

Buty (buty)

Memory limit: 128 MB Time limit: 2.00 s

Ojciec Wirgiliusz uczył dzieci swoje, a miał ich wszystkich 123456. Pewnego razu zabrał niektóre swoje dzieci, dokładnie N spośród nich, żeby kupić im buty na WF.

Na półce w sklepie znajduje się M par butów sportowych. Każda para ma przypisany rozmiar, oraz cenę (wyrażoną w Bitylingach). Każde dziecko musi mieć kupione buty w dokładnie takim samym rozmiarze jak rozmiar ich bieżących butów.

Wirgiliusz zastanawia się teraz, ile musi wziąć ze sobą pieniędzy do sklepu, żeby każde dziecko w sklepie mogło cieszyć się własną parą nowych butów. Może się okazać, że sklep nie ma wystarczającej liczby butów na stanie, wtedy Wirgiliusz będzie musiał poszukać innego sklepu.

Napisz program, który sprawdzi, czy możliwe jest kupienie butów dla dzieci w sklepie, a jeżeli tak, to jaki jest najmniejszy możliwy łączny koszt zakupów.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne N i M , oznaczające odpowiednio liczbę dzieci w sklepie oraz liczbę par butów na sprzedaż.

W drugim wierszu wejścia znajduje się N liczb naturalnych s_i oznaczających rozmiary butów dzieci, które wybrały się z Wirgiliuszem do sklepu.

W kolejnych M wierszach znajdują się opisy par butów na sprzedaż. W i -tym z nich umieszczono dwie liczby naturalne r_i i c_i , oznaczające odpowiednio rozmiar i cenę i -tej pary butów do sprzedania.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna liczba całkowita oznaczająca minimalną liczbę bitylingów potrzebną do zakupu butów. Jeżeli nie da się kupić butów dla każdego, należy zamiast tego wypisać jedno słowo NIE.

Ograniczenia

$1 \leq N \leq 123\,456$, $1 \leq M \leq 200\,000$, $20 \leq s_i, r_i \leq 50$, $1 \leq c_i \leq 500$.

Przykład

Input	Output	Explanation
3 7	418	Można kupić dwie pary butów o rozmiarze 36 za 129 i 139 bitylingów oraz jedną parę w rozmiarze 41 za 150 bitylingów.
36 41 36		
36 139		
38 100		
41 150		
36 199		
38 100		
36 129		
40 279		

Input	Output	Explanation
5 12	NIE	Niestety, w tym przypadku nie jest możliwe zadowolenie wszystkich dzieci.
37 41 42 42 42		
36 199		
37 199		
37 199		
40 219		
41 219		
41 219		
41 219		
41 219		
41 219		
41 219		
42 219		
42 219		

Gra terenowa (gra-terenowa)

Memory limit: 128 MB

Time limit: 1.00 s

Jasio zabrał się ostatnio za pisanie gry komputerowej – *Symulator Gry Terenowej*. Jednym z problemów jaki musi rozwiązać jest generowanie planszy dla graczy.

Każda plansza reprezentuje kawałek lasu o długości N i szerokości M metrów. Każdy metr kwadratowy lasu może być albo zarośnięty przez drzewa i krzewy (oznaczamy to pole za pomocą symbolu #), albo kawałkiem polany (używamy symbolu 0).

Na początku rozgrywki, gracz musi wybrać lokalizację swojej *bazy*, gdzie będzie się znajdowała flaga, którą będzie chciał wykraść przeciwnik. Z oczywistych względów, najlepszym kandydatem na bazę jest pole, które jest polaną, oraz dla którego wszystkie pola, które mają z nim chociaż jeden punkt wspólny są zarośnięte.

Jasio ma przygotowaną pewną liczbę *szablonów map*, dla których niektóre pola mają już przypisane wartości, niektóre zaś jeszcze nie (te komórki oznaczmy symbolem ?). Wiadomo, że dla każdej komórki ? ma ona **co najwyżej dwóch** sąsiadów o wartości ?.

Jasio zastanawia się teraz dla danego szablonu, ile maksymalnie pól będzie spełniało wymogi dobrej bazy, jeżeli w optymalny sposób przypisze wartości wszystkim komórkom (takie plansze będą ciekawsze dla graczy).

Napisz program, który odpowie na pytanie Jasia i niniejszym przyspieszy publikację gry Jasia.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne N i M oznaczające wymiary kawałka lasu, na którym będzie toczyła się rozgrywka.

W kolejnych n wierszach znajduje się po m symboli 0, # lub ?, o podanym wyżej znaczeniu.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć maksymalna liczba pól spełniających kryteria dobrej bazy po przypisaniu wartości komórkom ?.

Ograniczenia

$1 \leq N, M \leq 1\,000$.

Przykład

Input	Output
5 6	3
#####?	
#?0##?	
##?###	
###0?	
#00##0	

Input	Output
1 1	1
?	

Klapy (klapy)

Memory limit: 128 MB

Time limit: 2.00 s

Jasio wybrał się ze znajomymi na majówkę na Mazury. Ekipa podjęła decyzję żeby na wyprawę pojechać jego samochodem. Niestety, w trakcie podróży okazało się, że pogoda jest dość chłodna i Jasio rozważa teraz włączenie klimatyzacji aby podnieść temperaturę w pojeździe.

Układ klimatyzacji jest dość niewygodny w obsłudze. Składa się z N nawiewów ustawionych obok siebie. i -ty z nich ma moc M_i , czyli zwiększa temperaturę w pojeździe o dokładnie M_i stopni Bajcjusza. Czasami moc nawiewu może być ujemna i wtedy zmniejsza on temperaturę w samochodzie.

Na szczęście Jasio ma ze sobą dwie klapy. Dla każdej z nich może podjąć decyzję aby jej użyć lub nie. Użycie klapy polega na nałożeniu jej na wybrane przez siebie **trzy kolejne** nawiewy. Przykryte przez klapy nawiewy przestają zmieniać temperaturę.

Klapy mogą się nakładać na siebie, ale jeżeli już zostaną użyte, to muszą w całości przykryć jakieś trzy sąsiednie nawiewy (nie jest na przykład możliwe przykrycie tylko dwóch pierwszych nawiewów lub ich częściowe przykrywanie).

Jasio zastanawia się o ile najwięcej stopni Bajcjusza jest w stanie podnieść temperaturę (jeżeli w ogóle opłaca mu się uruchomić klimatyzację).

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N oznaczająca liczbę nawiewów w samochodzie Jasia.

W drugim wierszu wejścia znajduje się N liczb całkowitych M_i oznaczających moce kolejnych nawiewów w samochodzie Jasia.

Wyjście

W pierwszym wierszu wyjścia powinna się znaleźć jedna liczba całkowita oznaczająca największą możliwą liczbę stopni Bajcjusza, o które może wzrosnąć temperatura w samochodzie Jasia.

Ograniczenia

$$3 \leq N \leq 200\,000, \quad -1\,000\,000 \leq M_i \leq 1\,000\,000.$$

Przykład

Input 6 -2 7 -1 -13 2 -7	Output 5	Explanation Jasio może użyć obu klap przykrywając sumarycznie cztery nawiewy.
Input 6 -3 0 1000 -3 -6 2	Output 997	Explanation Wystarczy użyć jednej klapy do osiągnięcia optymalnego wyniku.
Input 8 -1 -1 -1 -1 -1 -1 -1 -1	Output 0	Explanation W tym przypadku nie warto włączać klimatyzacji.
Input 3 2 0 23	Output 25	Explanation Jasiowi opłaca się włączyć klimę i nie używać żadnej z klap.

Lider kontratakuje (lider-2)

Memory limit: 512 MB

Time limit: 2.00 s

Jasio lubi sobie utrudniać życie. . . Ostatnio udało mu się zrobić zadanie *Lider*, w którym dla danego ciągu długości N trzeba było wskazać, czy posiada on lidera, czyli element występujący więcej niż $\frac{N}{2}$ razy.

Uznał to zadanie za zbyt proste, by móc wystarczająco cieszyć się z jego rozwiązania. Wymyślił więc trudniejszy problem, który jednak okazał się być ponad jego siły.

Twoim zadaniem jest napisać program, który dla ciągu liczb policzy ile jego spójnych przedziałów (fragmentów od i -tej do j -tej pozycji dla $1 \leq i \leq j \leq N$) zawiera jakiegoś lidera.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N – długość ciągu Jasia. W drugim wierszu znajduje się N liczb naturalnych A_i będących kolejnymi wyrazami ciągu.

Wyjście

W pierwszym wierszu wyjścia powinna się znaleźć jedna liczba całkowita, oznaczająca liczbę spójnych przedziałów wejściowego ciągu, które zawierają jakiegoś lidera.

Ograniczenia

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

Przykład

Input	Output	Explanation
3 1 2 3	3	Tylko przedziały jednoelementowe posiadają lidera.
5 4 4 2 2 4	11	

Odcykanie (odcykanie)

Memory limit: 128 MB

Time limit: 1.00 s

Jasio wraz z przyjaciółmi w trakcie przerwy między zajęciami szkolnymi wpadli na pomysł wypisania na tablicy swoich ulubionych słów.

Zabawa trwała w najlepsze, ale gdy przyszła kolej na Jasia, przypomniał on sobie, że jego rówieśnicy bardzo nie lubią *nudnych* słów. Słowo s uznajemy za nudne, jeżeli istnieje jakieś inne słowo w mające długość będącą dzielnikiem $|s|$ oraz s da się otrzymać przez konkatenację odpowiedniej liczby wystąpień słowa w (innymi słowy: w jest pierwiastkiem słowa s).

Jasio ma swoje ulubione słowo, ale bardzo nie chciałby zdenerwować swoich znajomych, więc jeżeli to konieczne postanowił zamienić jakieś litery w tym słowie, by nie było nudne. Oczywiście Jasio może zamienić literę tylko na inną literę z alfabetu łacińskiego.

Napisz program, który wyznaczy minimalną liczbę liter, którą musi podmienić.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N oznaczająca długość słowa s .

W drugim wierszu znajduje się napis s złożony z małych liter alfabetu łacińskiego.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna liczba całkowita oznaczająca minimalną liczbę zmian, jakie musi wykonać Jasio w swoim ulubionym słowie.

Ograniczenia

$2 \leq N \leq 500\,000$.

Przykład

Input	Output	Explanation
4 mama	1	Jasio może zamienić swoje słowo na przykład na <code>maaa</code> albo <code>mmaa</code> .
5 kopor	0	

Sortowanie przez xorowanie (sor-xor)

Memory limit: 128 MB Time limit: 1.00 s

Jasio poznał ostatnio wiele różnych ciekawych metod na sortowanie ciągu liczb takich jak sortowanie przez scalanie, sortowanie bąbelkowe albo sortowanie przez zliczanie.

Zainspirowany tą tematyką postanowił wymyślić swój własny algorytm sortujący, który ma jednak parę wad i ograniczeń.

Ciągi, których sortowanie rozważa Jasio składają z liczb całkowitych o wartościach A_i z przedziału $[0, 2^K)$ zapisanych na dokładnie k bitach. Algorytm Jasia pozwala na wykonywanie dowolnie wiele razy następującej operacji:

- Wybierz niepusty spójny fragment ciągu od l -tej do r -tej pozycji włącznie i każdej liczbie A_i ($l \leq i \leq r$) zamień wszystkie bity na przeciwne (zamień A_i na $A_i \oplus (2^K - 1)$).

Twoim zadaniem jest napisać program, który obliczy jaka jest minimalna liczba operacji konieczna do posortowania ciągu niemalejąco, albo wypisze, że nie jest to możliwe.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite N i K oznaczające długość ciągu oraz liczbę bitów, za pomocą których da się zapisać wszystkie elementy ciągu.

W drugim wierszu wejścia znajduje się N liczb całkowitych pooddzielanych pojedynczymi odstępami oznaczających kolejne elementy ciągu A_i .

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć minimalna liczba operacji potrzebna do tego, żeby ciąg stał się niemalejący, albo jedna liczba -1 jeżeli posortowanie nie jest możliwe.

Ograniczenia

$1 \leq N \leq 500\,000$, $1 \leq K \leq 20$.

Przykład

Input	Output	Explanation
4 3 5 2 4 1	2	Możemy wybrać przedziały pozycji $[1, 3]$ oraz $[2, 4]$ otrzymując na końcu ciąg $(2, 2, 4, 6)$.

Input	Output
7 2 0 1 0 1 0 1 0	-1

Stan splątany (stan-splątany)

Memory limit: 256 MB

Time limit: 2.00 s

Mechanika kwantowa była działem nauki, który od zawsze fascynował Jasia. Teraz, gdy ma już za sobą maturę, ma więcej czasu na eksperymenty, więc postanowił zbudować układ, w którym mógłby badać stan *cząstek splątanych*. Układ składa się z N komór, w których każda zawiera dokładnie jedną cząstkę. Komory połączone są ze sobą dokładnie $N - 1$ tunelami kwantowymi, w taki sposób że każda para cząsteczek może się skomunikować za pomocą (być może niebezpośredniego) połączenia tunelami na dokładnie jeden sposób.

Jasio poczynił już pewne obserwacje, ale teraz chciałby je potwierdzić doświadczalnie. Będzie do tego jednak najpierw potrzebował *zneutralizować* układ. Niektóre cząstki w układzie komór Jasia są aktywne, a niektóre nie. Jego zadaniem jest wprowadzić wszystkie cząstki w stan nieaktywny, tak by mógł kontynuować eksperymenty w warunkach sterylnych.

Do neutralizacji skorzysta z pewnej własności kwantowej – niektóre cząstki podlegają *splątaniu*. Na potrzeby tego zadania możemy przyjąć, że cząstki w danych dwóch komorach a i b mogą ulec splątaniu, jeżeli odległość między nimi (mierzona w liczbie tuneli na ścieżce między nimi) jest nie większa niż K . Jeżeli jakieś dwie cząstki są ze sobą splątane, to Jasio może skorzystać z takiej więzi i dokonać zamiany stanów tych cząstek. Każda z dwóch cząstek jeżeli była aktywna, to po takiej operacji będzie nieaktywna, oraz na odwrót – jeżeli była nieaktywna, to się uaktywni.

Napisz dla Jasia program, który sprawdzi, czy możliwa jest neutralizacja układu, a jeżeli tak, to jaka jest najmniejsza liczba operacji zamian, by osiągnąć zamierzony efekt.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite N i K oznaczające odpowiednio liczbę komór w układzie kwantowym Jasia, oraz maksymalną odległość między splątanymi cząstkami.

W drugim wierszu wejścia znajduje się N liczb całkowitych. i -ta z nich oznacza stan cząstki w i -tej komorze na początku eksperymentu, gdzie 1 oznacza cząstkę aktywną, a 0 cząstkę nieaktywną.

W i -tym z kolejnych $N - 1$ wierszy znajdują się dwie liczby naturalne a_i oraz b_i oznaczające bezpośredni tunel między komorami o tych numerach.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna liczba całkowita oznaczająca minimalną liczbę operacji koniecznych do zamiany stanu wszystkich cząsteczek na nieaktywne, albo napis NIE jeżeli nie jest to możliwe.

Ograniczenia

$2 \leq N \leq 200\,000$, $1 \leq K \leq N$.

Przykład

Input	Output	Explanation
7 2	3	Układ może zostać zneutralizowany za pomocą operacji na parach (2, 6), (1, 2) oraz (7, 4).
1 0 0 1 0 1 1		
1 2		
2 3		
3 4		
5 2		
6 5		
7 3		

Input

4 2
1 1 0 1
1 2
1 3
1 4

Output

NIE

Trójkąty i kwadraty (tro-kwa)

Memory limit: 256 MB

Time limit: 5.00 s

Jasio dostał na egzaminie zadanie narysowania grafu nieskierowanego o dokładnie N wierzchołkach oraz M krawędziach (bez krawędzi wielokrotnych i pętelek). Niestety przez roztargnienie oraz to, że po głowie chodzi mu wciąż pewna piosenka, zrozumiał opacznie, że jego zadaniem jest narysowanie grafu, na dokładnie N wierzchołkach, który ma dokładnie M trójkątów i kwadratów.

Trójkątem nazywamy trzy wierzchołki, takie że są połączone ze sobą krawędzią, zaś kwadratem nazwiemy cztery wierzchołki a, b, c, d takie, że mamy krawędzie $a \leftrightarrow b, b \leftrightarrow c, c \leftrightarrow d$ oraz $d \leftrightarrow a$, przy czym dwa kwadraty są różne, jeżeli posiadają różne zbiory krawędzi. Dla przykładu: czwórki $[a, b, c, d]$, $[b, c, d, a]$ oraz $[d, c, b, a]$ są sobie parami równe, ale $[a, b, c, d]$ różni się od $[a, c, b, d]$.

Twoim zadaniem mogłoby być napisanie programu, który powiedziałby Jasiowi, że źle zrozumiał treść, ale to byłoby zbyt proste. Domyślasz się już pewnie, że aby otrzymać punkty za to zadanie, trzeba rozwiązać zadanie wymyślane przez Jasia. Napisz program, który poprawnie skonstruuje wymagany graf, albo wypisze, że nie jest to możliwe.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita T oznaczająca liczbę przypadków testowych.

W i -tym z kolejnych T wierszy znajdują się po dwie liczby całkowite N_i i M_i , oznaczające odpowiednio liczbę wierzchołków oraz sumaryczną liczbę trójkątów i kwadratów, które powinien mieć graf z i -tego przypadku testowego.

Wyjście

Na wyjściu wypisz kolejno odpowiedzi dla wszystkich przypadków testowych. Każdy z nich powinien mieć formę zgodną z poniższą specyfikacją.

W pierwszym wierszu dla wyjścia i -tego przypadku testowego powinna się znaleźć jedna liczba całkowita $0 \leq K_i \leq \frac{N_i \cdot (N_i - 1)}{2}$, oznaczająca liczbę krawędzi, użytych do konstrukcji, albo liczba -1 jeżeli nie da się zbudować takiego grafu.

Jeżeli konstrukcja jest możliwa, to w i -tym spośród kolejnych k wierszy powinny znaleźć się po dwie liczby całkowite a_i oraz b_i ($a_i \neq b_i$) oddzielone pojedynczym odstępem, oznaczające połączenie nieskierowaną krawędzią między wierzchołkami a_i i b_i . Każdą krawędź w przypadku testowym można wypisać co najwyżej jeden raz.

Ograniczenia

$$1 \leq T \leq 50, 2 \leq N_i \leq 100, 0 \leq M_i \leq \frac{N_i \cdot (N_i - 1)}{2}.$$

Przykład

Input	Output	Explanation
3	0	W czterowierzchołkowej klicie występują cztery trójkąty i trzy kwadraty.
2 0	-1	
5 8	6	
6 7	1 2	
	1 3	
	1 4	
	2 3	
	2 4	
	3 4	

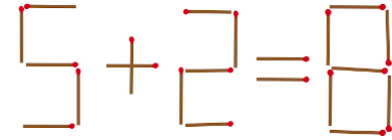
Zapałki (zapalki)

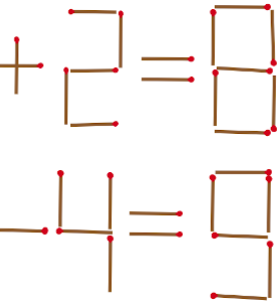
Memory limit: 128 MB

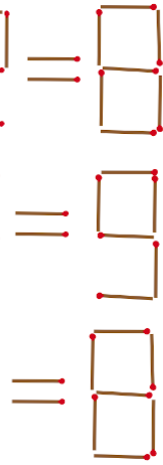
Time limit: 1.00 s


Jasio znalazł ostatnio na kuchennej kilka opakowań starych zapałek. Na opakowaniach znajdowały się rebusy, które wymagały przestawienia **dokładnie jednej** zapałki, tak aby równanie było spełnione.

Niestety Jasio jest bardzo niecierpliwy i zdołał rozwiązać tylko pierwszą z następujących łamigłówek:

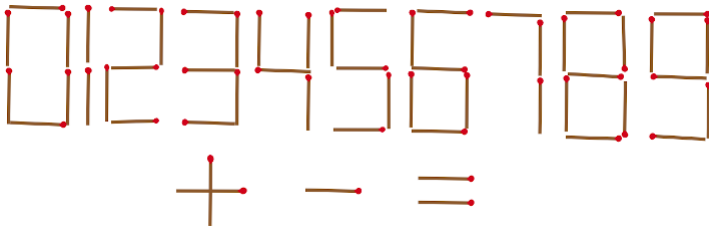
① 

② 

③ 

④ 

Napisz dla Jasia program, który rozwiąże dla niego wszystkie zagadki!



Powyżej przedstawiono wszystkie dozwolone cyfry i symbole (zwrot zapałek nie ma znaczenia).

Wejście

W pierwszym (jedynym) wierszu wejścia znajduje się jedna liczba naturalna t oznaczająca numer łamigłówki.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinno się znaleźć rozwiązanie łamigłówki o danym numerze, napisane w jednej linii bez białych znaków (zgodnie z formatem jak w przykładzie).

Rozwiązania łamigłówek są jednoznaczne.

Ograniczenia

$$1 \leq t \leq 4.$$

Przykład

Input

1

Output

5+3=8