

### Objaśnienie przykładu 1

Tekst  $abcd$  rozkładamy na  $ab$  i  $cd$ . Tekst  $cdba$  rozkładamy na  $cd$  i  $ba$ . W tym wypadku  $cd$  jest równoważne  $cd$ . Postępując tą samą metodą stwierdzimy, że  $ab$  i  $ba$  są równoważne.

# Znaczki

---

Dostępna pamięć: 32MB

Bitek i Bajtek zbierają znaczki. Bitek planuje wziąć udział w wystawie połączonej z konkursem na największy zbiór znaczków. Aby zwiększyć swoje szanse na odniesienie spektakularnego sukcesu, postanowił pożyczyć od Bajtka te znaczki, których sam nie posiada. Chciałby teraz wiedzieć, o ile znaczków zwiększy się jego kolekcja.

## Wejście

W pierwszej linii wejścia znajdują się dwie liczby naturalne  $n$  (mniejsza niż 100 000) i  $m$  (mniejsza niż 10 000 000), oznaczające odpowiednio liczbę znaczków Bitka oraz znaczków Bajtka. W drugiej linii wejścia znajduje się  $n$  różnych liczb naturalnych mniejszych niż miliard – typy znaczków Bitka. W trzeciej linii wejścia znajduje się  $m$  różnych liczb naturalnych mniejszych niż miliard – typy znaczków Bajtka. Możesz założyć, że w Bitek i Bajtek podali każdy posiadany znaczek tylko raz.

## Wyjście

W jednym wierszu wyjścia znajduje się jedna liczba całkowita: liczba znaczków w kolekcji Bajtka, których nie posiada Bitek.

## Przykład

Wejście	Wyjście
6 4 1 2 3 4 5 6 2 4 5 7	1

# Tunele

---

Limit pamięci: 32MB

Janek zarządza dużą firmą dostawczą. Właśnie zamierza opracować nowy plan dostaw do swoich kontrahentów. Do wszystkich można dojechać nowo wybudowaną autostradą. Niestety samochody na autostradzie muszą pokonywać również tunele. Każdy z samochodów oraz każdy z tuneli ma określoną wysokość. Aby ciężarówka Janka przejechała pod tunelem, musi być niższa niż wysokość tunelu. Janek zna wysokości ciężarówek i tuneli i zastanawia się, do którego miejsca autostrady dojedzie każde z jego aut.

## Wejście

W pierwszej linii wejścia znajduje się dwie liczby całkowite  $a$  i  $b$  ( $1 \leq a, b \leq 500\,000$ ), odpowiednio ilość tuneli i samochodów. W drugiej linii znajduje się  $a$  oddzielonych spacjami liczb całkowitych  $t_1, t_2, \dots, t_a$  ( $1 \leq t_i \leq 1\,000\,000\,000$ ), gdzie  $t_i$  oznacza wysokość  $i$ -tego tunelu. W trzeciej linii znajduje się  $b$  oddzielonych spacjami liczb całkowitych  $s_1, s_2, \dots, s_b$  ( $1 \leq s_i \leq 1\,000\,000\,000$ ), gdzie  $s_i$  oznacza wysokość  $i$ -tego samochodu.

## Wyjście

Pierwszy i jedyny wiersz wyjścia zawiera  $b$  liczb całkowitych  $d_1, d_2, \dots, d_b$ , gdzie  $d_i$  oznacza numer ostatniego tunelu, przez który może przejechać  $i$ -ty samochód lub 0, jeśli nie może przejechać przez żaden z tuneli.

## Przykład

Wejście	Wyjście
5 3	5 1 0
8 6 5 7 9	
4 7 9	

## Bierki (OIG)

---

Zadanie zapożyczone: I OIG. Limit pamięci: 32MB

Jaś lubi budować trójkąty z bierek. W tym celu trzyma je w worku, z którego wybiera trzy bierki na chybił-trafił. Bierki mogą mieć różne długości i nie zawsze Jaś może zbudować trójkąt, a wtedy wpada w histerię. Mama Jasia ma dość histerycznych napadów synka i dlatego poprosiła Ciebie o pomoc. Należy odrzucić niektóre bierki w taki sposób, aby z pozostałych zawsze dało się ułożyć trójkąt, jednocześnie zostawiając jak najwięcej bierek w worku.

Zadanie

Opracuj program, który:

- wczyta ze standardowego wejścia liczbę bierek w worku oraz ich długości,
- obliczy największą liczbę bierek, którą można pozostawić w worku, tak żeby z każdych trzech z nich można było utworzyć trójkąt,
- wypisze wynik na standardowe wyjście.

Wejście

W pierwszym wierszu zapisano liczbę  $n$  ( $5 \leq n \leq 30\,000$ ), oznaczającą liczbę bierek w worku. W każdym z  $n$  następnym wierszy zapisano długość jednej bierki. Długość bierki jest liczbą całkowitą z przedziału  $[1, 500]$ .

Wyjście

W pierwszym wierszu wypisz liczbę bierek, które powinny zostać w worku.

Przykład

Dla danych wejściowych:	poprawną odpowiedzią jest:
10 7 1 2 8 10 6 1 7 9 9	7

# Zawody wędkarskie

---

Dostępna pamięć: 32MB

Jasio organizuje zawody wędkarskie. Dokonał właśnie przeglądu wszystkich  $n$  stanowisk, jakimi dysponuje. Stanowiska umiejscowione są w linii prostej nad jeziorem w odległościach  $x_1, x_2, \dots, x_n$  od początku jeziora. W czasie konkursu zawodnicy nie mogą sobie nawzajem przeszkadzać. Dlatego Jaś chciałby teraz tak rozmieścić w wędkarzy, aby najmniejsza odległość pomiędzy dwoma najbliższymi z nich była jak największa. Pomóż znaleźć Jasiowi taki rozkład wędkarzy.

## Wejście

W pierwszej linii wejścia znajdują się dwie liczby całkowite:  $n$  oraz  $w$  ( $2 \leq w \leq n \leq 10^6$ ). W kolejnej linii znajduje się  $n$  liczb całkowitych  $x_i$  ( $0 \leq x_i \leq 10^9, x_i - 1 < x_i$ ). Każda z wartości oznacza odpowiednio odległość  $i$ -tego stanowiska od początku jeziora.

## Wyjście

Jedna liczba całkowita wyznaczająca najlepszą możliwą odległość pomiędzy dwoma najbliższymi stanowiskami.

## Przykład

Wejście 6 3 2 5 10 13 18 19	Wyjście 8
-----------------------------------	--------------

## Ocenianie

Podzadanie	Ograniczenia	Liczba punktów
1	$n \leq 10^4$	40
2	brak dodatkowych założeń	60

# Marchewki

---

Dostępna pamięć: 64MB

Bitomir zasadził na swoim polu marchewki. Każda marchewka jest oddalona od innej marchewki o jeden bitometr (jednostka długości w Bajtocji) w poziomie i w pionie. Bitomir chce zakupić specjalny zbierak do marchewek. Zbierak zamontowany jest na długim ramieniu i potrafi zebrać wszystkie marchewki w promieniu swojego działania. Bitomir zastanawia się teraz, jakiej długości ramię (w bitometrach) musi posiadać zbierak, by jednorazowo zebrał co najmniej  $m$  marchewek. Cena zbieraka zależy od długości ramienia. Bitomir chce zakupić jak najtańsze urządzenie. Pomóż mu wyznaczyć najlepszy zbierak!

## Wejście

W pierwszej linii wejścia znajduje się jedna liczba całkowita  $m$  ( $2 \leq m \leq 5 \cdot 10^{12}$ ), oznaczająca liczbę marchewek, którą chcemy jednorazowo zebrać.

## Wyjście

Na wyjściu powinna znaleźć się jedna liczba całkowita oznaczająca minimalną długość ramienia zbieraka.

## Przykład

Wejście 13	Wyjście 2
---------------	--------------

## Ocenianie

Podzadanie	Ograniczenia	Liczba punktów
1	$n \leq 10^5$	15
2	$n \leq 10^8$	15
3	brak dodatkowych założeń	60

# Pszczółki

---

Dostępna pamięć: 256MB

Pszczóły rozmnażają się w specyficzny sposób: z niezaplodnionych jaj rodzą się samce (trutnie), a z zapłodnionych – samice (pszczóły). Rodzina pszczoły jest więc nietypowa – ma ona co prawda i ojca, i matkę, ale już tylko jednego dziadka i dwie babki.

Janek dogłębnie bada ten temat i zastanawia się, ile męskich przodków żyło w rodzinie pszczoły w  $n$ -tym pokoleniu. Pomóż Jankowi dokonać obliczeń!

## Wejście

Pierwszy wiersz zawiera jedną liczbę całkowitą  $n$  ( $0 < n \leq 10^6$ ) – numer pokolenia pszczoły, dla którego szukamy liczby trutni.

## Wyjście

Jedna liczba całkowita – liczba trutni w  $n$ -tym pokoleniu. Ponieważ liczba trutni bardzo szybko rośnie, wystarczy, że wypiszesz wynik mod  $10^9+7$ .

## Przykład

Wejście 5	Wyjście 5
--------------	--------------

# iGruszka

---

Dostępna pamięć: 64MB

Firma *iGruszka* wyprodukowała nowy model smartfona. Telefon ma dotykowy ekran, całkiem niezłe parametry i śliczne logo w kształcie nadgryzionej gruszki. Zarząd firmy postanowił jak najwięcej zarobić na swoim sztandarowym produkcie. Przeprowadzono więc bardzo szczegółowe badania sprawdzające, jaką kwotę są gotowi zapłacić poszczególni klienci za aparat. Sprawdzone też, jaką liczbę telefonów mogą wyprodukować firmowe fabryki. Znając koszty produkcji urządzenia, wyniki badań klientów oraz liczbę możliwych do zamówienia smartfonów, oblicz najlepszą cenę urządzenia (tzn. taką, by firma *iGruszka* zarobiła jak najwięcej) oraz zysk firmy.

## Wejście

Pierwszy wiersz wejścia zawiera trzy liczby całkowite  $s$ ,  $k$  oraz  $n$  – odpowiednio liczba możliwych do wyprodukowania telefonów, koszt jednego telefonu oraz liczba potencjalnych klientów ( $0 \leq s, k, n \leq 10^6$ ). W kolejnych  $n$  liniach znajdują się maksymalne kwoty  $x_i$ , które  $i$ -ty klient jest gotowy zapłacić za telefon ( $0 \leq x_i \leq 10^6$ ).

## Wyjście

Na wyjściu w jednym wierszu wypisz jedną liczbę całkowitą – maksymalny możliwy do uzyskania zysk firmy.

## Przykład

Wejście	Wyjście
5 8 5	14
15	
11	
10	
12	
16	



# Spadek

---

Dostępna pamięć: 64MB

Stary Jan ma dwóch synów. Ponieważ chłopcy często się kłócą, postanowił zapobiec następnym niesnaskom i sam podzieli majątek między nich. Jan chce rozdzielić majątek na dwie części tak, aby różnica wartości pomiędzy nimi była jak najmniejsza. Jeśli nie da się podzielić po równo, starszy syn otrzyma więcej.

Wejście

W pierwszym wierszu standardowego wejścia zapisano liczbę naturalną  $p$  ( $1 \leq p \leq 150$ ) – liczbę składników majątku, a następnie ich wartości  $w_i$  ( $1 \leq w_i \leq 10\,000$ ,  $i = 1, 2, 3 \dots p$ ).

Wyjście

W jednym wierszu standardowego wyjścia zapisz kwoty wartości części majątku starszego i młodszego syna, rozdzielając je spacją.

Przykład

<b>Wejście</b> 4 2 2 2 5	<b>Wyjście</b> 6 5
--------------------------------	-----------------------

<b>Wejście</b> 5 10 5 10 5 5	<b>Wyjście</b> 20 15
------------------------------------	-------------------------

# Poszukiwania

Dostępna pamięć: 32MB

Książę Bratumił II Wstydlivy przez stosunek do białogłów swój przydomek otrzymał. Przeto jego rodziciel, król Rufin IV Rzadkowąsy, martwił się wielce. 40 lat Bratumiła dojrzałością późną na pewno zwać nie można, wszelako żona i dzieci dla syna chęcią przedłużenia rodu królewskiego nieodzownie Rufinowi wydawać się mogły. Toteż synowi nakazał, by królestwo całe przewędrował i za żonę jakową dorodną infantkę pojał. Syn wolę ojca spełnić zaprzysiął, aliści warunek także wysunął: spędzi w podróży na poszukiwaniach księżniczki o dwa dni mniej, niż monarchia ojca hrabstw w szerokości i w długości w sumie liczy, a we wszelkim odwiedzionym hrabstwie dokładnie jeden dzionek pobędzie. Rufin tą przymusowością strapił się bardzo, azaliż synowi przyrzeczenie dał. Wszelako sam mu szlak zdeklarował się wyznaczyć taki, ażeby Bratumił na jak największą liczbę księżniczek okiem rzucił.

Wtórą noc już Rufin oka zmrużyć nie może. Jak trasę wyznaczyć? Na szczęście informatyk z eskapady dalekiej akuratnie przybył i laptop najnowszej generacji z procesorem wielordzeniowym sprowadził. Tedy król pchnął umyślnego po nadwornego, by mu ową trasę ustalił.

## Zadanie

Mapa państwa to tablica kwadratowa o wymiarach  $n \times n$ . Każdy kwadrat określa jedno hrabstwo, w każdej komórce znajduje się liczba zamieszkujących go księżniczek.

Napisz program, który dla danej tablicy liczb  $n \times n$  oblicza ścieżkę o maksymalnej liczbie księżniczek prowadzącą od lewego górnego pola tablicy do prawego dolnego. Ścieżka składa się z sekwencji kroków. Liczba kroków wynosi dokładnie  $2 \cdot n - 2$ .

Maksymalną liczbą księżniczek ścieżki jest suma liczb w każdym z odwiedzionych pól tablicy.

Maksymalna ścieżka w przykładowej tablicy o wymiarze  $6 \times 6$  pokazana jest poniżej.

1	2	1	3	1	2
5	2	1	9	1	2
2	2	1	2	1	2
1	3	2	4	8	9
1	2	2	3	1	2
2	1	1	5	2	9

## Wejście

W pierwszym wierszu wejścia zapisana została jedna liczba całkowita  $n$ , oznaczającą liczbę wierszy oraz kolumn tablicy. Dalej następuje  $n \times n$  liczb naturalnych nie większych niż 1000,

w porządku wiersz-kolumna, tj. pierwszych n liczb tworzy pierwszy wiersz tablicy, następnych n liczb tworzy drugi wiersz itd. Liczby w wierszu są od siebie oddzielone przynajmniej jedną spacją. Liczba wierszy i kolumn jest z zakresu od 1 do 1000 włącznie.

## Wyjście

Pierwsza i jedyna liczba określa maksymalną liczbę księżniczek, jakie może poznać Bratumił.

## Przykład

Wejście 3 1 1 1 1 1 1 1 1 1	Wyjście 5
Wejście 6 1 2 1 3 1 2 5 2 1 9 1 2 2 2 1 2 1 2 1 3 2 4 8 9 1 2 2 3 1 2 2 1 1 5 2 9	Wyjście 52

# Stadion

---

Dostępna pamięć: 32 MB

Trwają przygotowania do następnych Mistrzostw Europy w piłce nożnej. W planie nie może zabraknąć stadionu w okolicach naszego pięknego miasta. Tym razem zbudujemy największy stadion na planie kwadratu! Potrzebne więc będzie znalezienie w pobliskiej puszczy kwadratowego obszaru o jak największym polu. W zadaniu tym przeszkadzają rosnące w puszczy drzewa. Dysponujemy mapą puszczy o wymiarach  $N \times M$  oraz rozmieszczeniem drzew.

## Wejście

W pierwszym wierszu standardowego wejścia zapisano dwie liczby całkowite dodatnie, oddzielone pojedynczym odstępem:  $n$  i  $m$  ( $1 \leq n, m \leq 1000$ , odpowiednio liczba kolumn i wierszy). W drugim wierszu zapisano liczbę drzew  $d$  ( $1 \leq d \leq 1\,000\,000$ ), których współrzędne podano w kolejnych  $d$  wierszach.

## Wyjście

W pierwszym wierszu standardowego wyjścia zapisz pole największego kwadratu, jaki można wyznaczyć pomiędzy drzewami w puszczy.

## Przykład

Wejście	Wejście	Wejście
6 7	10 17	15 20
1	2	3
2 3	5 6	3 6
	7 8	9 8
		12 15
Wyjście	Wyjście	Wyjście
16	81	121

# Zbiórka

---

Limit pamięci: 64MB

Pałac króla Bitolandii jest w ruinie, a kasa państwa świeci pustkami! Obywatele Bitolandii postanowili przeprowadzić zbiórkę pieniędzy na ratowanie cennego zabytku. W centrum stolicy, Bitogrodu, ustawiono wielką skrzynię, do której każdy może wrzucić datek. Zbiórka trwa już jakiś czas. Nikt jednak nie wie, jaka kwota mogła się dotychczas zbierać. Gdyby tylko społeczny komitet ratowania zabytku miał pewność, że zebrane środki są wystarczające, zakończono by kwestę i przystąpiono do prac. W Bitolandii używa się tylko monet, a każda ma określoną wagę i nominał. Czy znając wagę zebranych monet możliwe jest określenie minimalnej kwoty, jaką do tej pory zebrano?

## Wejście

Pierwszy wiersz zawiera jedną liczbę całkowitą  $c$  – ciężar zgromadzonych pieniędzy, nie więcej niż 10000. W drugiej linii znajduje się jedna liczba całkowita  $n$  – liczba różnych monet używanych w danej walucie ( $1 \leq n \leq 1000$ ). W kolejnych liniach znajdują się opisy monet zawierające po dwie liczby całkowite, nominał  $m$  oraz wagę  $w$  ( $1 \leq m \leq 50000$ ,  $1 \leq w \leq 10000$ ).

## Wyjście

Wypisz minimalną ilość pieniędzy, które można osiągnąć za pomocą monet z danej masy całkowitej. Jeśli waga nie może być osiągnięta, należy wypisać komunikat "NIEMOZLIWE".

## Przykład

Wejście: 100 2 1 10 20 5	Wyjście: 10	Wejście: 100 2 10 1 20 5	Wyjście: 400	Wejście: 5 2 2 2 4 4	Wyjście: NIEMOZLIWE
--------------------------------------	----------------	--------------------------------------	-----------------	----------------------------------	------------------------

# Mnóż, odejmuj, dodawaj

Dostępna pamięć: 32MB

Czy dodawanie, odejmowanie i mnożenie może sprawiać kłopoty? Oczywiście że nie! Dlatego mam dla ciebie proste zadanie: dla podanego w specjalnym zapisie wyrażenia arytmetycznego oblicz jego wartość! Specjalność tego zapisu polega na tym, że najpierw zawsze pojawiają się argumenty, a dopiero później działania M, O lub D (mnożenie, odejmowanie lub dodawanie). W wyrażeniu tym jednak nie musisz się martwić kolejnością działań arytmetycznych – po prostu wykonuj je zgodnie z wejściem!

## Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba  $N$  ( $1 \leq N \leq 10$ ) – liczba wyrażeń do rozpatrzenia. W kolejnych  $N$  liniach znajdują się krótkie wyrażenia arytmetyczne. Długość żadnego z nich nie przekracza 1000 znaków. Wyrażenia składają się wyłącznie z wartości  $x_i$  ( $0 \leq x_i \leq 10^{15}$ ) oraz działań opisanych odpowiednimi literami.

## Wyjście

W oddzielnych liniach należy wypisać  $N$  liczb całkowitych – wyniki obliczeń. Możesz założyć, że końcowy wynik nigdy nie przekroczy  $|10^{15}|$ .

## Przykład

Wejście	Wyjście
5	9
4 5 D	-10
10 20 O	11
2 3 M 5 D	17
2 3 5 M D	9
5 5 M 20 O 2 2 D D	

## Wyjaśnienie:

Kolejne wyrażenia to odpowiednio:  $4+5$ ,  $10-20$ ,  $2*3+5$ ,  $2+3*5$ ,  $5*5-20+2+2$

# Wodzirej

---

Dostępna pamięć: 32MB

Janek urządził imprezę. Tym razem postanowił wybrać jedną osobę, która będzie decydowała o przebiegu zabaw – wodzireja. Chciałby, aby osoba ta była lubiana przez jak największą grupę uczestników spotkania. Pomóż wskazać mu tę osobę oraz grono osób, które ją lubią.

## Wejście

Pierwszy wiersz danych zawiera dwie liczby całkowite  $n$  i  $m$ , odpowiednio liczbę osób biorących udział w przyjęciu oraz liczbę wzajemnych sympatii ( $2 \leq n \leq 10\,000$ ,  $2 \leq m \leq 1\,000\,000$ ). W kolejnych  $m$  liniach znajdują się po dwie liczby całkowite  $u_i$  i  $v_i$  ( $1 \leq u_i, v_i \leq 10\,000$ ) – informacje o parach osób, które się lubią wzajemnie.

## Wyjście

Program powinien wypisać w pierwszym wierszu jedną liczbę – najbardziej lubianą osobę na przyjęciu (wodzireja). Jeśli jest więcej takich osób – tę o najmniejszej liczbie porządkowej.

W drugiej linii program powinien wypisać w kolejności rosnącej wszystkie osoby, które lubią wodzireja.

## Przykład

Wejście	Wyjście
5 4	2
1 3	3 4 5
2 3	
4 2	
5 2	

# Gorące pudełko

Dostępna pamięć: 256MB

Czarująca pani Char, przedszkolanka w przedszkolu Słoneczny Integer, lubi urządzać dla swoich podopiecznych różne zabawy. Jedną z nowych zabaw, które pani Char zaproponowała dzieciom, jest zabawa w gorące pudełko. Polega ona na tym, że pani Char wręcza pierwszemu dziecku kolorowe pudełko, ono podaje je drugiemu, drugie - trzeciemu, itd.. Ostatnie dziecko podaje pudełko pierwszemu.

W czasie zabawy gra muzyka. Gdy melodia ucichnie, dziecko, które trzyma pudełko, dostaje pyszne ciasteczko i kończy zabawę. Pani Char przerywa muzykę w różnych nieoczekiwanych dla dzieci momentach. Pudełko porusza się raz w prawo, raz w lewo. Zabawa trwa tak długo, aż zostanie tylko jedno dziecko - zwycięzca gry.

Każdy z maluchów chciałby oczywiście wygrać zabawę. Mały Bitek zauważył w pewnym momencie, że pani Char zawsze używa takiej samej sekwencji ruchów. Zauważył też, że może wybrać takie miejsce wśród dzieci, które zapewni mu wygraną. Zastanawia się teraz, które to miejsce. Pomożesz mu?

## Wejście

W pierwszej linii wejścia znajduje się jedna liczba całkowita  $n$  ( $2 \leq n \leq 10^6$ ), oznaczająca liczbę dzieci w przedszkolu.

W kolejnych  $n-1$  liniach wejścia znajduje się sekwencja ruchów wykonywanych przez panią Char, przy czym  $i$ -ty ruch opisany jest przez liczbę całkowitą  $x_i$  ( $1 \leq x_i \leq 100$ ) oznaczającą liczbę przekazania przez dzieci pudełka, po których ucichnie muzyka. Ruchy wykonywane są na przemian raz w prawo, raz w lewo, przy czym pierwszy ruch wykonywany jest zawsze w prawo. Dziecko, które odpadło, oddaje pudełko temu dziecku, od którego je otrzymało.

## Wyjście

Na wyjściu powinna znaleźć się jedna liczba całkowita oznaczająca liczbony od lewej numer miejsca, które powinien zająć mały Bitek, żeby wygrać zabawę.

## Przykład

Wejście	Wyjście
5	3
3	
2	
5	
1	

Objaśnienie przykładu: Pogrubieniem oznaczmy położenie pudełka. Na początku otrzymuje gorące pudełko zawsze pierwsze dziecko: [1,2,3,4,5]. Muzyka cichnie po trzech podaniach, więc odpada dziecko czwarte: [1,2,3,5]. Teraz pudełko przechodzi o dwa w lewo, odpada dziecko pierwsze: [2,3,5]. Dalej - cyklicznie - pięć razy w prawo - odpada piąte: [2,3]. I raz w lewo - zostało tylko dziecko trzecie: [3].

## Ocenianie



Podzadanie	Ograniczenia	Punkty
1	$n \leq 10^3$	30
2	brak dodatkowych założeń	70

# Kamienie

---

Dostępna pamięć: 64MB

Janek jest kolekcjonerem kamieni. Zbiera je i kataloguje, nadając każdemu unikatowy numer. Dbą o to, by każdy kamień był inny i nie pozwala, aby w kolekcji znalazły się dwa takie same. Kiedy chodzi plażą, gdzie może znaleźć nowe okazy do swoich zbiorów, podnosi każdy napotkany kamień, ocenia go, nadaje mu numer katalogowy i – jeśli jeszcze takiego w kolekcji nie posiada – zabiera go. Trudno jest mu jednak szybko ocenić, czy kolejny znaleziony kamień jest potrzebny. Pomóż mu!

## Wejście

W pierwszym wierszu standardowego wejścia zapisana jest jedna liczba całkowita  $n$  ( $2 \leq n \leq 1\,000\,000$ ) oznaczająca liczbę kamieni. Kolejne  $n$  wierszy zawiera po jednej liczbie całkowitej  $k_i$  ( $0 \leq k_i \leq 10^9$ ) oznaczającej numer katalogowy  $i$ -tego kamienia.

## Wyjście

W  $n$  wierszach standardowego wyjścia Twój program powinien zapisać po jednej literze 'T' lub 'N', oznaczającą przydatność kamienia w kolekcji Janka.

## Przykład

Wejście	Wyjście
5	T
1	T
2	T
4	N
2	T
5	

# Loginy

---

Dostępna pamięć: 128MB

Pan Integer pracuje nad otwarciem nowego portalu internetowego. Jednym z elementów systemu jest poczta elektroniczna. Pan Integer rozdzielił już pracę pomiędzy zespoły. Twoim zadaniem będzie opracowanie modułu rejestracji loginów.

Moduł rejestracji otrzymuje proponowany login. Jeśli taka nazwa jeszcze nie wystąpiła w bazie, moduł wysyła informację *OK* do użytkownika i zapisuje go w bazie. Jeśli podana nazwa już występuje w bazie, wówczas tworzony jest nowy login, system wysyła go do użytkownika i zapisuje w bazie. Nowy login tworzony jest przez dodanie kolejnego jak najmniejszego numeru do proponowanej przez użytkownika nazwy.

## Wejście

W pierwszej linii wejścia znajduje się jedna liczba całkowita  $n$  ( $1 \leq n \leq 5 \cdot 10^5$ ), oznaczająca liczbę rozpatrywanych loginów.

W kolejnych  $n$  liniach znajdują się rozpatrywane loginy. Każdy login składa się wyłącznie z małych liter alfabetu łacińskiego i jest nie dłuższy niż 30 znaków.

## Wyjście

Na wyjściu dla każdego loginu wypisywana jest odpowiedź wysyłana dla użytkownika: *OK* – w przypadku udanej rejestracji, lub nowy login dla już istniejącej nazwy.

## Przykład

Wejście	Wyjście
6	OK
mirek	OK
katarzyna	mirek1
kirek	mirek2
mirek	OK
nowy	mirek3
mirek	

## Ocenianie

Podzadanie	Ograniczenia	Punkty
1	$n \leq 10^3$	30
2	brak dodatkowych założeń	70

# Izba przyjęć

Dostępna pamięć: 256MB

Na izbie przyjęć chorzy pacjenci oczekują na przyjęcie z przejęciem patrząc na zegarek. Ale to nie czas decyduje o kolejności. Ważniejszy jest stan chorego. Dlatego często na SORach oznacza się pacjentów opaskami w kolorach czerwonym, żółtym i zielonym. Kolor czerwony oznacza, że pacjent musi być przyjęty w pierwszej kolejności - w trybie natychmiastowym. Taki kolor przydzielany jest osobom w bardzo złym stanie, często przywiezionym przez karetki. Kolor żółty jest przydzielany pacjentom, którzy wymagają pilnej interwencji lekarza. Kolorem zielonym są oznaczani pacjenci z najniższym stopniem zagrożenia życia, przyjmowani w trzeciej kolejności.

Pan Integer opracował dokładniejszy sposób określania stanu pacjentów. Wprowadził  $k$  kolorów opasek. Jeśli mamy pacjentów z opaskami w kolorach  $i$  oraz  $j$  i wiemy, że  $i < j$ , to pacjent z opaską w kolorze  $i$  zostanie przyjęty jako pierwszy. Dla pacjentów z tym samym kolorem opaski liczy się czas przyjęcia.

Pan Integer wymyślił system, który ma przyspieszyć pracę pobliskiej izby przyjęć i obserwuje teraz, jak działa on w rzeczywistości. Co sekundę przychodzi nowy pacjent lub jeden z oczekujących pacjentów jest przyjmowany przez dyżurującego lekarza. Pan Integer potrafi natychmiast sklasyfikować przybyłych (określić kolor ich opasek). Nie nadąża jednak ze wskazywaniem pacjentów, którzy mają być w danym momencie obsłużeni. Czy potrafisz napisać program, który pomoże mu w pracy?

## Wejście

W pierwszej linii wejścia znajduje się jedna parzysta liczba całkowita  $n$  ( $1 \leq n \leq 2 \cdot 10^6$ ). W kolejnych  $n$  liniach znajduje się po jednej liczbie całkowitej  $k_i$  oznaczającej kolor opaski pacjenta, który przyszedł w  $i$ -tej sekundzie ( $1 \leq k_i \leq 109$ ) lub informacja o przyjęciu w tym momencie kolejnej osoby przez lekarza ( $k_i = -1$ ).

Możesz założyć, że zawsze jest ktoś, kto oczekuje pomocy i że wszyscy pacjenci zostaną kiedy obsłużeni. Możesz również założyć, że testach ocenianych na 30 punktów zachodzi warunek  $n \leq 10^3$ .

## Wyjście

Na wyjściu powinno znaleźć się  $n$  liczb całkowitych oznaczających kolejność przyjęcia pacjentów przez lekarza. Pacjentów oznacz czasem przyjścia na izbę przyjęć.

## Przykład

Wejście	Wyjście
10	3
3	1
5	7
2	5
-1	2
4	
-1	
1	
-1	

-1	
-1	

# Bursztyn

Dostępna pamięć: 32MB

Czy wiedzieliście, że w województwie lubelskim są ogromne złoża bursztynu, porównywalne z tymi na wybrzeżu Bałtyku? Wójt jednej z lubelskich gmin postanowił na własną rękę eksploatować złoża znajdujące się na zarządzanym przez niego terenie. W tym celu zabezpieczył już w przyszłorocznym budżecie pewną kwotę na zakup terenów bursztynonośnych. Ze względu na zakupiony sprzęt teren ten musi mieć kształt kwadratu o określonym wymiarze  $k$ . Wójt sporządził już mapę terenu z informacjami o ilości bursztynu na poszczególnych działkach. Teraz tylko chciałby wybrać taki kwadrat, którego eksploracja przyniesie mu największy zysk.

## Wejście

W pierwszym wierszu standardowego wejścia zapisano trzy liczby całkowite dodatnie, oddzielone pojedynczym odstępem:  $n$ ,  $m$  i  $k$ , gdzie ( $1 \leq n, m, k \leq 1000$ ). W kolejnych  $n$  wierszach zapisano po  $m$  liczb całkowitych nieujemnych, które odpowiadają ilości bursztynu na poszczególnych działkach (wielkości te nie przekraczają miliona).

## Wyjście

W pierwszym wierszu standardowego wyjścia zapisz największą sumaryczną ilość bursztynu kwadratowego obszaru.

## Przykład

<b>Wejście</b> 5 6 2 1 2 3 4 5 6 5 4 3 2 1 6 3 4 5 6 7 1 4 5 6 7 8 1 8 9 1 2 3 1	<b>Wyjście</b> 28
--	----------------------

<b>Wejście</b> 4 5 3 1 2 3 4 5 5 4 3 2 1 3 4 5 6 7 4 5 6 7 8	<b>Wyjście</b> 45
---	----------------------

<b>Wejście</b> 4 4 2 1 2 3 4 5 4 3 2 3 4 5 6 4 5 6 7	<b>Wyjście</b> 24
---	----------------------

# Na rzymskie

---

Dostępna pamięć: 32MB

Wczytaj liczbę zapisaną w systemie dziesiętnym i wypisz jej wartość zapisaną cyframi rzymskimi. Wartości cyfr pojawiających się w testach przedstawione są w tabelce poniżej.

I	1
IV	4
V	5
IX	9
X	10
L	50
C	100
D	500
M	1000

Wejście

Liczba arabska  $n$  ( $1 \leq n \leq 3000$ ).

Wyjście

Pierwszy i jedyny wiersz wyjścia powinien wypisać jedno słowo – wartość  $n$  w zapisie rzymskim.

Przykład

Wejście 55	Wyjście LV
---------------	---------------

# Dwukolorowanie

Dostępna pamięć: 64MB

Twierdzenie o czterech barwach mówi, że każdą mapę przedstawioną na płaszczyźnie można pokolorować z użyciem czterech kolorów w taki sposób, żeby żadne dwa sąsiadujące obszary nie były pomalowane na ten sam kolor. Problem ten był otwarty przez ponad sto lat i został udowodniony dopiero w roku 1976 za pomocą komputera.

Twoim zadaniem jest rozwiązanie prostszego problemu. Sprawdź, czy dany graf (spójny, nieskierowany i bez pętli) jest dwukolorowalny, tzn. czy jego wierzchołki mogą być pomalowane na kolory czerwony i czarny w taki sposób, aby sąsiadujące ze sobą wierzchołki nigdy nie były tego samego koloru.

## Wejście

Dane wejściowe składają się z pewnej liczby zestawów testowych. W pierwszym wierszu każdego zestawu znajduje się liczba wierzchołków  $n$  ( $1 \leq n \leq 100\,000$ ). Etykietą każdego wierzchołka jest liczba z zakresu od 0 do  $n-1$ . Drugi wiersz zawiera liczbę krawędzi  $k$  ( $1 \leq k \leq 1\,000\,000$ ). Każdy z kolejnych  $k$  wierszy zawiera numery dwóch wierzchołków – numery te opisują krawędź.

## Wyjście

Sprawdź, czy graf jest dwukolorowalny, i wypisz wynik: **TAK** – jeśli graf jest dwukolorowalny, lub **NIE** – w przeciwnym wypadku.

## Przykład

Wejście	Wejście
3	9
3	8
0 1	0 1
2 0	0 2
1 2	0 3
	0 4
Wyjście	4 5
NIE	4 6
	4 7
	4 8
	Wyjście
	TAK



# Prehistoria

---

Limit pamięci: 64 MB

Na dziewiczych, zamieszkałych przez dinozaury, mamuty, wiewiórki biegające za żołędziem i inne dziwne stwory terenach rozciąga się kraina zwana przez jej pierwotnych mieszkańców Bajtolandią. Na owym obszarze żyje  $n$  plemion. Między niektórymi z nich są dwukierunkowe drogi, których jest w sumie  $m$ . Wodzowie postanowili połączyć ich plemiona w sojusze, aby łatwiej było im przetrwać. Zasada łączenia jest prosta: plemię  $a$  zawiąże sojusz z plemieniem  $b$  jeśli:

- istnieje droga łącząca plemię  $a$  i plemię  $b$ ,

lub

- istnieje takie plemię  $c$ , że zarówno plemię  $a$  i plemię  $b$  mają z nim sojusz.

Proces zawierania sojuszy trwa tak długo, aż nie da się zawrzeć żadnego innego.

Ludzie zamieszkujący te tereny są prymitywni, aczkolwiek bardzo inteligentni, więc zgadali się i założyli Koło Informatyczne Bajtolandii (KIB). Zasada przyjęcia do KIBu jest prosta. Obecni członkowie zadają kandydatowi  $q$  pytań o treści "Czy plemię  $p$  jest w sojuszu z plemieniem  $k$ ". Jaskiniowcy mają już swoje komputery stworzone z kamieni oraz śladowych ilości drewna, lecz mimo to odpowiadanie na owe pytania sprawia im trudność. Napisz odpowiedni program i pomóż im dostać się do KIBu.

## Zadanie

Napisz program, który:

- wczyta ze standardowego wejścia ilość plemion, dróg, opisy tych dróg, ilość pytań KIBu, oraz ich opisy.
- wyliczy odpowiedzi na zapytania,
- wypisze wynik na standardowe wyjście.

## Wejście

W pierwszym wierszu standardowego wejścia zapisane są dwie liczby całkowite  $n$  i  $m$  ( $1 \leq n, m \leq 1000000$ ) oznaczające odpowiednio ilość plemion i dróg w Bajtolandii. W następnych  $m$  wierszach znajdują się opisy tych dróg, składające się z dwóch liczb całkowitych  $a$  i  $b$  ( $1 \leq a, b \leq n$ ,  $a \neq b$ ) oznaczające, że plemię  $a$  i plemię  $b$  łączy droga. W następnym wierszu znajduje się jedna liczba całkowita  $q$  ( $1 \leq q \leq 1000000$ ), a w następnych  $q$  wierszach po dwie liczby całkowite  $p$  i  $k$  ( $1 \leq p, k \leq n$ ,  $p \neq k$ ), oznaczające pytanie KIBu. W testach wartych około 40% punktów zachodzi warunek ( $1 \leq n, m, q \leq 1000$ ).

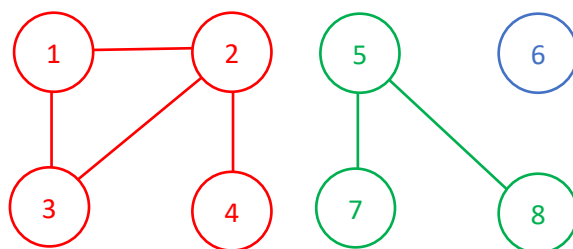
## Wyjście

Twój program powinien wypisać na wyjście  $q$  wierszy, w  $i$ -tym z nich powinno znajdować się pojedyncze słowo TAK, jeśli odpowiedź na  $i$ -te pytanie KIBu jest twierdząca, lub NIE w przeciwnym wypadku.

## Przykład

Wejście	Wyjście
8 6	TAK
1 2	NIE
5 7	TAK
3 2	NIE
5 8	
1 3	
2 4	
4	
1 4	
6 7	
7 5	
3 8	

Ilustracja przykładu:



# Mysz w labiryncie

---

Dostępna pamięć: 32MB

Do labiryntu trafiła mysz. Pomóż jej odnaleźć wyjście!

## Wejście

W pierwszej linii wejścia znajdują się dwie liczby całkowite  $n$  i  $m$  ( $2 \leq n, m \leq 1000$ ) oznaczające rozmiar labiryntu:  $n$  kolumn i  $m$  wierszy. W kolejnych liniach znajdują się znaki oznaczające kolejno:

- - - możliwość przejścia (korytarz);
- x - brak przejścia (ściana);
- o - początkowe położenie myszy;
- w - wyjście (końcowe położenie myszy).

## Wyjście

Jedna liczba całkowita oznaczająca ilość pól odwiedzonych przez mysz, lub słowo „NIE” – jeżeli taka ścieżka nie istnieje.

## Przykład

Wejście	Wyjście
8 8	17
w--x-x--	
xx-x-x--	
---x----	
--xxxx--	
--x--x--	
-----x-o	
xxxx----	
-----x-	

# Grand Prix Bajtocji

---

Dostępna pamięć: 64 MB

W Bajtocji wybudowano nowoczesny tor wyścigowy. Składa się on połączonej ze sobą sieci dróg, które mogą się łączyć ze sobą wyłącznie na skrzyżowaniach. Pomiedzy każdą parą skrzyżowań zaprojektowana została dokładnie jedna droga. Drogi (i skrzyżowania) są na tyle szerokie, że można na nich wygodnie wyprzedzać inne samochody (nawet jeżdżąc w dwóch różnych kierunkach), ale nie można na nich zawracać.

Aby urozmaicić wyścigi właściciele toru starają się umieścić pole startu / mety na różnych skrzyżowaniach. Zastanawiają się teraz, czy w ogóle jest możliwe wytyczenie takiej zamkniętej trasy.

## Wejście

Skrzyżowania ponumerowane są kolejnymi liczbami naturalnymi od 1 do  $n$  ( $1 \leq n \leq 1000$ ).

Pierwszy wiersz danych zawiera dwie liczby naturalne  $n$  (liczba skrzyżowań) oraz  $k$  – liczba dróg ( $1 \leq k \leq 10^6$ ). Kolejne  $k$  wierszy zawiera po dwie oddzielone pojedynczym odstępem liczby naturalne  $u$  i  $w$  oznaczające numery skrzyżowań połączonych drogą ( $1 \leq u, w \leq 1000$ ,  $u \neq w$ ).

## Wyjście

Program powinien wypisać słowo **TAK**, jeśli można wytyczyć trasę o własnościach opisanych w temacie, lub słowo **NIE** – w przeciwnym przypadku.

## Przykład

Wejście:	Wejście
4 3	4 4
1 2	1 2
1 3	1 3
2 4	1 4
	2 4
Wyjście:	Wyjście
NIE	TAK

# Głuchy telefon

---

Dostępna pamięć: 128MB

Śliczna pani Char, przedszkolanka w przedszkolu Słoneczny Integer, lubi urządzać dla swoich podopiecznych różne zabawy. Jedną z nowych zabaw, które pani Char zaproponowała dzieciom, jest zabawa w głuchy telefon. Polega ona na tym, że pani Char szepcze do ucha jakiegoś trudne słowo pierwszemu dziecku, ono powtarza je drugiemu, drugie - trzeciemu, itd... Ostatnie ma wymówić je na głos. Dzieci (jak to dzieci) od razu zaznaczyły, że nie mogą powtarzać podanego słowa dowolnemu dziecku, tylko będą je mówić wyłącznie swojemu najlepszemu koledze lub najlepszej koleżance. Pani Char nie chce zmuszać do niczego maluchów. Zauważyła też od razu, że utworzyły się grupki, podgrupki i jakieś kliki, i że każdej z nich musi podać oddzielnie nowe słowo. Zastanawia się teraz, jaka jest minimalna liczba różnych słów, jaką musi wyszeptać dzieciom do ucha, aby każde z nich wzięło udział w zabawie.

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę całkowitą  $n$  ( $1 \leq n \leq 10^5$ ) - liczbę dzieci. W kolejnych  $n$  wierszach znajduje się numer dziecka  $x_i$ , do którego chce wyszeptać słowo  $i$ -te dziecko ( $1 \leq x_i \leq n$ ).

Wyjście

Wypisz minimalną liczbę dzieci, którym musi wyszeptać do ucha słowa pani Char.

Przykład

Wejście	Wyjście
4	1
2	
3	
4	
3	

# Porządek alfabetyczny

---

Dostępna pamięć: 64MB

Bajtomirek, kolekcjoner niespotykanych książek, odnalazł niedawno starodruk napisany w dziwnym, nieużywanym już języku. Litery w książce były zapisane co prawda znakami łacińskimi, ale kolekcjoner nie potrafił ich rozczytać. Na szczęście książka zawierała na końcu krótki indeks występujących w niej słów, ale kolejność pozycji w indeksie był inny od tego, którego można by się spodziewać. Kolekcjoner opracował już sposób na odczytanie periodyku. Musi tylko poznać kolejność liter w alfabecie z książki. Umożliwi mu w tym indeks słów. Ale wydaje mu się to bardzo żmudnym i mało ciekawym zajęciem. Dlatego, aby zakończyć pracę kolekcjonerską, Bajtomirek poprosił Ciebie, byś na podstawie indeksu słów odtworzył pierwotną kolejność liter alfabetu użytego w książce.

## Wejście

Wejście składa się z uporządkowanej listy słów złożonych wyłącznie z wielkich liter alfabetu łacińskiego. W każdej linii znajduje się jedno słowo nie dłuższe niż 20 znaków. Koniec listy słów sygnalizowany jest znakiem #. Bajtomirek nie wie, czy w książce zostały użyte wszystkie litery alfabetu łacińskiego. Dla każdego zestawu słów istnieje tylko jedno poprawne rozwiązanie.

## Wyjście

Wynikiem działania programu powinien być jeden wiersz zawierający wielkie litery w kolejności zgodnej z alfabetem użytym w książce.

## Przykład

Wejście	Wyjście
ABF	ACFB
CA	
CAF	
CAB	
FBBA	
#	

# Mapa

---

Dostępna pamięć: 128MB

Jaś jak co wieczór spędza czas ze swoim tatą. Postanowił poprosić go o trochę pieniędzy, aby mógł pójść do kina wraz z przyjaciółmi. Tata Jasia, który dba o jego rozwój, wymyślił mu następującą grę.

Tata pokazał mu mapę lasu, który rośnie niedaleko ich domu. Mapa zawiera spis  $n$  polan, które znajdują się w lesie oraz  $m$  ścieżek łączących niektóre z nich. Tata Jasia bardzo lubi chodzić do lasu, żeby podbiegać, ponieważ jest to bardzo zdrowe i buduje kondycje, przed dorocznym biegiem po ulicach Wałbrzycha (przecież nie Bajtocji). Biega on tylko po ścieżkach, z polany na polanę. Ma on jednak swoje upodobania i każdą ścieżką biega tylko w jedną stronę, tzn. każda ścieżka jest dla niego jednokierunkowa.

Tata nie pokazał synkowi tej mapy bez powodu. Zadał on mu pytanie, o to, czy startując z pewnej polany i biegnąc pewnymi ścieżkami (w odpowiednim kierunku) da się wrócić na polanę z której się zaczęło. Innymi słowy tata pyta Jasia czy w tym lasku ma możliwość biegać w kółko. Pytanie jest jednak bardzo trudne i przerasta Jasia. Pomóż mu i napisz odpowiedni program.

## Zadanie

Napisz program, który:

wczyta ze standardowego wejścia liczbę polan, ścieżek oraz ich opis, wyliczy odpowiedzi na zapytanie, wypisze wynik na standardowe wyjście.

## Wejście

W pierwszym wierszu standardowego wejścia zapisane są dwie liczby całkowite nieujemne  $n$  ( $1 \leq n \leq 100000$ ) i  $m$  ( $0 \leq m \leq 500000$ ), oznaczające odpowiednio ilość polan oraz ścieżek. Następne  $m$  wierszy zawiera po dwie liczby całkowite  $a_i$  i  $b_i$  ( $1 \leq a_i, b_i \leq n$ ), oznaczające, że w lasku istnieje ścieżka, którą można przebiec od polany  $a_i$  do polany  $b_i$ . Może się zdarzyć, że  $a_i = a_j$  i  $b_i = b_j$  dla  $i \neq j$ . W testach wartych około 33% punktów zachodzą dodatkowe warunki ( $1 \leq n, m \leq 1000$ ).

## Wyjście

Jeśli w owym lesie da się biegać w kółko na wyjście należy wypisać pojedyncze słowo "TAK". W przeciwnym wypadku należy wypisać "NIE".

## Przykład

Wejście	Wyjście
4 5	TAK
1 2	
2 4	
3 1	
3 4	
2 3	

# Średniowiecze

---

Dostępna pamięć: 128MB

Ponure czasy nastały w Bajtolandii. Dzięki zwierzęta, zbóje i magiczne stwory szaleją i zagrażają bezpieczeństwu Bajtolandczyków, ale jak zwykle to nie to nas interesuje. Interesuje nas sam KIB, który nadal ma się dobrze. Ma też nowe wyzwania dla młodych algorytmików chcących wejść w ich szeregi.

W Bajtolandii istnieje wiele smoczyczych i tajemniczych jam, między nimi jedna, która znajduje się tuż obok siedziby KIBu. Jako, że mistrzowie stowarzyszenia byli w niej wiele razy to sporządzili jej dokładną mapę, która wisi przy wejściu do siedziby. Smocza jama wygląda następująco.

- Jest w niej  $n$  jaskiń, numerowanych od  $1$  do  $n$ .
- Między niektórymi z jaskiń są jednokierunkowe tunele (smok jest bardzo szybki i wszyszkowidzący, więc pilnuje wszystkich tuneli, aby nikt nie chodził nimi w niewłaściwą stronę)(tunel może prowadzić od pewnej jamy z powrotem do niej).
- Każdy tunel ma swój koszt, czyli ilość złota, którą należy zapłacić smokowi, aby przejść danym tunelem (smok pilnuje również, aby każdy zawsze płacił).
- Wejście do jamy to jaskinia o numerze  $1$ , a smoczy skarbca to jaskinia o numerze  $n$ .

Aby od wejścia dostać się do skarbcza należy oczywiście iść różnymi tunelami i wiele zapłacić. Pytanie kwalifikacyjne do KIBu jest następujące. „*Ile należy minimalnie w sumie zapłacić, aby od wejścia dostać się do smoczego skarbcza?*”To zadanie jest jednak bardzo trudne. Pomóż młodym algorytmikom i napisz odpowiedni program.

## Zadanie

Napisz program, który:

- wczyta ze standardowego wejścia ilość jaskiń, tuneli oraz opisy tych tuneli.
- wyliczy odpowiedzi na pytanie KIBu,
- wypisze wynik na standardowe wyjście.

## Wejście

W pierwszym wierszu standardowego wejścia zapisane są dwie liczby całkowite  $n$  i  $m$  ( $1 \leq n \leq 200000$ ,  $1 \leq m \leq 1000000$ ) oznaczające odpowiednio ilość jaskiń i dróg w smoczyczej jamie. W następnych  $m$  wierszach znajdują się opisy tych tuneli, składające się z trzech liczb całkowitych  $a, b$  i  $c$  ( $1 \leq a, b \leq n$ ,  $0 \leq c \leq 10^9$ ) oznaczające, że istnieje tunel zaczynający się w jaskini  $a$ , kończący się w jaskini  $b$ , którego przejście kosztuje  $c$  sztuk złota.

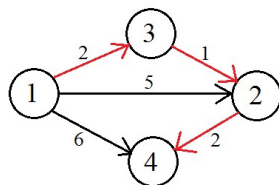
## Wyjście

Twój program powinien wypisać jedną liczbę całkowitą, będącą odpowiedzią na pytanie KIBu. Jeśli nie da się dotrzeć od wejścia do smoczyczego skarbcza, program powinien wypisać  $-1$ .

## Przykład



Wejście	Wyjście
4 5	5
1 2 5	
2 4 2	
1 3 2	
3 2 1	
1 4 6	



# Samochody

Dostępna pamięć: 32MB

W najbliższym czasie zaplanowane zostały zawody samochodowe. Wiadomo, że odbędzie  $k$  wyścigów. Chęć udziału w wyścigach zgłosiło  $n$  samochodów, którym nadano numery od 1 do  $n$ . Wszystkie samochody mają swoją własną maksymalną prędkość początkową. Jednak przed każdym wyścigiem po- zostawiono zawodnikom czas, w którym samochody mogą zostać ulepszone, a ich prędkość maksymalna może wówczas wzrosnąć.

Według zasad zawodów, w  $i$ -tym wyścigu wezmą udział pojazdy o numerach od  $a_i$  do  $b_i$ . Oczywiście ma wygrać najlepszy, co w tym przypadku najprawdopodobniej oznacza: osiągający największą prędkość.

Znasz początkowe prędkości maksymalne wszystkich  $n$  samochodów i plan ich ulepszeń. Wiesz zatem, ile samochodów zostanie ulepszonych przed  $i$ -tym wyścigiem, które to będą pojazdy i o ile wzrośnie prędkość maksymalna każdego z nich. Posiadasz też informację o tym, które samochody wezmą udział w każdym z wyścigów.

Tak rozległa wiedza pozwala wierzyć, że jesteś w stanie przewidzieć wyniki. Napisz program, który pomoże Ci to zrobić.

## Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite:  $n$  i  $k$  ( $1 \leq n \leq 300\,000$ ,  $1 \leq k \leq 300\,000$ ), oznaczające odpowiednio: liczbę wszystkich samochodów oraz liczbę mających się odbyć wyścigów.

W kolejnym wierszu znajduje się  $n$  liczb, gdzie  $i$ -ta liczba to  $v_i$  ( $1 \leq v_i \leq 1\,000\,000$ ), czyli prędkość samochodu o  $i$ -tym numerze.

W kolejnych wierszach znajdują się opisy kolejnych  $k$  wyścigów. Każdy taki opis składa się z wartości  $x_i$  ( $0 \leq x_1 + x_2 + \dots + x_n \leq 300\,000$ ), czyli liczby ulepszeń przed  $i$ -tym wyścigiem,  $x_i$  wierszy zawierających dwie liczby:  $s_i$ ,  $d_i$  ( $1 \leq s_i \leq n$ ,  $0 \leq d_i \leq 1\,000\,000$ ), z których pierwsza oznacza numer ulepszanego samochodu, a druga wartość, o jaką zwiększy się jego prędkość maksymalna, w ostatnim wierszu opisu znajduje się dwie liczby  $a_i$  oraz  $b_i$  ( $1 \leq a_i \leq b_i \leq n$ ), oznaczające, że w danym wyścigu wezmą udział samochody o numerach od  $a_i$  do  $b_i$ .

## Wyjście

Twój program powinien wypisać na standardowe wyjście  $k$  wierszy. W  $i$ -tym wierszu ma znajdować się numer samochodu, który powinien wygrać  $i$ -ty wyścig. Możesz założyć, że odpowiedź zawsze będzie jednoznaczna.

## Przykład

Wejście	Wyjście
5 3	2
7 6 4 12 1	4
2	3
2 7	
1 3	
1 4	

0	
3 5	
3	
5 14	
2 2	
3 14	
1 5	

# Nasionka

---

Dostępna pamięć: 32MB

Ola znalazła pracę w pięknym ogrodzie. Do jej zadań należy troszczenie się o zaopatrzenie ogrodników.

Aktualnie ogrodnicy skończyli planować, jakie rabaty kwiatów urządzą w najbliższej przyszłości. Są bardzo dokładni, więc zrobili to naprawdę szczegółowo. Ola dostała listę, z której dowiedziała się, że chcą oni posiać  $n$  rodzajów kwiatów i będzie im potrzebne  $z_i$  nasionek  $i$ -tego rodzaju.

Teraz Ola, która naprawdę przejmuje się swoją pracą, obawia się, czy fundusze, które ma do wydania na nasionka, są wystarczające, żeby zrealizować całe zamówienie.

Na szczęście okazało się, że jeden ze sklepów ogrodniczych obchodzi dziesięciolecie swojego istnienia i z tej okazji planuje szereg promocji. W najbliższym czasie ma się odbyć  $k$  akcji promocyjnych. W ramach  $j$ -tej akcji będzie można kupić po  $x_j$  nasionek z rodzajów, których numery mieszczą się w przedziale od  $a_j$  do  $b_j$ .

Ola postanowiła skorzystać z okazji i pokazać, że jest odpowiednią osobą do tej pracy. Chce dołożyć wszelkich starań, by zrealizować całe zamówienie, w miarę możliwości nie przekraczając zaplanowanej na to sumy pieniędzy. W tym celu ma zamiar kupić jak najwięcej potrzebnych nasionek w promocyjnej cenie.

Pomóż Oli, która i tak ma bardzo dużo zajęć. Napisz program, który po każdej promocji będzie ją informował, ile rodzajów nasionek udało jej się właśnie skompletować oraz które to rodzaje, tak by mogła skreślić je z listy zakupów. Dzięki Twojej pomocy na pewno będzie jej łatwiej zorientować się, jaką część zamówienia ogrodników już zrealizowała. Ola będzie Ci bardzo wdzięczna.

## Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite:  $n$  i  $k$  ( $1 \leq n \leq 500\,000$ ,  $1 \leq k \leq 100\,000$ ), oznaczające odpowiednio: liczbę rodzajów nasionek oraz liczbę zaplanowanych akcji promocyjnych.

W kolejnym wierszu znajduje się  $n$  liczb całkowitych, gdzie  $i$ -ta liczba oznacza, że ogrodnicy zamówili  $z_i$  ( $1 \leq i \leq n$ ,  $0 \leq z_i \leq 1\,000\,000\,000$ ) nasionek  $i$ -tego rodzaju.

Następne  $k$  wierszy to opisy kolejnych akcji promocyjnych. Każdy taki opis składa się z trzech liczb całkowitych  $a_j$ ,  $b_j$ ,  $x_j$  ( $1 \leq a_j \leq b_j \leq n$ ,  $0 \leq x_j \leq 1\,000\,000\,000$ ), oznaczających, że w ramach  $i$ -tej promocji Ola będzie mogła kupić za połowę ceny po  $x_j$  nasionek z każdego rodzaju o numerze od  $a_j$  do  $b_j$ .

## Wyjście

Twój program powinien wypisać na standardowe wyjście  $k$  wierszy.

W  $i$ -tym wierszu ma na początku znajdować się liczba  $m_i$  oznaczająca, ile rodzajów nasionek udało się skompletować dzięki  $i$ -tej akcji promocyjnej, a następnie  $m_i$  liczb (w kolejności rosnącej) informujących, jakiego rodzaju były to nasionka.

## Przykład

Wejście	Wyjście
8 4	1 5
10 8 14 21 6 73 1 16	1 2
2 5 7	3 3 4 7
1 6 1	0
3 7 20	
4 8 4	

# Proste prostopadłe

---

Dostępna pamięć: 32MB

Dane są dwie proste, każda opisana za pomocą dwóch punktów. Mając dane współrzędne kartezjańskie punktów A, B (pierwsza prosta) oraz C i D (druga prosta) określ, czy te dwie proste są prostopadłe.

Wejście

Pierwszy wiersz danych zawiera liczbę naturalną  $W$  z zakresu od 1 do 1000, określającą ilość zestawów danych, czyli opisów par wektorów do wczytania. Każdy zestaw danych obejmuje dwa wiersze: w pierwszym znajdują się współrzędne pierwszej prostej (punkty A i B), a w drugim – współrzędne drugiej prostej (punkty C i D). Współrzędne punktów są liczbami całkowitymi, których wartość bezwzględna nie przekracza stu milionów. Liczby w wierszu oddzielone są od siebie pojedynczymi spacjami.

Wyjście

Program powinien wypisać dla każdego przypadku wiersz tekstu zawierający słowo TAK, jeśli dane wektory są prostopadłe, lub słowo NIE, jeśli nie są prostopadłe.

Przykład

Wejście	Wynik
2	TAK
0 0 1 1	NIE
2 0 1 1	
0 0 2 0	
0 0 2 1	

# Fioletowe lasery

---

Dostępna pamięć: 256MB. Limit czasu: 1s

Bajtomir zajmuje się w Bajtocji wywiadem. Dostał właśnie nowe zadanie. Na podstawie meldunków szpiegów może wskazać położenie czujników ruchu w pewnym niedostępnym budynku. Dowiedział się też, że czujniki można zneutralizować specjalnym fioletowym laserem, ale trzeba to zrobić jednym strzałem. Bajtomir zastanawia się teraz, czy jest to możliwe. Pomóż mu!

## Wejście

W pierwszej linii wejścia znajduje się jedna liczba całkowita  $n$  ( $1 \leq n \leq 10^6$ ), oznaczająca liczbę czujników ruchu w budynku. W kolejnych  $n$  liniach znajdują się po dwie liczby  $x_i$  oraz  $y_i$  ( $-10^6 \leq x_i, y_i \leq 10^6$ ) współrzędne  $i$ -tego czujnika ruchu.

## Wyjście

Na wyjściu powinno znaleźć się jedno słowo: TAK lub NIE - odpowiedź na pytanie z zadania.

## Przykład

<b>Wejście</b> 3 0 0 -5 -5 5 5	<b>Wyjście</b> TAK
--	-----------------------

<b>Wejście</b> 3 0 0 1 2 2 1	<b>Wyjście</b> NIE
--	-----------------------

# Wiśniowy sad

Dostępna pamięć: 128 MB

Bajtomirek zakupił piękny wiśniowy sad. Sad jest piękny, ponieważ wszystkie drzewa posadzono obok siebie w odległości 1 zarówno w pionie, jak i w poziomie. Sad jest ogrodzony zwykłą siatką, a słupki, do których przymocowana jest siatka, są umieszczone w różnych miejscach, zawsze jednak o współrzędnych całkowitych względem położenia drzew. Żadne drzewo nie znajduje się na linii ogrodzenia. Bajtomirek zastanawia się teraz, ile dokładnie jest drzew wiśniowych w jego sadzie. Chciałby to szybko obliczyć, nie wie jednak jak. Pomóż mu!

## Wejście

Pierwszy wiersz zawiera jedną liczbę całkowitą  $n$  – liczbę słupków ogrodzeniowych. Kolejnych  $n$  linii zawiera po dwie liczby całkowite  $x$  oraz  $y$  – współrzędne kolejnych słupków ( $3 \leq n \leq 10^3$ ,  $0 \leq x, y \leq 10^6$ ).

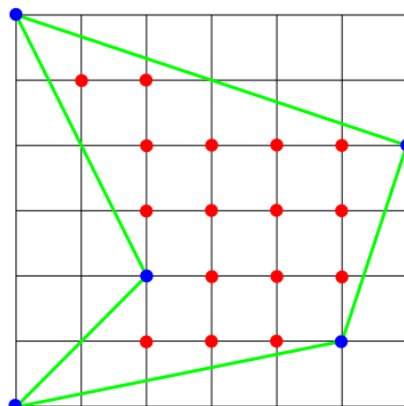
## Wyjście

Program powinien wypisać liczbę drzew wiśniowych znajdujących się wewnątrz ogrodzenia.

## Przykład

Wejście: 5 0 0 5 1 6 4 0 6 2 2	Wejście 16
--	---------------

Ilustracja przykładu:





# Bombardowanie

---

Dostępna pamięć: 128MB

Stało się! Bitocja wypowiedziała wojnę Bajtocji! Wychodząc z założenia, że najlepszą formą obrony jest atak, już pierwszego dnia wojny Bajtocjanie wysłali swoje bombowce z bombami bitowymi nad sąsiednią wyspę. Bomby bitowe zmieniają wszystkie bity w urządzeniach elektronicznych o wartości 1 na losowe siejąc popłoch w dowództwie przeciwnika. Niestety, z powodu gęstej mgły piloci zrzucili swoje ładunki w różnych miejscach wyspy i teraz nie wiadomo, ile z nich na pewno trafiło w terytorium Bitocji. Ponieważ Bitocja ma kształt wielokąta wypukłego o znanych wierzchołkach, a Ty znasz miejsca zrzucenia ładunków, możesz określić tę liczbę!

## Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite  $n$  i  $m$  ( $3 \leq n, m \leq 10^5$ ), odpowiednio liczba punktów określających kształt Bitocji oraz liczba bomb bitowych zrzuconych przez Bajtocję. W kolejnych  $n$  liniach znajdują się po dwie liczby całkowite  $x_i$  i  $y_i$  ( $-10^9 \leq x_i, y_i \leq 10^9$ ) – współrzędne wierzchołków Bitocji. W następnych  $m$  liniach znajdują się po dwie liczby całkowite  $x_j$  i  $y_j$  ( $-10^9 \leq x_j, y_j \leq 10^9$ ) – współrzędne zrzutów bomb bitowych.

## Wyjście

Jedna liczba całkowita oznaczająca, ile bomb na pewno trafiło w terytorium Bitocji.

## Przykład

Wejście	Wyjście
4 3	2
0 0	
5 0	
5 5	
0 5	
1 1	
6 1	
1 4	

# Mur

Dostępna pamięć: 64MB

Kiedy w Bitomiu odkryto złoto, mieszkańcy miasta wpadli w popłoch. Do ich miasta zaczęły ciągnąć dziesiątki, później setki, a w końcu tysiące poszukiwaczy łatwego zarobku. Aby nie dopuścić do katastrofy, postanowiono ogrodzić Bitom murem. Jednocześnie postanowiono, że mur będzie możliwie najmniejszy.

Ktoś szybko naniósł współrzędne domów na mapę, ktoś drugi zaczął już wyrabiać cegły, inni zaczęli składać zamówienia na cement. Ale jaki ma być przebieg muru?

## Zadanie

Napisz program, który:

- wczyta ze standardowego wejścia ilość domów i ich współrzędne,
- wyznaczy domy, obok których będzie bezpośrednio przebiegał mur,
- wypisze wynik na standardowe wyjście.

## Wejście

W pierwszym wierszu standardowego wejścia zapisana jest jedna liczba całkowita  $n$  ( $2 \leq n \leq 500\,000$ ) oznaczająca ilość domów. Kolejne  $n$  wierszy zawiera po dwie liczby całkowite  $a$  i  $b$  ( $-10^9 \leq a, b \leq 10^9$ ) oznaczające współrzędne kolejnych domów. W części testów wartych około 40% punktów zachodzi warunek ( $2 \leq n \leq 2000$ ) oraz ( $-10000 \leq a, b \leq 10000$ ).

## Wyjście

W pierwszym wierszu standardowego wyjścia Twój program powinien wypisać jedną liczbę całkowitą  $X$ , oznaczającą liczbę domów obok których stanie mur. W kolejnych  $X$  liniach znajdą się kolejne współrzędne domów. Jako pierwszy wypisz dom znajdujący się jak najbardziej na zachód. Jeśli jest takich kilka, wypisz ten, który znajduje się najbardziej na południu. Kolejne domy wypisuj w kolejności przeciwnej do ruchu wskazówek zegara.

## Przykład

Wejście	Wyjście
10	8
1 4	1 0
3 3	4 2
4 2	5 3
3 5	6 4
1 2	6 6
5 4	3 5
1 0	1 4
6 6	1 2
5 3	
6 4	

# Czołgiści

---

Dostępna pamięć: 128MB

W Skaryszewie odbywa się doroczny ogólnoswiatowy zjazd czołgistów. Przybyło na niego swoimi czołgami  $n$  czołgistów. Każdy czołg ma nazwę (typ czołgu, nazwa nadana przez czołgistę itp.). Znając wszystkie nazwy czołgów musisz stwierdzić, ile jest par czołgów o tej samej nazwie.

## Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita  $n$  ( $1 \leq n \leq 1000$ ) oznaczająca liczbę czołgistów (i tym samym liczbę czołgów). W kolejnych  $n$  wierszach znajdują się nazwy czołgów. Nazwa składa się wyłącznie z małych liter alfabetu angielskiego i ma nie więcej niż 2000 znaków.

## Wyjście

Na wyjście należy wypisać jedną liczbę całkowitą, oznaczającą ilość par czołgów o tych samych nazwach.

## Przykład

Wejście	Wyjście
5 rudy tygrys rudy rudy tygrys	4

# Ści(ą)gany

---

Dostępna pamięć: 64MB

Janek wymyślił nowy aforyzm. Ostatnio stało się to jego hobby. Zauważył jednak, że niektórzy zaczęli wykorzystywać jego sentencję w swoich wypracowaniach bez pytania go o zgodę. Trochę go to irytuje, więc postanowił sprawdzić, kto ściąga i wypisuje jego maksymę.

## Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita  $n$  ( $1 \leq n \leq 100$ ), oznaczająca liczbę wypracowań do sprawdzenia. W drugim wierszu zapisana jest sentencja Janka: ciąg znaków nie dłuższy niż 1000000 złożony wyłącznie z dużych liter alfabetu łacińskiego. W kolejnych  $n$  liniach znajdują się wypracowania do sprawdzenia (ciągi znaków nie dłuższe niż 1000000 złożony wyłącznie z dużych liter alfabetu łacińskiego, po jednym wypracowaniu w linii).

## Wyjście

W kolejnych  $n$  znajdują się komunikaty *TAK* lub *NIE* - informacja, czy w danym wypracowaniu zawarta jest sentencja Janka.

## Przykład

Wejście	Wyjście
3	TAK
ABC	NIE
ABABABCAB	NIE
ABABABABAB	
AB	

# Czy tu już byłem?

Dostępna pamięć: 256MB

Bajtek zwiedzał Bajtogród przez cały dzień. Wyraźnie widać już po nim zmęczenie. Podobają się mu jednak tutejsze skrzyżowania. Wokół każdego z nich stoi zawsze dokładnie  $n$  budynków. Skrzyżowania różnią się od siebie tylko wysokościami poszczególnych domów. Każde skrzyżowanie jest otoczone budynkami o charakterystycznym dla siebie ciągu wysokości.

Bajtek znalazł się właśnie na kolejnym skrzyżowaniu, ale w żaden sposób nie może sobie przypomnieć, czy wcześniej już tu był. Na szczęście każde skrzyżowanie, które dzisiaj odwiedził, opisał w zeszycie. Który to już raz Bajtek odwiedził aktualne skrzyżowanie?

## Wejście

W pierwszej linii wejścia znajdują się dwie liczby całkowite  $n$  oraz  $k$  ( $1 \leq n \leq 10^5, 1 \leq k \leq 100$ ), oznaczające odpowiednio liczbę budynków na każdym ze skrzyżowań oraz liczbę skrzyżowań, które wcześniej odwiedził Bajtek. W drugiej linii wejścia znajduje się  $n$ -elementowy ciąg wysokości kolejnych budynków na aktualnym skrzyżowaniu  $s_i$  ( $1 \leq s_i \leq 10^6$ ), a jego elementy są oddzielone pojedynczymi odstępami. W kolejnych  $k$  liniach znajdują się opisy odwiedzonych wcześniej przez Bajtka skrzyżowań. Każdy opis to  $n$ -elementowy ciąg wysokości kolejnych budynków na jednym z odwiedzonych skrzyżowań  $o_i$  ( $1 \leq o_i \leq 10^6$ ).

## Wyjście

Na wyjściu w kolejnych  $k$  liniach wypisz komunikat TAK lub NIE - odpowiedź na pytanie, czy Bajtek był już wcześniej na tym skrzyżowaniu. Bajtek mógł odwiedzić aktualne skrzyżowanie wcześniej więcej niż raz.

## Przykład

Wejście	Wyjście
5 3	TAK
1 2 3 4 5	TAK
2 3 4 5 1	NIE
3 4 5 1 2	
3 5 4 1 2	

## Ocenianie

Podzadanie	Ograniczenia	Liczba punktów
1	$n, k \leq 10^2$	25
2	brak dodatkowych założeń	75

# Numer telefonu

---

Dostępna pamięć: 256MB

W Bajtolandii każdy numer telefonu składa się tylko z cyfr 3 oraz 5 i zawsze zaczyna się od 3. Bitek nie lubi pamiętać długich i monotonicznych numerów - zamiast tego woli zapamiętywać prefikso-sufiksy dla każdej z cyfr. Zastanawia się teraz, jak odtworzyć numer telefonu i czy dobrze zapamiętał numer.

## Wejście

W pierwszej linii wejścia znajduje się jedna liczba całkowita  $n$  ( $1 \leq n \leq 10^6$ ), oznaczająca długość tablicy prefikso-sufiksów. W drugiej linii wejścia znajduje się  $n$ -elementowa tablica prefikso-sufiksów  $s_i$  ( $0 \leq s_i \leq 10^6$ ), a jej elementy są oddzielone pojedynczymi odstępami.

## Wyjście

Na wyjściu powinien w pierwszej linii znaleźć się  $n$ -cyfrowy numer telefonu odtworzony z tablicy prefikso-sufiksów lub napis NIE - odpowiedź na pytanie, czy tablica była poprawnie zapamiętana.

## Przykład

Wejście 8 0 0 1 1 2 3 2 3	Wyjście 35335353
Wejście 3 0 0 2	Wyjście NIE

# Nawiasy

---

Dostępna pamięć: 64MB

Kiedy czasem dostaniemy do obliczenia wartość wyrażenia arytmetycznego, już sam pierwszy rzut oka sprawia, że mamy ochotę rozwiązać to zadanie pojutrze i mieć dzięki temu dwa dni wolnego. Najgorsze są jednak przypadki takie, w których po długich i wyczerpujących obliczeniach dowiadujesz się, że otrzymałeś błędny zapis wyrażenia!

Napisz program, który sprawdzi, czy dla podanego zapisu użyte w nim nawiasy umieszczone są w możliwy do przyjęcia sposób (zachowana jest właściwa kolejność nawiasów otwierających i zamykających, każdy nawias musi być sparowany z nawiasem tego samego typu).

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba  $n$  ( $1 \leq n \leq 10$ ). W kolejnych  $n$  wierszach znajduje się po jednym wyrażeniu złożonym z nie więcej niż 1 000 000 znaków. Wyrażenie składa się z nawiasów: (, ), {, }, [, ], <, >.

Wyjście

W  $n$  wierszach standardowego wyjścia Twój program powinien zapisać jedną literę 'T' lub 'N', oznaczającą poprawność zapisu nawiasów w wyrażeniu arytmetycznym.

Przykład

Wejście	Wejście
3	T
( [ ] )	N
< ( > )	T
{ ( ) ( ) }	

# Wojna domowa

Dostępna pamięć: 256MB

Bajtocja przez lata rozwijała się przekształcając się stopniowo w nowoczesny kraj. Jedyne co przeszkadzało w dalszym rozwoju państwa, to słabo rozwinięta sieć drogowa. Król Baltazar postanowił ulepszyć transport w kraju. W tym celu każda droga została zamieniona w jednokierunkowy trakt. W ten sposób zwiększyła się przepustowość dróg, ale nie zawsze było już możliwe dotarcie z dowolnego miasta do każdego innego! Doprowadziło to dalszych napięć w kraju, aż w końcu w Bajtocji wybuchła wojna domowa! W rezultacie rewolucji powstało wiele nowych królestw. Miasta podzieliły się na nowe państwa według bardzo prostej zasady: jeśli istnieje droga z miasta **a** do **b** i z miasta **b** do **a**, to są one częścią tego samego królestwa.

Gildia kupiecka stara się odnaleźć w nowej rzeczywistości. Aby dalej prowadzić interesy, musi wiedzieć, ile królestw powstało w wyniku rewolucji, oraz czy istnieje droga (w dwóch kierunkach) między dwoma wskazanymi miastami.

Napisz program, który pomoże w działalności gildii kupieckiej!

## Wejście

W pierwszej linii wejścia znajdują się dwie liczby całkowite **n** oraz **k** ( $1 \leq n \leq 2 \cdot 10^5$ ,  $1 \leq k \leq 10^6$ ) oznaczające liczbę państw w Bajtocji oraz liczbę istniejących dróg. W kolejnych **k** liniach znajdują się po dwie liczby całkowite **u** oraz **w** ( $1 \leq u, w \leq n$ ,  $u \neq w$ ) oznaczające odpowiednio miasto początkowe oraz miasto końcowe, pomiędzy którymi istnieje droga po reformach Bajtazara.

W **k + 1** linii wejścia znajduje się jedna liczba całkowita **m** ( $1 \leq m \leq 10^5$ ) oznaczająca liczbę par miast do sprawdzenia. W kolejnych **m** liniach znajduje się po dwie liczby całkowite **a** oraz **b** ( $1 \leq a, b \leq n$ ) oznaczające numery miast, dla których należy sprawdzić, czy istnieje droga (w obu kierunkach).

## Wyjście

Na wyjściu w pierwszym wierszu powinna znaleźć się jedna liczba całkowita oznaczająca liczbę różnych państw, na które podzieliła się Bajtocja.

W kolejnych **m** liniach znajdują się odpowiedzi na pytania, czy istnieje dwukierunkowa droga między kolejną parą miast **a** i **b** (**TAK** lub **NIE**).

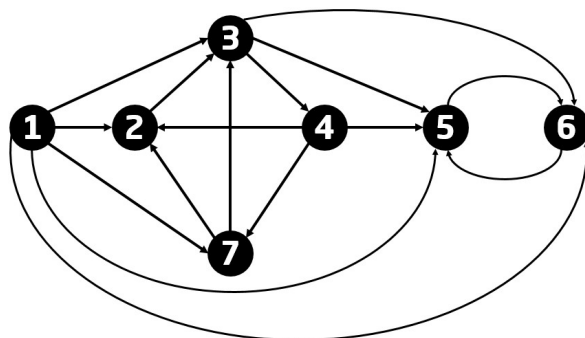
## Przykład

Dla danych wejściowych:	poprawnym wynikiem jest:
7 16	3
1 2	TAK
2 3	NIE
3 4	TAK
4 2	TAK
7 2	
4 5	
5 6	
6 5	



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

Przykładowe państwo:



Ocenianie

Podzadanie	Ograniczenia	Punkty
1	$n, k, m \leq 10^3$	30
2	brak dodatkowych założeń	70