

Parzyste kryształy (A)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

W pełnym tramwajów mieście mieszka N krasnali. Każdy z nich zebrał pewną liczbę kryształów, ale każdy inną, oznaczoną jako A_1, A_2, \dots, A_N .

Sprawdź, czy da się wybrać dwóch różnych krasnali tak, aby łączna liczba ich kryształów była liczbą parzystą. Jeśli tak, należy znaleźć największą możliwą taką wartość.

Wejście

W pierwszym wierszu wejścia znajduje się liczba naturalna N oznaczająca liczbę krasnali.

W drugim wierszu wejścia znajduje się N liczb całkowitych nieujemnych oznaczających liczbę kryształów każdego z krasnali.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć maksymalna parzysta suma liczby kryształów dwóch różnych krasnali lub -1 , jeżeli takiej nie ma.

Ograniczenia

$2 \leq N \leq 200\,000$, $0 \leq A_i \leq 10^9$,
wartości A_i są parami różne.

Przykłady

Wejście

3
2 3 4

Wyjście

6

Wyjaśnienie

Wartości, które można przedstawić jako sumę dwóch różnych elementów ciągu A , to 5, 6 oraz 7. Wśród nich znajduje się jedna liczba parzysta, więc odpowiedź to 6.

Wejście

2
1 0

Wyjście

-1

Wyjaśnienie

Wartość, którą można przedstawić jako sumę dwóch różnych elementów ciągu A , to 1. Nie jest to liczba parzysta, więc należy wypisać -1 .

Wejście

4
5 6 7 8

Wyjście

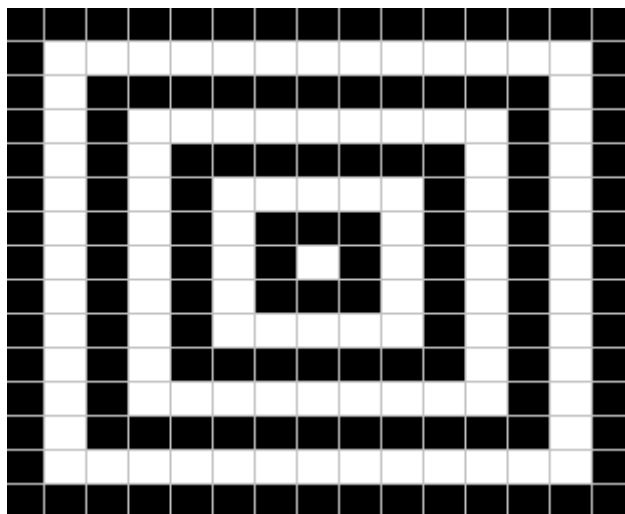
14

Antyczna Plansza (B)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Krasnal archeolog odkrył ostatnio we Wrocławiu antyczną planszę. Ma ona 15 wierszy i 15 kolumn, a każde jej pole jest pomalowane na biało lub czarno. Plansza wygląda następująco:



Twoim zadaniem jest napisanie programu, który wypisze B lub C w zależności od tego, czy pole w N -tym wierszu od góry i M -tej kolumnie od lewej jest odpowiednio białe czy czarne.

Wejście

W pierwszym (jedynym) wierszu wejścia znajdują się dwie liczby N i M – wiersz i kolumna pola, o które pyta krasnal archeolog.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna znaleźć się litera B, jeśli pole jest białe, lub C, jeśli jest czarne.

Ograniczenia

$1 \leq N, M \leq 15$.

Przykłady

Wejście

3 5

Wyjście

C

Wyjaśnienie

Pole w trzecim wierszu i piątej kolumnie jest czarne.

Wejście

4 5

Wyjście

B

Wyjaśnienie

Pole w czwartym wierszu i piątej kolumnie jest białe.

Wejście

8 8

Wyjście

B

Kompleksy krasnali (c)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Wrocławskie krasnale są nieco przewrażliwione na punkcie swojego wzrostu. Żeby poprawić sobie samoocenę, postanowiły policzyć swój łączny wzrost – razem zawsze wyglądają na wyższe.

W grupie znajduje się N krasnali. Najniższy krasnal ma A centymetrów wzrostu, a najwyższy B centymetrów. Wzrost każdego skrzata wyraża się całkowitą liczbą centymetrów.

Krasnale zastanawiają się, ile różnych wartości może mieć suma wzrostów wszystkich N krasnali.

Wejście

W pierwszym (jedynym) wierszu wejścia znajdują się trzy liczby całkowite N , A , B , oddzielone pojedynczymi odstępami, oznaczające kolejno liczbę krasnali, wzrost najniższego krasnala oraz wzrost najwyższego krasnala.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna liczba całkowita, oznaczająca liczbę różnych wartości, jakie może mieć suma wzrostów wszystkich N krasnali.

Jeśli podane liczby nie dają się pogodzić z opisem (krasnale się pomyliły przy pomiarze), na wyjściu należy wypisać 0.

Ograniczenia

$1 \leq N, A, B \leq 10^9$.

Przykłady

Wejście

4 4 6

Wyjście

5

Wyjaśnienie

Mamy 5 możliwych sum:

$18 = 4 + 4 + 4 + 6$, $19 = 4 + 4 + 5 + 6$,

$20 = 4 + 4 + 6 + 6$, $21 = 4 + 5 + 6 + 6$,

$22 = 4 + 6 + 6 + 6$.

Wejście

5 4 3

Wyjście

0

Wyjaśnienie

Najniższy krasnal ma 4 cm, a najwyższy 3 cm — tak się nie da, więc wynik to 0.

Wejście

1 7 10

Wyjście

0

Wyjaśnienie

Jest jeden krasnal, więc „najniższy” i „najwyższy” to ta sama osoba; podane 7 i 10 się wykluczają, stąd odpowiedź 0.

Wejście

1 3 3

Wyjście

1

Krasnal Ustawiacz (D)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

We Wrocławiu zbliża się Wielki Festiwal Krasnali. Z tej okazji N krasnali z całego miasta (i okolicznych osiedli) ma utworzyć na wrocławskim Rynku paradny szereg. Każdy z przybywających krasnali nosi na czapce numer, przy czym numery te mogą się *powtarzać*. Wiadomo, że krasnale będą przybywać na Rynek w określonej kolejności, a ich numery to A_1, A_2, \dots, A_N .

Za organizację szyku odpowiada Krasnal Ustawiacz, który słynie ze swoich żartów. Początkowo szereg jest pusty. Za każdym razem, gdy na Rynku zjawia się i -ty krasnal (z numerem A_i), Ustawiacz wykonuje dwie czynności:

1. Nakazuje nowo przybyłemu krasnalowi stanąć na samym końcu obecnego szeregu.
2. Natychmiast po tym uderza swoim mosiężnym kilofem w brukową kostkę, wywołując magiczne zawirowanie, które odwraca kolejność wszystkich krasnali stojących obecnie w szeregu (ten, kto stał pierwszy, staje się ostatnim, drugi przedostatnim itd.).

Napisz program, który wyznaczy ostateczny układ szeregu wrocławskich krasnali po wykonaniu tych N operacji.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N , oznaczająca liczbę krasnali biorących udział w paradzie.

W drugim wierszu znajduje się N liczb całkowitych A_1, A_2, \dots, A_N oddzielonych pojedynczymi odstępami, oznaczających numery na czapkach kolejno przybywających krasnali.

Wyjście

Twój program powinien wypisać na standardowe wyjście dokładnie jeden wiersz. Powinno się w nim znaleźć N liczb całkowitych oddzielonych pojedynczymi odstępami — ostateczna kolejność numerów krasnali w szeregu na wrocławskim Rynku.

Ograniczenia

$$1 \leq N \leq 200\,000, 0 \leq A_i \leq 10^9.$$

Przykłady

Wejście

```
4
10 20 30 40
```

Wyjście

```
40 20 10 30
```

Wyjaśnienie

- Początkowo szereg jest pusty: []
- Krasnal 10 dołącza i szereg jest odwracany: [10] → [10]
- Krasnal 20 dołącza: [10, 20] → [20, 10]
- Krasnal 30 dołącza: [20, 10, 30] → [30, 10, 20]
- Krasnal 40 dołącza: [30, 10, 20, 40] → [40, 20, 10, 30]

Ostateczny szyk na Rynku to:
40 20 10 30.

Wejście

6
1 9 2 3 7 2

Wyjście

2 3 9 1 2 7

Wejście

2
1 4

Wyjście

4 1

Wejście

6
0 6 7 6 7 0

Wyjście

0 6 6 0 7 7

Tabliczka Krasnali (Easy) (E)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Uwaga! To jest **łatwiejsza** wersja zadania Tabliczka Krasnali. W tej wersji $N = 9$. Reszta treści jest bez zmian.

Jasiu bawi się swoją kolekcją krasnali. Postanowił ułożyć je w kwadratową siatkę o wymiarach $N \times N$. Zauważył, że krasnal stojący w i -tym rzędzie i j -tej kolumnie posiada dokładnie $i \times j$ złotych monet (dla $1 \leq i, j \leq N$).

Jasiu ma jednak swoją nielubianą liczbę X . Postanowił podliczyć całkowitą sumę złotych monet posiadanych przez wszystkie krasnale, **pomijając** jednak te krasnale, które mają dokładnie X złotych monet.

Twoim zadaniem jest napisanie programu, który obliczy i wypisze sumę złotych monet wszystkich krasnali, które nie posiadają ich dokładnie X .

Wejście

W pierwszym (i jedynym) wierszu wejścia znajduje się jedna liczba całkowita X – nielubiana liczba Jasia.

Wyjście

W pierwszym wierszu wyjścia powinna się znaleźć jedna liczba całkowita – łączna suma złotych monet krasnali, które nie mają ich dokładnie X .

Ograniczenia

$$N = 9, 1 \leq X \leq N^2.$$

Przykłady

Wejście

1

Wyjście

2024

Wyjaśnienie

W tym przykładzie $X = 1$. Tylko jeden krasnal (w pierwszym rzędzie i pierwszej kolumnie) ma dokładnie 1 monetę. Suma monet wszystkich pozostałych krasnali wynosi 2024.

Wejście

11

Wyjście

2025

Wyjaśnienie

W tym przykładzie $X = 11$. Żaden z krasnali w siatce 9×9 nie posiada dokładnie 11 monet, więc sumujemy monety wszystkich krasnali, co daje 2025.

Wejście

2

Wyjście

2021

Tabliczka Krasnali (Hard) (F)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Uwaga! To jest **trudniejsza** wersja zadania Tabliczka Krasnali. W tej wersji $N = 10^6$. Reszta treści jest bez zmian.

Jasiu bawi się swoją kolekcją krasnali. Postanowił ułożyć je w kwadratową siatkę o wymiarach $N \times N$. Zauważył, że krasnal stojący w i -tym rzędzie i j -tej kolumnie posiada dokładnie $i \times j$ złotych monet (dla $1 \leq i, j \leq N$).

Jasiu ma jednak swoją nielubianą liczbę X . Postanowił podliczyć całkowitą sumę złotych monet posiadanych przez wszystkie krasnale, **pomijając** jednak te krasnale, które mają dokładnie X złotych monet.

Twoim zadaniem jest napisanie programu, który obliczy i wypisze sumę złotych monet wszystkich krasnali, które nie posiadają ich dokładnie X .

Wejście

W pierwszym (i jedynym) wierszu wejścia znajduje się jedna liczba całkowita X – nielubiana liczba Jasia.

Wyjście

W pierwszym wierszu wyjścia powinna się znaleźć jedna liczba całkowita – łączna suma złotych monet krasnali, które nie mają ich dokładnie X .

Ograniczenia

$$N = 10^6, 1 \leq X \leq N^2.$$

Przykłady

Wejście

1

Wyjście

250000500000249999999999

Wyjaśnienie

W tym przykładzie $X = 1$. Tylko jeden krasnal (w pierwszym rzędzie i pierwszej kolumnie) ma dokładnie 1 monetę. Suma monet wszystkich pozostałych krasnali wynosi 250000500000249999999999.

Wejście

11

Wyjście

250000500000249999999978

Wyjaśnienie

W tym przykładzie $X = 11$. Krasnale mające 11 monet stoją tylko na polach (1, 11) oraz (11, 1). Suma monet wszystkich pozostałych krasnali wynosi 250000500000249999999978.

Wejście

2

Wyjście

250000500000249999999996

Korporacja (G)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Krasnal Bitek dostał właśnie nową pracę w technologicznej korporacji Facebajt. Jako menedżer ma pod swoją opieką N zespołów programistycznych, liczących odpowiednio A_1, A_2, \dots, A_N osób.

W ramach firmowej restrukturyzacji Bitek może wielokrotnie wybrać dwa zespoły liczące x oraz y pracowników i połączyć je w jeden nowy zespół liczący $x + y$ osób. Każda taka operacja zmniejsza łączną liczbę zespołów o 1.

Zbliża się koniec kwartału i Bitek musi przygotować raport. Aby statystyki ładnie prezentowały się przed zarządem, średnia liczba osób w zespole **po wszystkich połączeniach** musi być liczbą całkowitą.

Jaka jest minimalna liczba połączeń zespołów, które Bitek musi wykonać, aby osiągnąć ten cel? Można pokazać, że osiągnięcie tego celu jest zawsze możliwe.

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę naturalną N .

Drugi wiersz wejścia zawiera N liczb naturalnych A_1, A_2, \dots, A_N , oddzielonych pojedynczymi odstępami, oznaczających liczbę osób w kolejnych zespołach.

Wyjście

Na wyjściu wypisz jedną liczbę całkowitą: minimalną liczbę połączeń, które musi wykonać Bitek.

Ograniczenia

$$1 \leq N \leq 10^6, 1 \leq A_i \leq 1000.$$

Przykłady

Wejście

3
1 1 1

Wyjście

0

Wyjaśnienie

Średnia już wynosi 1, więc Bitek nie musi nic robić.

Wejście

3
3 5 8

Wyjście

1

Wyjaśnienie

Aktualnie średnia wynosi $5\frac{1}{3}$. Bitek może połączyć zespoły o 3 i 5 osobach i dostać dwa 8-osobowe zespoły. Wtedy średnia wynosi 8 i jest już liczbą całkowitą.

Wejście

4
1 2 3 4

Wyjście

2

Czapki (H)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

W ogrodzie Małgosi stoją trzy krasnale, a każdy z nich nosi szpiczastą czapkę. Wysokości czapek od lewej do prawej wynoszą A_1 , A_2 oraz A_3 .

Małgosia chce, żeby jej ogród prezentował się pięknie, dlatego marzy o tym, by ciąg wysokości czapek był *arytmetyczny*, to znaczy, żeby zachodziła równość $A_2 - A_1 = A_3 - A_2$.

Aby osiągnąć swój cel, Małgosia może wykonywać ruchy. W jednym ruchu wybiera ona indeks $i \in \{1, 2, 3\}$ i dokłada do czapki i -tego krasnala kawałek materiału, zwiększając jej wysokość o 1.

Pomóż Małgosi i wyznacz najmniejszą liczbę ruchów, jaką musi wykonać, aby ciąg wysokości czapek był arytmetyczny.

Wejście

W pierwszym (jedynym) wierszu wejścia znajdują się trzy liczby całkowite A_1 , A_2 , A_3 – początkowe wysokości czapek krasnali.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna liczba całkowita – najmniejsza liczba ruchów, jaką musi wykonać Małgosia, aby ciąg A_1 , A_2 , A_3 stał się arytmetyczny.

Ograniczenia

$$1 \leq A_1, A_2, A_3 \leq 10^{15}.$$

Przykłady

Wejście

4 8 10

Wyjście

2

Wyjaśnienie

Małgosia może dwukrotnie zwiększyć wysokość czapki pierwszego krasnala, otrzymując ciąg 6, 8, 10.

Wejście

10 3 4

Wyjście

4

Wyjaśnienie

Małgosia może czterokrotnie zwiększyć wysokość czapki drugiego krasnala, otrzymując ciąg 10, 7, 4, w którym kolejne różnice wynoszą -3 .

Wejście

1 2 3

Wyjście

0

Wyjaśnienie

Ciąg jest już arytmetyczny, więc nie trzeba wykonywać żadnego ruchu.

Wejście

1000000000 1 1000000000

Wyjście

999999999

Kieszonkowe (I)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Krasnal Fontanek troszczy się o dobry stan wrocławskich fontann. Niedawno wyłowił z nich N monet (oczywiście wyłącznie po to, aby nie zapychały odpływów). Wartość i -tej monety wynosi W_i złotych.

Fontanek chce podzielić zebrane monety między swoich dwóch synów — starszego Grosika oraz młodszego Miedzika. Uznał jednak, że dzieci nie powinny dostać wszystkich pieniędzy naraz, więc postanowił wypłacać im kieszonkowe.

Przez najbliższe $\frac{N}{2}$ tygodni każdego tygodnia przekaże każdemu z synów po jednej monecie. Każda moneta zostanie rozdana dokładnie raz.

Jeżeli w danym tygodniu przekazane monety mają różne wartości, droższą monetę zawsze otrzyma starszy syn, czyli Grosik. Jeśli obie monety mają tę samą wartość, Fontanek może rozdać je dowolnie.

Miedzika zadowoli nawet mniejsze kieszonkowe — w końcu wydatki pięciolatka ograniczają się głównie do lodów od czasu do czasu. Mimo tego Fontanek chciałby, aby Miedzik czuł się potraktowany możliwie sprawiedliwie. Dlatego zależy mu na tym, aby łączna wartość monet otrzymanych przez młodszego syna była jak największa (a więc jak najbardziej zbliżona do kwoty otrzymanej przez Grosika).

Pomóż Fontankowi obliczyć maksymalną sumę wartości monet, jaką może otrzymać Miedzik.

Wejście

W pierwszym wierszu wejścia znajduje się jedna parzysta liczba całkowita N — liczba wyłowionych monet.

W drugim wierszu znajduje się N liczb całkowitych W_1, W_2, \dots, W_N oznaczających wartości kolejnych monet.

Wyjście

W pierwszym (jedynym) wierszu wyjścia należy wypisać jedną liczbę całkowitą — maksymalną możliwą łączną wartość monet otrzymanych przez Miedzika.

Ograniczenia

$$2 \leq N \leq 100\,000, 0 \leq W_i \leq 10^9.$$

Przykłady

Wejście

```
6
1 2 10 5 7 4
```

Wyjście

```
12
```

Wyjaśnienie

Fontanek może rozdać monety następująco: w pierwszym tygodniu monety o wartościach 4 i 5, w drugim tygodniu monety o wartościach 10 i 7, w trzecim tygodniu monety o wartościach 2 i 1. Miedzik otrzyma łącznie $4 + 7 + 1 = 12$ złotych kieszonkowego. Można pokazać, że przy żadnym dozwolonym podziale Miedzik nie dostanie więcej niż 12 złotych.

Wejście

```
2
1 1000000000
```

Wyjście

```
1
```

Wejście

```
4
5 5 5 5
```

Wyjście

```
10
```

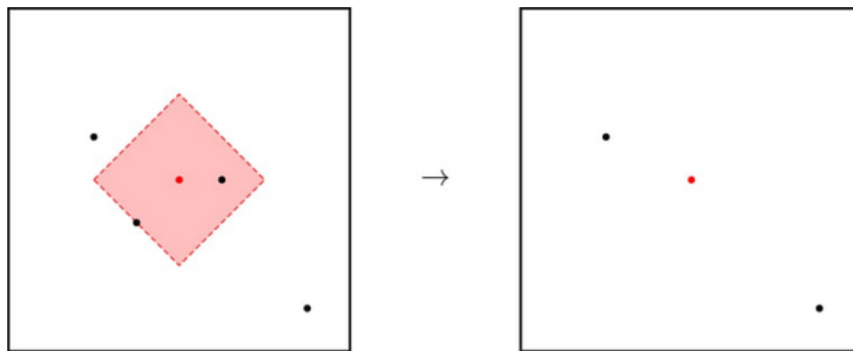
Magnetyczne Krasnale (J)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

We Wrocławiu testowane są nowe *magnetyczne krasnale*. Magnetyczne krasnale, oprócz bycia (uroczą) wizytówką miasta, mogą zostać *naładowane*. Naładowany krasnal natychmiast przyciąga do siebie wszystkie krasnale w pobliżu. Taka funkcjonalność ma wiele zalet, między innymi, być może, da się w ten sposób zebrać wszystkie krasnale w jednym miejscu jedynie poprzez naładowania. Twoim zadaniem jest rozstrzygnąć, czy dla zadanego ustawienia krasnali jest to możliwe, a jeśli tak – jaka minimalna liczba naładowań jest do tego potrzebna.

Dane jest rozstawienie N krasnali. Każdy krasnal znajduje się w pewnym punkcie płaszczyzny, i -ty w punkcie o współrzędnych (X_i, Y_i) . W *pobliżu* danego krasnala są wszystkie krasnale znajdujące się w odległości taksówkowej nie większej niż D od tego krasnala, tj. krasnal j -ty jest w pobliżu krasnala i -tego, jeśli $|X_i - X_j| + |Y_i - Y_j| \leq D$. Naładowanie krasnala skutkuje tym, że wszystkie krasnale w jego pobliżu natychmiast przesuwać się do punktu (X_i, Y_i) , gdzie i to numer naładowanego krasnala. W jednym punkcie może znajdować się wiele krasnali. Naładowań należy dokonywać po kolei, nie można naładowywać wielu krasnali jednocześnie. Powiemy, że krasnale można *zebrać*, jeśli istnieje ciąg naładowań, po którym wszystkie krasnale znajdują się w jednym punkcie.



Przykład operacji naładowania. Po naładowaniu krasnala w środku dwa inne krasnale przesuwać się na jego pozycję. Po prawej czerwona kropka jest wspólną pozycją tych krasnali.

Wejście

W pierwszym wierszu wejścia znajduje się liczba przypadków testowych T .

W pierwszym wierszu każdego przypadku testowego znajduje się liczba naturalna N oraz parametr D .

W każdym z kolejnych N wierszy znajduje się para liczb naturalnych X_i, Y_i , oznaczających współrzędne i -tego krasnala. Pary współrzędnych nie powtarzają się, tzn. nie ma dwóch krasnali początkowo stojących w tym samym miejscu.

Wyjście

Dla każdego przypadku testowego należy wypisać w jednym wierszu pojedynczą liczbę – minimalną liczbę naładowań, jakie są potrzebne do zebrania krasnali, albo -1 , jeśli nie da się zebrać krasnali.

Ograniczenia

$1 \leq T \leq 100$, $2 \leq N \leq 100$, $0 \leq D \leq 10^6$, $0 \leq X_i, Y_i \leq 100\,000$.

Przykłady

Wejście

3
3 2
0 0
3 3
1 1
3 3
6 7
8 8
6 9
4 1
0 0
0 1
0 2
0 3

Wyjście

-1
1
-1

Wyjaśnienie

- W pierwszym przypadku testowym znajdują się trzy krasnale w punktach $(0, 0)$, $(3, 3)$ i $(1, 1)$, a D wynosi 2. Możliwe jest przesunięcie dwóch krasnali do tego samego punktu za pomocą jednego naładowania, ale nie da się zebrać wszystkich trzech krasnali przy użyciu dowolnej liczby naładowań.
- W drugim przypadku testowym znajdują się trzy krasnale w punktach $(6, 7)$, $(8, 8)$ i $(6, 9)$, a D wynosi 3. Jeśli naładujemy dowolnego krasnala, pozostałe dwa przesuną się na tę samą pozycję, więc potrzebujemy tylko jednej operacji naładowania.
- W trzecim przypadku testowym znajdują się cztery krasnale w punktach $(0, 0)$, $(0, 1)$, $(0, 2)$ i $(0, 3)$, a D wynosi 1. Można wykazać, że przesunięcie wszystkich krasnali na tę samą pozycję za pomocą sekwencji naładowań jest niemożliwe.

Wejście

1
2 1000000
0 0
100000 100000

Wyjście

1

Wejście

1
4 0
6 6
6 7
7 6
7 7

Wyjście

-1

Liczby szczęśliwe (K)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Wrocławskie krasnale od pokoleń prowadzą starą księgę, w której zapisują tzw. *liczby szczęśliwe*.

Za *szczęśliwą* uznaje się każdą dodatnią liczbę całkowitą, której zapis dziesiętny składa się wyłącznie z cyfr 4 i 7. Przykładowo liczby 4, 47 i 744 trafiają do księgi, natomiast 5, 17 czy 467 są pomijane.

Krasnale zapisują liczby w księdze w kolejności rosnącej, przeznaczając na każdą z nich osobną stronę. Oznacza to, że:

- na pierwszej stronie znajduje się liczba 4,
- na drugiej 7,
- na trzeciej 44,
- na czwartej 47 i tak dalej.

Pewnego dnia krasnal Szczęśliwek znalazł liczbę szczęśliwą N . Chciałby ustalić, na której stronie jest ona zapisana.

Pomóż Szczęśliwkowi i wypisz numer strony, na której zapisana jest liczba N .

Wejście

W jedynym wierszu wejścia znajduje się jedna liczba całkowita N .

Wyjście

W jedynym wierszu wyjścia powinna się znaleźć jedna liczba całkowita, oznaczająca numer strony księgi, na której zapisana jest liczba N .

Ograniczenia

$1 \leq N \leq 10^{18}$. Gwarantowane jest, że N jest liczbą szczęśliwą.

Przykłady

Wejście

4

Wyjście

1

Wyjaśnienie

4 jest najmniejszą liczbą szczęśliwą, więc znajduje się na stronie 1.

Wejście

74

Wyjście

5

Wyjaśnienie

Liczby szczęśliwe w kolejności rosnącej to: 4, 7, 44, 47, 74. Liczba 74 znajduje się na piątej pozycji.

Wejście

444744

Wyjście

67

Ściąganie (L)

Limit pamięci: 256 MB

Limit czasu: 2.00 s

Ostatnio krasnoludkowi bliźniacy Zgapuś i Ściąguś pisali w szkole test z matematyki, który, jak twierdzili, poszedł im dobrze. Ku zdziwieniu ich mamy obaj otrzymali ocenę *niedostateczną* z dopiskiem *praca niesamodzielna*. Oburzona krasnalka natychmiast poprosiła synów o pokazanie kart odpowiedzi, aby sprawdzić, czy nauczycielka miała rację.

Zasady testu były proste — uczniowie wpisywali rozwiązania kolejnych zadań, każde w osobnej kolumnie; każde rozwiązanie było pojedynczą liczbą całkowitą. Mama postanowiła porównać karty obu krasnoludków kolumna po kolumnie i sprawdzić, w ilu miejscach wpisy synów różnią się od siebie.

Bracia nie chcieli przyznać się do współpracy podczas testu, więc postanowili zmodyfikować swoje karty odpowiedzi tak, aby wyglądały na możliwie najbardziej różne. Mama nie wie, ile było zadań na sprawdzianie, dlatego każdy z nich może wyciąć pewien początkowy fragment oraz pewien końcowy fragment swojej karty (fragmenty mogą być także puste). Zmodyfikowane karty odpowiedzi powinny być równej długości, gdyż różna liczba zadań na tym samym teście mogłaby wzbudzić podejrzenia.

Niestety, szybko okazało się, że znalezienie takich fragmentów, które różnią się na jak największej liczbie pozycji, wcale nie jest łatwiejsze od samego testu z matematyki. Bliźniaki nie poradziły sobie z tym problemem i ostatecznie dostały szlaban na komputer.

Pomóż bliźniakom: dla każdego sprawdzianu wyznacz maksymalną liczbę kolumn, na których mogą się różnić zmodyfikowane karty po wycięciu początkowych i końcowych fragmentów.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna T oznaczająca liczbę przypadków testowych. Opis każdego przypadku testowego składa się z trzech wierszy.

W pierwszym wierszu opisu przypadku testowego znajduje się jedna liczba naturalna N oznaczająca liczbę zadań na teście z matematyki.

W drugim i trzecim wierszu opisu przypadku testowego znajdują się ciągi A_1, A_2, \dots, A_N oraz B_1, B_2, \dots, B_N liczb całkowitych, opisujące odpowiednio odpowiedzi Zgapusia i Ściągusia.

Wyjście

Na wyjściu powinno znaleźć się T wierszy.

W i -tym z nich powinna znaleźć się odpowiedź do i -tego przypadku testowego, będąca jedną nieujemną liczbą całkowitą, oznaczającą maksymalną liczbę kolumn, na których mogą się różnić odpowiedzi bliźniaków po wycięciu początkowych i końcowych fragmentów.

Ograniczenia

$1 \leq T \leq 10\,000$, $1 \leq N \leq 10\,000$, $-10^9 \leq A_i, B_i \leq 10^9$,

suma N po wszystkich przypadkach testowych nie przekracza 10 000.

Przykłady

Wejście

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

Wyjście

```
4
5
0
```

Wyjaśnienie

W pierwszym przypadku testowym Zgapiś mógłby zostawić fragment na pozycjach od 4 do 7, wówczas jego odpowiedzi to $[2, 2, 2, 1]$. Ściaguś mógłby zostawić fragment od 2 do 5, czyli $[1, 1, 1, 2]$. Zmodyfikowane karty odpowiedzi różnią się na wszystkich 4 kolumnach.

W ostatnim przypadku testowym odpowiedź to 0, ponieważ niezależnie od wybranych fragmentów, odpowiedzi krasnoludków będą takie same w każdej kolumnie.

Łamigłówka I (Ł1)

Limit pamięci: 64 MB

Limit czasu: 1.00 s

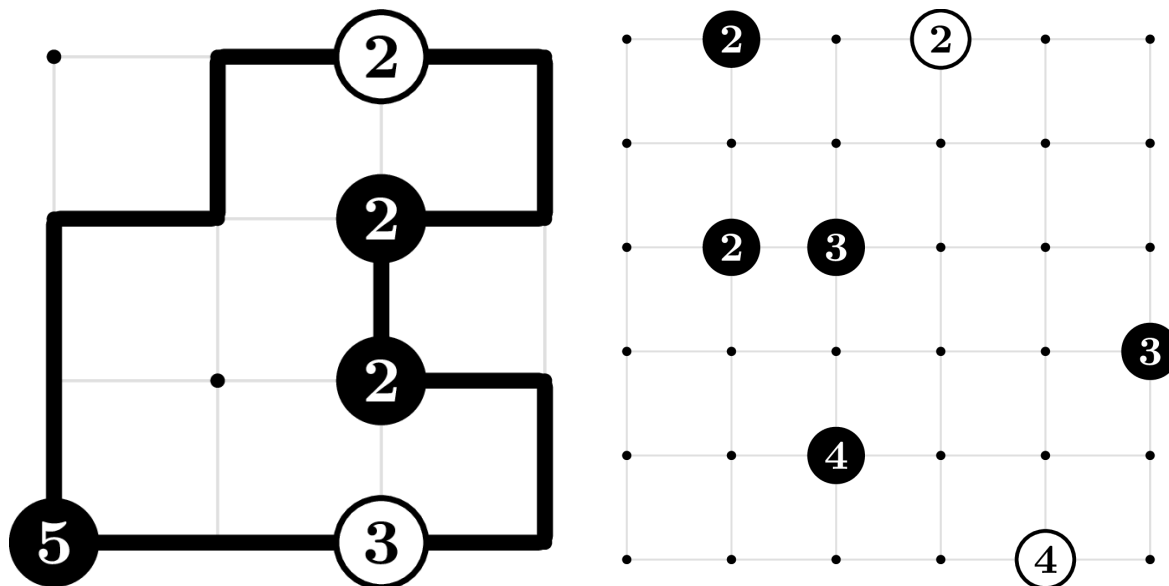
Krasnal Olaf przygotował dla Ciebie zadanie w postaci łamigłówki Shingoki .

Twoim zadaniem jest ją rozwiązać, a więc narysować jedną, zgodną ze wszystkimi zasadami, zamkniętą pętlę, utworzoną z odcinków łączących sąsiednie punkty kratowe.

Zasady są następujące:

- Pętla nie może się ze sobą przecinać i musi być jedynym zamkniętym obszarem na planszy.
- W niektórych punktach znajdują się kółka. Przez każde białe kółko pętla musi przechodzić na wprost, zaś w każdym czarnym kółku pętla musi zakręcać.
- Każda liczba wpisana w środek kółka ma być równa sumie długości dwóch prostych fragmentów pętli, wychodzących z tego kółka.

Lewy rysunek przedstawia przykładową łamigłówkę wraz z jej poprawnym rozwiązaniem. Twoim zadaniem jest rozwiązać łamigłówkę z prawego rysunku. Możesz założyć, że istnieje dokładnie jedno poprawne rozwiązanie.



Wejście

Brak.

Wyjście

Niech łamigłówka składa się z N^2 punktów kratowych ułożonych w kwadrat o N wierszach i N kolumnach. Rozwiązanie należy wypisać jako rysunek złożony z $2N - 1$ wierszy, z których każdy ma długość $2N - 1$ znaków.

W wierszach nieparzystych należy opisać punkty kratowe oraz poziome odcinki pętli. Na pozycjach nieparzystych należy wypisać cyfrę z kółka znajdującego się w danym punkcie albo znak $.$, jeśli w tym punkcie nie ma kółka. Na pozycjach parzystych należy wypisać znak $-$, jeśli pętla zawiera poziomy odcinek między sąsiednimi punktami, albo znak spacji w przeciwnym przypadku.

W wierszach parzystych należy opisać pionowe odcinki pętli. Na pozycjach nieparzystych należy wypisać znak $|$, jeśli pętla zawiera pionowy odcinek między sąsiednimi punktami, albo znak spacji w przeciwnym przypadku. Na pozycjach parzystych należy wypisać znak spacji.

Rozwiązanie lewego przykładu

```
print("""\n  . .-2-\n    |   |\n  .-. 2-\n    |   |\n  . . 2-\n    |   |\n5--.-3--\n""")
```

Szablon rozwiązania w Python

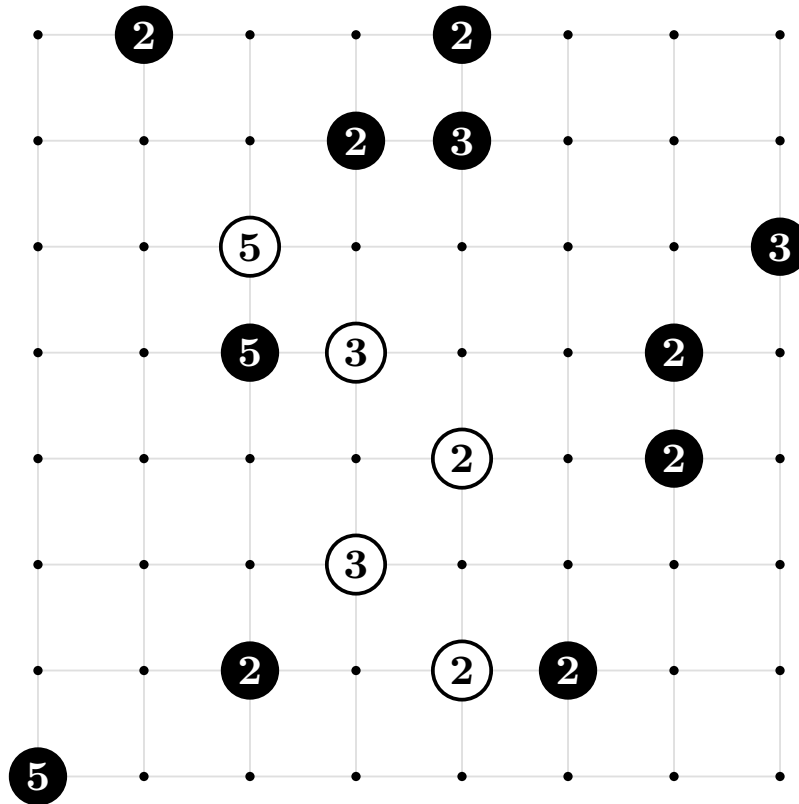
```
print("""\n . 2 . 2 . .\n\n . . . . .\n\n . 2 3 . . .\n\n . . . . . 3\n\n . . 4 . . .\n\n . . . . 4 .\n""")
```

Łamigłówka II (Ł2)

Limit pamięci: 64 MB

Limit czasu: 1.00 s

W drugiej łamigłówce obowiązują te same zasady co poprzednio. Jedyna różnica polega na tym, że plansza ma większy rozmiar.



Szablon rozwiązania w Python

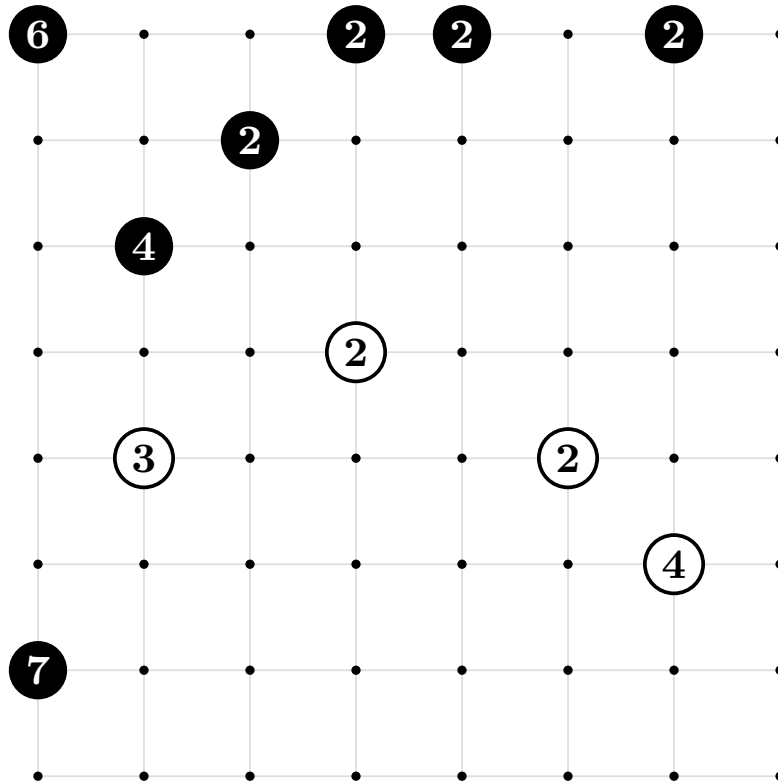
```
print("""\n. 2 . . 2 . . . .\n\n. . . 2 3 . . . .\n\n. . 5 . . . . 3\n\n. . 5 3 . . 2 .\n\n. . . . 2 . 2 .\n\n. . . 3 . . . .\n\n. . 2 . 2 2 . .\n\n5 . . . . . . .\n""")
```

Łamigłówka III (Ł3)

Limit pamięci: 64 MB

Limit czasu: 1.00 s

Trzecia łamigłówka działa według tych samych zasad, ale jest jeszcze trudniejsza.



Szablon rozwiązania w Python

```
print("""\n6 . . 2 2 . 2 .\n\n. . 2 . . . . .\n\n. 4 . . . . .\n\n. . . 2 . . . .\n\n. 3 . . . 2 . .\n\n. . . . . 4 .\n\n7 . . . . . . .\n\n. . . . . . . .\n""")
```

Krasnalowe NWD (M)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Krasnal Bitek ma tablicę długości N . Niedawno poznał pojęcie największego wspólnego dzielnika (NWD). NWD tablicy to największa liczba całkowita d taka, że każdy element tablicy jest podzielny przez d . Bitek bardzo by chciał, żeby NWD jego tablicy był jak największy.

Bitek znalazł ostatnio magiczną różdżkę, która pozwala mu wykonać dowolną liczbę operacji (w szczególności można nie wykonać żadnej). W jednej operacji Bitek wybiera indeks i ($1 \leq i \leq N$) oraz dowolną liczbę całkowitą dodatnią x , a następnie zamienia A_i na $A_i \bmod x$, czyli na resztę z dzielenia A_i przez x .

Ponieważ Bitek bardzo boi się zer, w tablicy w żadnym momencie nie może pojawić się liczba 0. Pomóż Bitkowi obliczyć maksymalny możliwy NWD tablicy, jaki da się uzyskać po wykonaniu kilku (być może żadnych) operacji.

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę całkowitą N .

Drugi wiersz wejścia zawiera N liczb całkowitych A_1, A_2, \dots, A_N , oddzielonych pojedynczym odstępem, oznaczających kolejne elementy tablicy Bitka.

Wyjście

Wypisz jedną liczbę całkowitą – maksymalny możliwy NWD tablicy po wykonaniu kilku (być może żadnych) operacji.

Ograniczenia

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

Przykłady

Wejście

```
3
3 9 11
```

Wyjście

```
3
```

Wyjaśnienie

Można na przykład wybrać indeks $i = 3$ (wskazujący na element o wartości 11) oraz $x = 4$, a następnie zamienić 11 na $11 \bmod 4 = 3$. Otrzymujemy tablicę $[3, 9, 3]$ o NWD równym 3. Da się pokazać, że lepszego wyniku osiągnąć nie można.

Wejście

```
4
6 12 18 24
```

Wyjście

```
6
```

Wyjaśnienie

Tablica ma już NWD równe 6, więc nie musimy wykonywać żadnej operacji.

Wejście

```
2
4 7
```

Wyjście

```
2
```

Wyjaśnienie

Możemy wybrać indeks $i = 2$ (element o wartości 7) oraz $x = 5$ i zamienić 7 na $7 \bmod 5 = 2$. Otrzymujemy tablicę $[4, 2]$ o NWD równym 2.

Równe podziały (N)

Limit pamięci: 256 MB

Limit czasu: 2.00 s

Krasnal Wrocek ma ciąg A składający się z N liczb. Chce podzielić go na cztery niepuste i spójne fragmenty: B, C, D oraz E .

Niech P, Q, R, S oznaczają sumy elementów odpowiednio w ciągach B, C, D, E . Wrocek jest szczęśliwszy, gdy różnica pomiędzy największą i najmniejszą z tych sum jest jak najmniejsza.

Pomóż mu i znajdź najmniejszą możliwą wartość tej różnicy.

Wejście

W pierwszym wierszu wejścia znajduje się liczba naturalna N , oznaczająca długość ciągu.

W drugim wierszu wejścia znajduje się N liczb naturalnych A_i , opisujących ciąg krasnala Wrocka.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć najmniejsza możliwa różnica pomiędzy maksimum i minimum liczb P, Q, R, S .

Ograniczenia

$4 \leq N \leq 200\,000$, $1 \leq A_i \leq 10^9$.

Przykłady

Wejście

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

Wyjście

```
2
```

Wyjaśnienie

Jeżeli podzielimy ciąg A na ciągi $B, C, D, E = (3), (2), (4), (1, 2)$, to ich sumy wynoszą kolejno $P = 3, Q = 2, R = 4, S = 3$. Największa i najmniejsza z tych wartości to 4 i 2, z różnicą 2. Można wykazać, że nie istnieje podział dający mniejszą różnicę.

Wejście

```
7
1 2 3 1000000000 4 5 6
```

Wyjście

```
999999994
```

Wejście

```
10
10 71 84 33 6 47 23 25 52 64
```

Wyjście

```
36
```

Pudełka (o)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Jasio ma N pudełek ułożonych w koło. W i -tym pudełku znajduje się A_i krasnali.

W jednym ruchu Jasio wybiera jedno pudełko — załóżmy, że ma ono numer i . Następnie dla każdego $j \in \{1, 2, \dots, N\}$ zabiera j krasnali z pudełka o numerze $i + j$. Pudełka są ułożone w koło, więc utożsamiamy pudełko o numerze k z pudełkiem o numerze $k + N$.

Zatem po takiej operacji:

- z pudełka o numerze $i + 1$ zniknie 1 krasnal,
- w pudełku $i + 2$ będą 2 krasnale mniej,
- ...
- z pudełka i zostanie zabrane N krasnali.

Jasio nie może wykonać ruchu, jeśli któreś z pudełek nie zawiera wystarczająco wielu krasnali.

Twoim zadaniem jest napisanie programu, który rozstrzygnie, czy Jasio może wykonać pewną liczbę ruchów tak, aby na końcu wszystkie pudełka były puste.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N — liczba pudełek.

W drugim wierszu wejścia znajduje się N liczb całkowitych A_1, A_2, \dots, A_N — liczby krasnali w kolejnych pudełkach.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinno się znaleźć jedno słowo:

- TAK, jeśli Jasio może opróżnić wszystkie pudełka pewnym ciągiem ruchów,
- NIE w przeciwnym przypadku.

Ograniczenia

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

Przykłady

Wejście

5
4 5 1 2 3

Wyjście

TAK

Wyjaśnienie

Jasio może wykonać jeden ruch — wybrać drugie pudełko. Wtedy z trzeciego pudełka zostanie zabrany jeden krasnal, z czwartego dwa, a ostatecznie z drugiego pudełka zostanie zabranych pięć krasnali.

Wejście

5
6 9 12 10 8

Wyjście

TAK

Wyjaśnienie

Jasio może wykonać trzy ruchy — wybrać trzecie, czwarte i piąte pudełko.

Wejście

4
1 2 3 1

Wyjście

NIE

Krasnaland (P)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Jasio został burmistrzem Krasnalandu – krainy składającej się z N wysp zamieszkałych przez krasnale. Wyspy są ponumerowane od 1 do N . Początkowo na żadnej wyspie nie ma ani lotniska, ani portu i żadnych dwóch wysp nie łączy droga. Jako nowy burmistrz Jasio chce nareszcie skomunikować cały Krasnaland ze sobą.

By osiągnąć ten cel, Jasio może:

- Wybrać liczbę i ($1 \leq i \leq N$) i wybudować na wyspie i lotnisko za X_i krasnalarów.
- Wybrać liczbę i ($1 \leq i \leq N$) i wybudować na wyspie i port za Y_i krasnalarów.
- Wybrać liczbę i ($1 \leq i \leq M$) i wybudować dwukierunkową drogę łączącą wyspy A_i i B_i za Z_i krasnalarów.

Jasio chce, żeby dla każdej pary różnych wysp U i V dało się dostać z wyspy U na wyspę V , wykonując następujące operacje dowolną liczbę razy w dowolnej kolejności:

- Jeśli na wyspach S i T znajdują się lotniska – przelecieć z wyspy S na wyspę T .
- Jeśli na wyspach S i T znajdują się porty – przepłynąć z wyspy S na wyspę T .
- Jeśli wyspy S i T są połączone drogą – przejechać z wyspy S na wyspę T .

Budżet Krasnalandu jest ograniczony, więc Jasio poprosił Cię o napisanie programu, który wyznaczy minimalny koszt (w krasnalarach) skomunikowania całej krainy.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite N i M – liczba wysp w Krasnalandzie i liczba możliwych do wybudowania dróg.

W drugim wierszu wejścia znajduje się ciąg N liczb całkowitych X_1, X_2, \dots, X_N – ceny budowy lotniska na odpowiednich wyspach w krasnalarach.

W trzecim wierszu wejścia znajduje się ciąg N liczb całkowitych Y_1, Y_2, \dots, Y_N – ceny budowy portu na odpowiednich wyspach w krasnalarach.

W każdym z kolejnych M wierszy znajdują się trzy liczby całkowite A_i, B_i, Z_i – wyspy, które może połączyć droga, oraz jej cena.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna liczba całkowita – minimalny koszt skomunikowania Krasnalandu w krasnalarach.

Ograniczenia

$2 \leq N \leq 200\,000$, $1 \leq M \leq 200\,000$, $1 \leq X_i, Y_i, Z_i \leq 10^9$, $1 \leq A_i < B_i \leq N$, $(A_i, B_i) \neq (A_j, B_j)$ dla $i \neq j$.

Przykłady

Wejście

4 2
1 20 4 7
20 2 20 3
1 3 5
1 4 6

Wyjście

16

Wyjaśnienie

Jasio wybuduje następujące obiekty, by jak najtaniej skomunikować Krasnaland:

- lotnisko na wyspie 1,
- lotnisko na wyspie 3,
- port na wyspie 2,
- port na wyspie 4,
- drogę łączącą wyspy 1 i 4.

Łącznie wyda $1 + 4 + 2 + 3 + 6 = 16$ krasnalarów.

Wejście

3 1
1 1 1
10 10 10
1 2 100

Wyjście

3

Wyjaśnienie

Jasio wybuduje lotniska na wszystkich wyspach łącznym kosztem 3 krasnalarów.

Wejście

7 8
35 29 36 88 58 15 25
99 7 49 61 67 4 57
2 3 3
2 5 36
2 6 89
1 6 24
5 7 55
1 3 71
3 4 94
5 6 21

Wyjście

160

Kr4sn4l (Q)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Próbował żeś już emacym przez poślifgryfka?

Kr4sn4l to groźny wirus komputerowy, który odtwarza na zapętleniu utwor *My jesteśmy krasnoludki*, a w dzień Barbórki wyświetla użytkownikowi pewien graf dwudzielny¹ i mówi:

Znajdź mi pieronie cykl zrychtowany z czterech wierzchołków, abo dysk idzie do fedrowanio.

Jeśli użytkownik nie spełni żądania (nie znajdzie w grafie cyklu długości 4), to *Kr4sn4l* formatuje mu dysk ce.

Grafy wyświetlane przez *Kr4sn4l*a są dość duże: mogą mieć po kilkaset tysięcy wierzchołków, więc zadanie nie jest trywialne. Wirus ma jednak pewną podatność: z dwóch zbiorów niezależnych², na które dzielą się wierzchołki tych grafów dwudzielnych, jeden ma zawsze nie więcej niż 3000 wierzchołków.

Zdarza się też, że w wyświetlonym grafie nie ma odpowiedniego cyklu. W takim wypadku nic już nie da się zrobić: *Kr4sn4l*a nie da się usunąć, bo zmienia gwarę systemu na ślōnsko godke.

Tak się składa, że nadszedł dzień Barbórki, a na Twoim ekranie wyświetlił się graf dwudzielny. Znajdź w nim dowolny cykl długości 4 lub stwierdź, że taki cykl nie istnieje.

Wejście

W pierwszym wierszu wejścia znajdują się trzy liczby naturalne: S , T i M . Graf wyświetlony przez *Kr4sn4l*a ma dokładnie $S + T$ wierzchołków i M krawędzi.

W i -tym z kolejnych M wierszy wejścia znajdują się dwie liczby całkowite U_i i V_i , oznaczające, że istnieje krawędź pomiędzy wierzchołkami o numerach U_i i V_i .

Wyjście

W jedynym wierszu wyjścia powinny znaleźć się cztery liczby całkowite, będące numerami wierzchołków cyklu długości 4 w dowolnej kolejności. Jeśli w danym grafie nie istnieje cykl długości 4, to zamiast tego w jedynym wierszu wyjścia powinna znaleźć się liczba -1.

Ograniczenia

$$2 \leq S \leq 300\,000,$$

$$2 \leq T \leq 3000,$$

$$4 \leq M \leq \min(S \cdot T, 300\,000),$$

$$1 \leq U_i \leq S,$$

$$S + 1 \leq V_i \leq S + T,$$

$$(U_i, V_i) \neq (U_j, V_j) \text{ dla } i \neq j.$$

Gwarantowane jest, że pomiędzy żadnymi dwoma wierzchołkami o numerach $1, \dots, S$, ani żadnymi dwoma wierzchołkami o numerach $S + 1, \dots, S + T$ nie ma krawędzi.

¹Graf jest dwudzielny, jeśli jego wierzchołki można pokolorować dwoma kolorami, tak, by każda krawędź łączyła wierzchołki różnych kolorów.

²Podzbiór wierzchołków w grafie nazwiemy niezależnym, jeśli każde dwa wierzchołki z tego zbioru nie są połączone żadną krawędzią. Na przykład graf jest dwudzielny wtedy i tylko wtedy, gdy jego zbiór wierzchołków można podzielić na dwa zbiory niezależne.

Przykłady

Wejście

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

Wyjście

1 2 4 5

Wyjaśnienie

W tym grafie jest tylko jeden cykl długości 4 (patrz rysunek poniżej).

Wejście

3 2 4
1 4
1 5
2 5
3 5

Wyjście

-1

Wyjaśnienie

W tym grafie nie ma żadnego cyklu długości 4.

Wejście

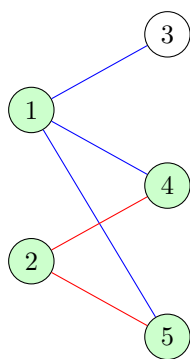
4 4 10
1 7
3 6
1 8
4 5
2 6
4 7
1 6
2 8
3 5
3 7

Wyjście

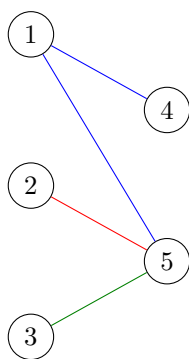
1 2 6 8

Wyjaśnienie

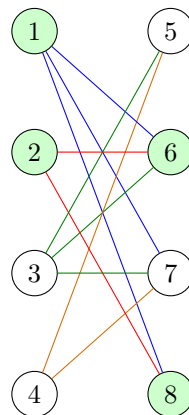
Cykle długości 4 w tym grafie to (1, 2, 6, 8), (1, 3, 6, 7) oraz (3, 4, 5, 7).



Przykład 1



Przykład 2



Przykład 3

Grządki (R)

Limit pamięci: 256 MB

Limit czasu: 5.00 s

Krasnal Ogrodnik postanowił posadzić w swoim ogródku warzywa. W tym celu przygotował N grządek, a w każdej N miejsc na pojedyncze sadzonki, w których planuje posadzić marchewki, selery i pory. Udało mu się już nawet posadzić warzywa w pierwszej grządce oraz po jednym warzywie w każdej z pozostałych grządek (najbardziej z lewej strony w każdej z nich).

Niestety, pojawiły się małe komplikacje. Ogrodnik przeczytał mianowicie, że nie jest najlepszym pomysłem, by warzywa tego samego rodzaju rosły obok siebie, ze względu na pojawiające się czasami w ogrodzie szkodniki.

Krasnal nie ma ochoty przesadzać już posadzonych warzyw, ale wymyślił sposób, żeby zapobiec problemowi w dalszej części ogrodu. Postanowił, że będzie sadił warzywa w kolejnych grządkach, a w każdej z nich kolejno od lewej do prawej. Rodzaj j -tego warzywa w i -tej grządce wybiera według następującej zasady:

- jeśli ani $(j - 1)$ -sze warzywo w tej samej grządce, ani j -te warzywo w $(i - 1)$ -szej grządce nie są **marchewką**, wybiera marchewkę;
- w przeciwnym przypadku, jeśli ani $(j - 1)$ -sze warzywo w tej samej grządce, ani j -te warzywo w $(i - 1)$ -szej grządce nie są **selerem**, wybiera selera;
- w przeciwnym przypadku wybiera **pora**.

Ogrodnik zastanawia się, czy obrana taktyka jest dobra, i chciałby wiedzieć, ile warzyw każdego rodzaju będzie ostatecznie miał posadzonych. Pomóż mu to policzyć.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N , oznaczająca zarówno liczbę grządek, jak i ich długość.

W drugim wierszu znajduje się N liczb całkowitych $A_{1,1}, A_{1,2}, \dots, A_{1,N}$, oznaczających rodzaje kolejnych warzyw w pierwszej grządce.

W kolejnych $N - 1$ wierszach znajduje się po jednej liczbie całkowitej; są to kolejno $A_{2,1}, A_{3,1}, \dots, A_{N,1}$ – rodzaje warzyw posadzonych najbardziej z lewej strony w kolejnych grządkach.

Dla wartości $A_{i,j}$ stosujemy następujące oznaczenia:

- 0 oznacza marchewkę,
- 1 oznacza selera,
- 2 oznacza pora.

Wyjście

Wypisz trzy liczby całkowite oznaczające kolejno liczbę posadzonych marchewek, selerów i porów po zastosowaniu techniki Ogrodnika.

Ograniczenia

$1 \leq N \leq 500\,000$, $A_{i,j} \in \{0, 1, 2\}$.

Przykłady

Wejście

4
1 2 0 2
0
0
0

Wyjście

7 4 5

Wyjaśnienie

Stan grządek po posadzeniu wszystkich warzyw wygląda następująco:

1 2 0 2
0 1 2 0
0 2 0 1
0 1 2 0

Wejście

1
0

Wyjście

1 0 0

Wyjaśnienie

Ogródek ma rozmiar 1×1 i zawiera tylko jedną sadzonkę marchewki – nie ma żadnego miejsca do wyliczenia.

Wejście

3
1 0 2
0
0

Wyjście

4 3 2

Wyjaśnienie

Stan grządek po wykonaniu wszystkich kroków:

1 0 2
0 1 0
0 2 1

Łącznie: 4 marchewki, 3 selery i 2 pory.

Krasnale na szychcie (s)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Hej ho! Hej ho! Do pracy by się szło!

Krasnal(udek)

Ludzie godajōm, że niy idzie chodzić na imprezy krasnali, jak sie mo szychtã. Idzie, ino trza rano wstawać. Na tym polego ôdpowiedzialność.

Krasnōl Ryszard

Krasnale wstawajōm rano o piyntyj, coby zdōnżyć do roboty na grubie. Jeszcze ćma za oknym, a ône już idōm z kilofami, bo fedrowanie samo sie niy zrobi.

Kopalnia mo N skrzyżowań pod ziymiom, i -ty krasnal je przidany do i -tego skrzyżowania, coby tam pilnowoł porzōndku, fedrowoł, niy marudził i niy łąził po cudzych chodnikach jak jaki smrōd po galotach.

Niystety, Impreza Krasnali 2 była tako grubo, że krasnale pomyliły skrzyżowania i rano każdy stoł niy tam, kaj mioł. Konkretnie, krasnale opisuje ciōng **rōżnych** liczb A_1, A_2, \dots, A_N taki, że krasnal i -ty prziszoł rano na skrzyżowanie A_i .

Niykere skrzyżowania sōm ze sobōm połōnczone, wiync jak dwa krasnale stojōm na takich placach, to mogōm sie migym pozamiyniać, niby że „jo tu stoł od rana, panie sztygarze” – naroz, coby żodne skrzyżowanie ani na chwilkã niy stoło puste. Dokłodnij, dano je lista M por liczb, co pokozywajōm, kere skrzyżowania sōm ze sobōm połōnczone. Takich zamian krasnale mogōm zrobić tela razy, wiela bydzie trza.

Terōż krasnale stojōm i rachujōm, wiela z nich da sie jeszcze przestawić na swoje richtich miejsce, zanim sztygar zacznie łązić z notesym.

Pomōż krasnalom ogarnōńć tyn bajzel i napisz program, kery wyrachuje nojwiynkszo liczba krasnali, co da sie ich poprzekłodać na swōj przypisany fyrtel.

Innymi słowy, dana jest permutacja liczb od 1 do N , oznaczona przez A_1, \dots, A_N , oraz ciąg par liczb $(X_i, Y_i)_{1 \leq i \leq M}$. Dowolnie wiele razy można wykonać operację zamiany A_{X_i} z A_{Y_i} . Należy odpowiedzieć na pytanie, ile może być maksymalnie indeksów i takich, że $A_i = i$ po pewnym ciągu takich operacji.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne N i M oznaczające liczbę krasnali i liczbę połączeń między skrzyżowaniami.

W drugim wierszu wejścia znajduje się N liczb naturalnych A_1, \dots, A_N oddzielonych pojedynczymi odstępami. Liczba A_i jest numerem skrzyżowania, przy którym początkowo stoi i -ty krasnal.

W i -tym spośród kolejnych M wierszy wejścia znajdują się dwie liczby naturalne $1 \leq X_i, Y_i \leq N$, oznaczające, że krasnale stojące na skrzyżowaniach o numerach X_i i Y_i mogą zamienić się miejscami.

Wyjście

W pierwszym (i jedynym) wierszu wyjścia powinna znaleźć się jedna liczba naturalna będąca odpowiedzią na pytanie, ile krasnali maksymalnie może stać na właściwych miejscach po wykonaniu pewnego ciągu dowolnych zamian.

Ograniczenia

$$2 \leq N \leq 100\,000,$$

$$1 \leq M \leq 100\,000,$$

$$1 \leq A_i \leq N,$$

liczby A_1, A_2, \dots, A_N są parami różne (czyli tworzą permutację),

$X_i \neq Y_i$ i jeśli $i \neq j$, to $\{X_i, Y_i\} \neq \{X_j, Y_j\}$.

Przykłady

Wejście

5 2
5 3 1 4 2
1 3
5 4

Wyjście

2

Wyjaśnienie

Po zamianie krasnali stojących na skrzyżowaniach 1 i 3 położenie krasnali będzie opisywał ciąg 1 3 5 4 2, więc dwa krasnale będą stały na swoich miejscach. Można pokazać, że nie da się lepiej ustawić krasnali.

Wejście

3 2
3 2 1
1 2
2 3

Wyjście

3

Wyjaśnienie

Po wykonaniu zamian (1, 2), (2, 3), (1, 2), otrzymujemy ciąg 1 2 3. Oczywiście jest to optymalny wynik.

Wejście

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

Wyjście

8

Wejście

5 1
1 2 3 4 5
1 5

Wyjście

5

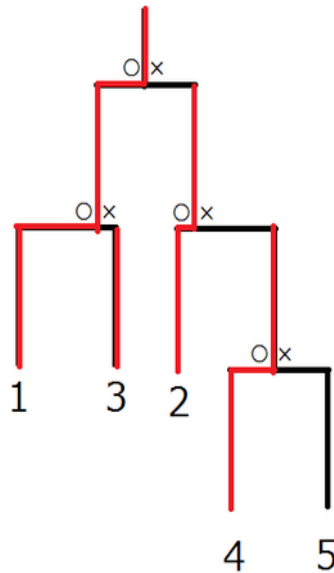
Turniej (τ)

Limit pamięci: 256 MB

Limit czasu: 2.00 s

W zawodach „Mistrzostwa Odry” wzięło udział N krasnali. Łącznie rozegrano $N - 1$ meczów w turnieju pucharowym. Z różnych powodów turniej może nie być „sprawiedliwy” dla wszystkich uczestników. Oznacza to, że liczba meczów, które trzeba rozegrać, aby zdobyć mistrzostwo, może być różna dla każdego zawodnika. Struktura turnieju jest formalnie opisana poniżej.

Po każdym meczu jest jeden zwycięzca i jeden przegrany. Ostatni pozostający w grze krasnal zostaje ogłoszony mistrzem.



Przykładowa struktura turnieju

Dla wygody, uczestnicy zostali ponumerowani od 1 do N . Krasnal o numerze 1 został mistrzem, a krasnal o numerze i (gdzie $2 \leq i \leq N$) przegrał w meczu z krasnałem o numerze A_i i tym samym odpadł z turnieju.

Formalny opis struktury turnieju jest następujący. W i -tym meczu przeciwko sobie grała jedna z poniższych par:

- dwóch z góry ustalonych uczestników,
- jeden z góry ustalony uczestnik oraz zwycięzca ustalonego wcześniej j -tego meczu ($j < i$),
- zwycięzca ustalonego wcześniej j -tego meczu oraz zwycięzca k -tego meczu ($j < i$, $k < i$, $j \neq k$).

Taka struktura jest poprawna wtedy i tylko wtedy, gdy żaden uczestnik, który już przegrał mecz, nie zagra ponownie w żadnym meczu, niezależnie od wyników pozostałych spotkań.

Zdefiniujemy głębokość turnieju jako maksymalną liczbę meczów, które musi rozegrać dowolny krasnal, aby zostać mistrzem.

Wyznacz minimalną możliwą głębokość turnieju.

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita N , oznaczająca liczbę krasnali w turnieju.

Następne $N - 1$ wierszy wejścia zawiera liczby całkowite A_2, A_3, \dots, A_N , po jednej liczbie w każdej linii. Liczba A_i oznacza numer krasnala, z którym przegrał krasnal i .

Istnieje co najmniej jeden turniej zgodny z danymi z wejścia.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć minimalna możliwa głębokość turnieju, w którym krasnal 1 zostaje mistrzem.

Ograniczenia

$2 \leq N \leq 100\,000$, $1 \leq A_i \leq N$.

Przykłady

Wejście

5
1
1
2
4

Wyjście

3

Wyjaśnienie

Następujący turniej i wyniki meczów są spójne z danymi:

- W pierwszym meczu krasnale 4 i 5 grają między sobą i wygrywa krasnal 4.
- W drugim meczu krasnale 2 i 4 grają między sobą i wygrywa krasnal 2.
- W trzecim meczu krasnal 1 i 3 grają między sobą i wygrywa krasnal 1.
- W czwartym meczu krasnal 1 i 2 grają między sobą i wygrywa krasnal 1.

Głębokość tego turnieju to 3, ponieważ krasnal 5 musiałby zagrać 3 mecze (i wszystkie wygrać), aby zostać mistrzem. Da się udowodnić, że przy danych warunkach nie ma turnieju z mniejszą głębokością, dlatego odpowiedź to 3.

Wejście

7
1
2
1
3
1
4

Wyjście

3

Wejście

4
4
4
1

Wyjście

3

Dwa tunele (u)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Krasnoludzki Zarząd Podziemi Wrocławskich podjął ostatnio decyzję o budowie dwóch tuneli pod Wrocławiem (nie wiadomo do końca po co, może jakieś metro. . .).

We wrocławskich podziemiach mieszka N krasnali, z których każdy, wedle zarządzenia, zostanie przydzielony do budowy dokładnie jednego tunelu. Dodatkowo, w zarządzeniu znajduje się warunek, że do każdego tunelu przydzielony zostanie co najmniej jeden krasnal.

Niestety, jak to zwykle bywa, prace nie mogą się tak od razu rozpocząć pierwszego dnia. Mianowicie, każdy z krasnali przedstawił zarządowi przedział dni $[P_i, K_i]$, w których jest on dostępny w tygodniu robót (warto zauważyć, że krasnoludzki tydzień ma 10^9 dni). Aby nie doszło do buntu, każdego dnia prac w danym tunelu wszystkie przydzielone do niego krasnoludki muszą stawić się w pracy.

Teraz zarząd ma nie lada zagwozdkę. Chciałby on rozdzielić krasnoludki do pracy pomiędzy dwa tunele w taki sposób, żeby sumaryczna liczba dni pracy w obu tunelach była jak największa (dni dla obu tuneli liczymy niezależnie). Pomóż zarządowi: wyznacz maksymalną sumaryczną liczbę dni pracy w obu tunelach.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N , oznaczająca liczbę krasnali we Wrocławiu.

W kolejnych N wierszach znajdują się po dwie liczby całkowite P_i oraz K_i , oznaczające, że krasnoludek o numerze i może pracować w dniach od P_i do K_i włącznie.

Wyjście

W pierwszym i jedynym wierszu wyjścia powinna się znaleźć jedna liczba całkowita, oznaczająca maksymalną sumaryczną liczbę dni, w których mogą odbywać się prace w tunelach.

Ograniczenia

$$2 \leq N \leq 100\,000, 1 \leq P_i \leq K_i \leq 10^9.$$

Przykłady

Wejście

4
4 7
1 4
5 8
2 5

Wyjście

6

Wyjaśnienie

W tym przypadku optymalny podział to pierwszy i trzeci krasnal w jednym tunelu, a drugi i czwarty w drugim tunelu. Wówczas w pierwszym tunelu prace będą trwały w dniach od 5 do 7 (3 dni), a w drugim tunelu w dniach od 2 do 4 (3 dni). Sumarycznie 6 dni.

Wejście

4
1 20
2 19
3 18
4 17

Wyjście

34

Wejście

10
457835016 996058008
456475528 529149798
455108441 512701454
455817105 523506955
457368248 814532746
455073228 459494089
456651538 774276744
457667152 974637457
457293701 800549465
456580262 636471526

Wyjście

540049931

Wejście

3
2 10
3 5
7 9

Wyjście

9

Wyjaśnienie

Pierwszy krasnal został przydzielony do pierwszego tunelu (9 dni pracy), a drugi i trzeci do drugiego tunelu (0 dni pracy). Czasami warto po prostu odpuścić sobie jedną linię metra. . .

Wieniec z kwiatów (v)

Limit pamięci: 256 MB

Limit czasu: 2.00 s

Krasnal Xorek jest romantykiem. Chciałby wyznać swoje uczucia krasnalce Nandy, wręczając jej piękny wieniec z polnych kwiatów. Wieniec, który przygotował, składa się z N kwiatów. Kwiat na pozycji i ma A_i płatków. Wieniec ma kształt okręgu, dla $1 \leq i < N$ kwiat na pozycji i sąsiaduje z kwiatem na pozycji $i + 1$, a dodatkowo kwiat na pozycji 1 sąsiaduje z kwiatem na pozycji N .

Gdy Xorek zobaczył gotowy wieniec, uznał, że kompozycja nie spełnia jego oczekiwań i postanowił ją zmienić. Stwierdził, że Nandy dużo bardziej spodobałby się wieniec, w którym na pozycjach $1, 2, \dots, N$ kwiaty mają odpowiednio B_1, B_2, \dots, B_N płatków.

Spotkanie z Nandy ma rozpocząć się już niebawem. Xorek nie ma już czasu sporządzić nowego wieńca od zera. Może natomiast przekształcić wieniec A_1, A_2, \dots, A_N w wieniec B_1, B_2, \dots, B_N , wykonując każdą z poniższych operacji **dokładnie raz**:

- wybrać liczbę całkowitą K ($0 \leq K < N$) i obrócić wieniec o K pozycji w lewo. Po obrocie kwiat z pozycji i znajdzie się na pozycji $((i - K - 1) \bmod N) + 1$ — innymi słowy, ciąg A_1, A_2, \dots, A_N stanie się $A_{K+1}, A_{K+2}, \dots, A_N, A_1, \dots, A_K$,
- wybrać liczbę całkowitą X ($0 \leq X < 2^{30}$) i rzucić pradawne krasnoludzkie zaklęcie *Xorrium*, które zmieni liczbę płatków każdego kwiatu z A_i na $A_i \oplus X$.

Czy jesteś w stanie pomóc Xorkowi wyznaczyć wszystkie pary liczb (K, X) , które pozwolą przekształcić wieniec A w wieniec B ?

Wejście

W pierwszym wierszu wejścia znajduje się jedna dodatnia liczba całkowita N oznaczająca długość oryginalnego i docelowego wieńca.

W kolejnym wierszu wejścia znajduje się ciąg N nieujemnych liczb całkowitych A_i oznaczających liczbę płatków na i -tym kwiecie wieńca początkowego.

W ostatnim wierszu wejścia znajduje się ciąg N nieujemnych liczb całkowitych B_i oznaczających liczbę płatków i -tego kwiatu w docelowym wieńcu.

Wyjście

Niech M oznacza liczbę par (K, X) pozwalających przekształcić wieniec początkowy w docelowy. Na wyjściu powinno znajdować się M wierszy. Jeśli takich par nie ma, wyjście jest puste.

W każdym z nich powinny znajdować się dwie liczby całkowite K_i oraz X_i , opisujące jedną poprawną parę. Wypisz pary posortowane rosnąco po K , a w przypadku remisu po X .

Ograniczenia

$$1 \leq N \leq 200\,000, 0 \leq A_i, B_i < 2^{30}.$$

Przykłady

Wejście

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

Wyjście

```
1 3
```

Wyjaśnienie

Po obrocie wieńca o jedną pozycję w lewo ciąg ilości płatków kwiatów jest następujący: $(2, 1, 0)$. Po rzuceniu zaklęcia *Xorrium* ciąg zmieni się na $(2 \oplus 3, 1 \oplus 3, 0 \oplus 3)$, co jest równe $(1, 2, 3)$. Można pokazać, że nie istnieją inne pary (K, X) przekształcające $(0, 2, 1)$ w $(1, 2, 3)$.

Wejście

5
0 0 0 0 0
2 2 2 2 2

Wyjście

0 2
1 2
2 2
3 2
4 2

Wejście

6
0 1 3 7 6 4
1 5 4 6 2 3

Wyjście

2 2
5 5

Wejście

2
1 2
0 0

Wyjście

Trzy Krasnale (w)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Jasio dostał na urodziny zabawkę składającą się z N pól ułożonych w rzędzie. Pola są ponumerowane od 1 do N od lewej do prawej.

Wszystkie pola są początkowo puste. Jasio może wykonać dowolną liczbę operacji następujących typów, w dowolnej kolejności:

- Wybrać trzy **sąsiednie** puste pola i położyć na każdym z nich po jednym krasnalu.
- Wybrać trzy **sąsiednie** pola, z których każde zawiera po krasnalu, po czym zdjąć z tych pól te trzy krasnale.

Po wykonaniu wszystkich operacji Jasio dostaje A_i punktów, jeśli na i -tym polu stoi krasnal. Wynik Jasia to sumaryczna liczba punktów za pola z krasnalami.

Twoim zadaniem jest napisanie programu, który znajdzie maksymalną liczbę punktów jaką Jasio jest w stanie uzyskać.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N – liczba pól, z których składa się zabawka Jasia.

W każdym z kolejnych N wierszy znajduje się jedna liczba całkowita A_i – liczba punktów, jaką Jasio otrzyma, jeśli finalnie na i -tym polu będzie stał krasnal.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna liczba całkowita – maksymalny wynik, jaki Jasio może osiągnąć.

Ograniczenia

$$3 \leq N \leq 500, -100 \leq A_i \leq 100.$$

Przykłady

Wejście

4
1
2
3
4

Wyjście

9

Wyjaśnienie

Oznaczmy pola z krasnalem jako o, a pola bez krasnala jako . (kropka).

Jedna z optymalnych sekwencji ruchów to:

.... → .ooo

Wynik Jasia wynosi wtedy $2 + 3 + 4 = 9$.

Wejście

6
3
-2
-1
0
-1
4

Wyjście

6

Wyjaśnienie

Jedna z optymalnych sekwencji ruchów to:

..... → ooo... → oooooo → o...oo

Wynik Jasia wynosi wtedy

$$3 + (-1) + 4 = 6.$$

Wejście

10
-84
-60
-41
-100
8
-8
-52
-62
-61
-76

Wyjście

0

Wyjaśnienie

Jasio może nie wykonać żadnego ruchu.

Sieć (x)

Limit pamięci: 512 MB

Limit czasu: 2.00 s

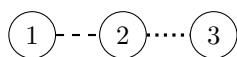
Nie da się ukryć, że technologia rozwija się w zatrważającym tempie, a nawet krasnoludki idą z duchem czasu. Z tego też powodu postanowiły one połączyć swoje N domków siecią przewodową, używając do tego $N - 1$ połączeń.

Sieć w wiosce powstała w szczególny sposób: krasnoludki na początku wybrały pewną liczbę $k \in [1, N - 1]$, po czym połączyły pewien domek o numerze z przedziału $[1, k]$ z pewnym domkiem z przedziału $[k + 1, N]$. Następnie, stosując tę samą zasadę, połączyły ze sobą domki od 1 do k oraz w ten sam sposób domki od $k + 1$ do N .

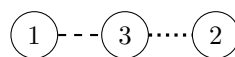
Teraz krasnoludkom zależy na tym, żeby odtworzyć połączenia w sieci. Niestety, jest to bardzo trudne zadanie, gdyż jedyną rzeczą zapisaną w dokumentacji jest schemat, według którego powstawała sieć. Schemat ma następującą postać:

- wyrażenie 1 oznacza domek (o numerze równym liczbie znaków 1, które dotychczas wystąpiły w schemacie, więc kolejne znaki 1 oznaczają domki o kolejnych numerach $1, 2, \dots, N$),
- wyrażenie $(E_1 E_2)$, gdzie E_1 i E_2 są poprawnymi wyrażeniami, oznacza stworzenie sieci dla kolejnych domków według schematu E_1 , następnie tak samo dla E_2 , po czym połączenie pewnego domku z E_1 z pewnym domkiem z E_2 .

Taki schemat ogranicza liczbę możliwych zestawów połączeń, ale nie jest jednoznaczny. Przykładowo, wyrażenie $(1(11))$ może opisywać dwa różne zestawy połączeń, pokazane poniżej.



Wariant 1



Wariant 2

Wewnętrzne wyrażenie (11) tworzy podsieć z dwóch domków o numerach 2 i 3 oraz łączy je krawędzią, na rysunkach przedstawiono ją linią **kropkowaną**. Zewnętrzne wyrażenie $(1(11))$ dokłada domek 1 i łączy go z wybranym domkiem podsieci $\{2, 3\}$, co przedstawiono linią **przerywaną** — w *Wariancie 1* wybrano domek 2, a w *Wariancie 2* domek 3. W ten sposób z wyrażenia $(1(11))$ można otrzymać dwa różne zestawy połączeń.

Jak łatwo zauważyć, liczba możliwych zestawów połączeń wynikających z jednego schematu może być bardzo duża. Szczęśliwie okazało się, że jeden z krasnoludków zauważył w dokumentacji **drugi** schemat, stworzony według tych samych zasad, opisujący tę samą sieć. Niestety, nawet dwa różne schematy mogą okazać się niewystarczające do jednoznacznego odtworzenia sieci.

Czy jesteś w stanie pomóc krasnoludkom policzyć, ile jest możliwych zestawów połączeń spełniających oba schematy?

Wejście

Na wejściu znajdują się dwa wiersze, a każdy z nich zawiera jeden opis schematu powstawania sieci, według podanych wyżej zasad.

Oba schematy są poprawne i mają tę samą długość, którą oznaczmy przez L .

Wyjście

W pierwszym (jedynym) wierszu wyjścia wypisz jedną liczbę całkowitą, oznaczającą liczbę możliwych zestawów połączeń, które można uzyskać z obu schematów. Ponieważ krasnoludki gubią się w dużych liczbach, wynik wypisz modulo 998 244 353.

Ograniczenia

$1 \leq L \leq 700\,000$.

Przykłady

Wejście

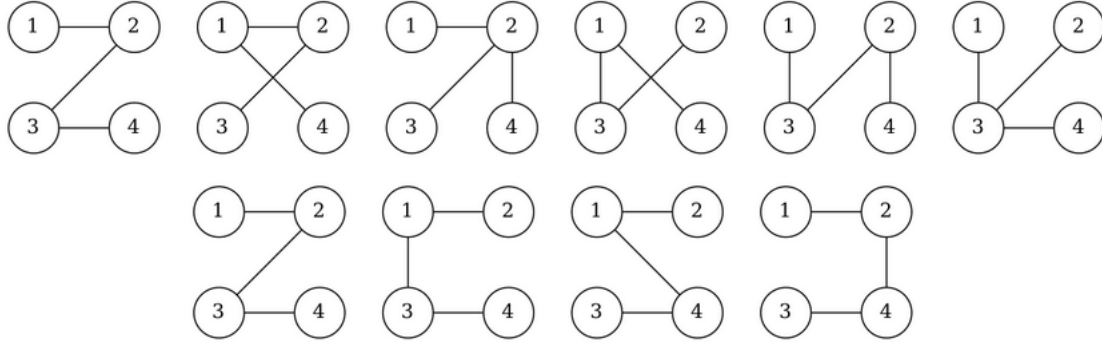
((1(11))1)
((11)(11))

Wyjście

1

Wyjaśnienie

Pierwszy wiersz poniższego obrazka pokazuje, jakie zestawy połączeń mogą zostać wygenerowane za pomocą pierwszego schematu, a drugi wiersz — za pomocą drugiego schematu. Jedyny wspólny zestaw to pierwszy zestaw w obu wierszach, dlatego odpowiedź to 1.



Wejście

(1(11))
(1(11))

Wyjście

2

Wejście

((11)(11))((11)1)
((1(11))(1(1(11))))

Wyjście

3

Wejście

((11)((1(11))1)((11)1))
(1((11)((11)(11)))(11))

Wyjście

4

Gra w Grę I (γ)

Limit pamięci: 1024 MB

Limit czasu: 2.00 s

Krasnale Algoś i Buguś postanowili rozegrać pojedynek w Krasnalowe Reversi. Zamiast zwykłej planszy wybrali ukorzenione drzewo.

Dane jest ukorzenione drzewo o N wierzchołkach ponumerowanych od 1 do N . Korzeniem drzewa jest wierzchołek 1. Dla każdego wierzchołka i takiego, że $2 \leq i \leq N$, jego ojcem jest wierzchołek P_i .

Na początku na żadnym wierzchołku nie leży żeton. Każdy żeton ma dwie strony: białą i czarną.

Krasnale wykonują ruchy na przemian. Pierwszy ruch należy do Algosa. W swoim ruchu Algoś kładzie żeton białą stroną do góry, a Buguś — czarną.

W jednym ruchu można położyć żeton na wierzchołku u tylko wtedy, gdy spełnione są oba poniższe warunki:

- na wierzchołku u nie leży jeszcze żaden żeton,
- na wszystkich potomkach wierzchołka u leżą już żetony.

Potomkami wierzchołka u są wszystkie wierzchołki znajdujące się niżej w jego poddrzewie. Wierzchołek u nie jest swoim własnym potomkiem.

Po położeniu żetonu na wierzchołku u wszystkie żetony leżące na jego potomkach zostają odwrócone na drugą stronę: z białej na czarną albo z czarnej na białą. Żeton właśnie położony na wierzchołku u nie zostaje odwrócony.

Gra kończy się, gdy na każdym wierzchołku leży żeton. Wynikiem Algosa jest liczba żetonów, które po zakończeniu gry leżą białą stroną do góry.

Algoś chce zmaksymalizować swój wynik, a Buguś chce go zminimalizować. Wyznacz wynik Algosa przy optymalnej grze obu krasnali.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N , oznaczająca liczbę wierzchołków drzewa.

W drugim wierszu znajduje się $N - 1$ liczb naturalnych P_2, P_3, \dots, P_N . Liczba P_i oznacza numer ojca wierzchołka i .

Wyjście

W jednym wierszu wyjścia należy wypisać jedną liczbę całkowitą: wynik Algosa przy optymalnej grze obu krasnali.

Ograniczenia

$2 \leq N \leq 200\,000$, $1 \leq P_i < i$ dla każdego $2 \leq i \leq N$.

Przykłady

Wejście

4
1 1 2

Wyjście

2

Wyjaśnienie

Na początku gry żeton można położyć tylko na wierzchołkach 3 oraz 4. Jedną z możliwych rozgrywek wygląda następująco:

- Algoś kładzie żeton białą stroną do góry na wierzchołku 4.
- Buguś kładzie żeton czarną stroną do góry na wierzchołku 2 i odwraca żeton na wierzchołku 4 na czarną stronę.
- Algoś kładzie żeton białą stroną do góry na wierzchołku 3.
- Buguś kładzie żeton czarną stroną do góry na wierzchołku 1 i odwraca żetony na wierzchołkach 2, 3 oraz 4.

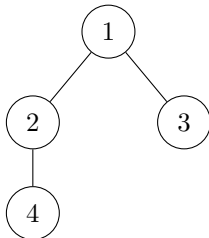
Na koniec tej rozgrywki żetony na wierzchołkach 1, 2, 3, 4 leżą odpowiednio czarną, białą, czarną i białą stroną do góry. Jest to jedna z optymalnych rozgrywek dla obu krasnali, więc odpowiedź wynosi 2. Jest to najlepszy wynik, jaki może uzyskać Algoś przy optymalnej grze Bugusia.

Wejście

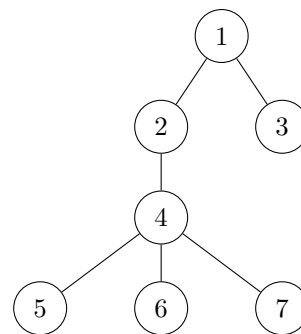
7
1 1 2 4 4 4

Wyjście

5



Przykład 1



Przykład 2

Gra w Grę II (z)

Limit pamięci: 256 MB

Limit czasu: 5.00 s

Krasnale Algoś i Buguś chcą rozstrzygnąć, który z nich jest lepszym strategiem.

Rozgrywka odbywa się na skierowanym grafie acyklicznym o N wierzchołkach ponumerowanych od 1 do N oraz M krawędziach.

Na początku gry na wierzchołkach 1 i 2 leży po jednym żetonie. Następnie krasnale wykonują ruchy na zmianę, zaczynając od Algosaia.

W jednym ruchu gracz wybiera jeden z dwóch żetonów. Jeśli wybrany żeton znajduje się aktualnie na wierzchołku u , gracz musi przesunąć go na dowolny wierzchołek v , dla którego istnieje skierowana krawędź z u do v . Nawet jeśli oba żetony znajdują się na tym samym wierzchołku, w jednym ruchu przesuwany jest dokładnie jeden z nich.

Gracz, który nie może wykonać żadnego ruchu, przegrywa.

Algoś i Buguś wybrali już graf, na którym mogliby rozegrać pojedynek. Zastanawiają się jednak nad wszystkimi możliwymi podzbiórami jego krawędzi. Każdy z 2^M podzbiorów krawędzi wyznacza osobną grę: rozgrywaną na grafie zawierającym wszystkie N wierzchołków oraz tylko krawędzie należące do tego podzbioru. Pusty podzbiór krawędzi również jest brany pod uwagę.

Pomóż krasnalom obliczyć, dla ilu podzbiorów krawędzi Algoś ma strategię wygrywającą. Wynik należy podać modulo $10^9 + 7$.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite N i M , oznaczające liczbę wierzchołków oraz liczbę krawędzi grafu.

W kolejnych M wierszach znajdują się opisy krawędzi. Każdy z nich zawiera dwie liczby całkowite A_i i B_i , oznaczające skierowaną krawędź z wierzchołka A_i do wierzchołka B_i .

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać liczbę podzbiorów krawędzi, dla których Algoś ma strategię wygrywającą, modulo $10^9 + 7$.

Ograniczenia

$2 \leq N \leq 15$, $1 \leq M \leq \frac{N(N-1)}{2}$, $1 \leq A_i < B_i \leq N$,
wszystkie pary (A_i, B_i) są różne.

Przykłady

Wejście

2 1

1 2

Wyjście

1

Wyjaśnienie

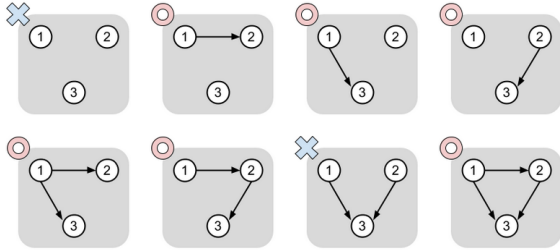


Wejście

3 3
 1 2
 1 3
 2 3

Wyjście

6

Wyjaśnienie**Wejście**

4 2
 1 3
 2 4

Wyjście

2

Wejście

5 10
 2 4
 3 4
 2 5
 2 3
 1 2
 3 5
 1 3
 1 5
 4 5
 1 4

Wyjście

816