

Krótkie spacery (A)

Limit pamięci: 512 MB

Limit czasu: 1.00 s

Król Bitocji uwielbia krótkie spacery. W szczególności, najbardziej lubi spacery, które trwają dokładnie 2 minuty. Z tego powodu kazał zbudować sieć chodników, po których będzie mógł odbywać swoje przechadzki. Wyzaczył N punktów w swoim ogrodzie. Połączenie dowolnej pary punktów chodnikiem spowoduje, że będzie w stanie przejść między tymi punktami w dokładnie minutę.

Król chciałby teraz połączyć ze sobą pewne z tych punktów chodnikiem tak, żeby z każdego punktu można było przejść chodnikami do dowolnego innego oraz żeby istniało dokładnie K **możliwości wybrania końców ściśle dwuminutowego spaceru**. Spacer między punktami u i v jest ściśle dwuminutowy, jeśli pomiędzy u oraz v nie da się przejść w minutę, ale w 2 minuty jest już to możliwe.

Przykładowo, jeżeli połączymy ze sobą punkty 1 i 2, 2 i 3, to spacer między 1 a 3 punktem będzie ściśle dwuminutowy, ale gdyby dodać chodnik między punktami 1 i 3, to nie byłoby ani jednego ściśle dwuminutowego spaceru.

Jako nadworny programista króla Bitocji, zadanie utworzenia sieci chodników zostało powierzone Tobie. Znajdź sieć chodników spełniającą wymogi króla lub stwierdź, że nie jest to możliwe. Jeżeli istnieje wiele możliwości utworzenia sieci chodników, należy wypisać dowolną z nich.

Wejście

W pierwszym i jedynym wierszu wejścia znajdują się dwie liczby całkowite N oraz K , oznaczające liczbę punktów w ogrodzie oraz oczekiwaną liczbę możliwych do odbycia dwuminutowych spacerów.

Wyjście

Jeżeli nie istnieje sieć chodników spełniająca żądania króla, należy wypisać -1 . W przeciwnym razie należy wypisać jedną liczbę M oznaczającą liczbę chodników. W następnych M wierszach należy wypisać opis chodników. Każdy wiersz powinien składać się z dwóch różnych liczb u oraz v ($1 \leq u, v \leq N$) oznaczających chodnik między punktami u oraz v . Żaden z chodników nie powinien zostać podany więcej, niż raz.

Ograniczenia

$$1 \leq N \leq 1000, 0 \leq K \leq \frac{N(N-1)}{2}.$$

Podzadania

Podzadanie	Warunki	Punkty
1	$K = 0$	20
2	$K \leq N$	20
3	Brak dodatkowych ograniczeń.	60

Przykład

Wejście

5 3

Wyjście

5

2 3

1 3

5 4

3 4

1 2

Wejście

5 8

Wyjście

-1

Zepsuty wyświetlacz (B)

Limit pamięci: 512 MB

Limit czasu: 2.00 s

W tym roku komitet główny Bitockiej Olimpiady Informatycznej Juniorów postanowił ustawić wielki wyświetlacz, na którym pokazane zostaną wyniki wszystkich zawodników, również Twoje. W trakcie zawodów do pokonania będzie N zadań wartych odpowiednio w_1, \dots, w_N punktów. Ponadto, każde zadanie można rozwiązać jedynie w pełni poprawnie lub błędnie, co skutkuje tym, że za i -te zadanie możesz otrzymać dokładnie 0 lub dokładnie w_i punktów. Twoim wynikiem w zawodach jest suma zdobytych punktów.

Zauważyłeś, że wyświetlacz jest zepsuty – jeżeli wynik danego zawodnika dzieli się przez K , wyświetlacz zawsze pokaże, że dany zawodnik otrzymał 0 punktów! Chcesz tego uniknąć za wszelką cenę w swoim przypadku (co pomyślą sobie koledzy, kiedy zobaczą 0 przy Twoim nazwisku?). Zastanawiasz się, jaki jest najwyższy możliwy do osiągnięcia wynik, który poprawnie wyświetli się na wyświetlaczu?

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne N oraz K . W następnym wierszu znajduje się N liczb naturalnych w_1, \dots, w_N .

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczbę oznaczającą największy możliwy wynik, który poprawnie wyświetli się na wyświetlaczu.

Ograniczenia

$1 \leq N \leq 1\,000\,000$, $1 \leq K \leq 10^9$, $1 \leq w_i \leq 10^9$ dla $i = 1, \dots, N$.

Podzadania

Każde podzadanie odpowiada jednej grupie testowej spełniającej warunki podzadania.

Podzadanie	Warunki	Punkty
1	$K = 2$	18
2	$N \leq 1\,000$, suma w_i nie przekracza 20 000.	33
3	Brak dodatkowych ograniczeń.	49

Przykład

Wejście

3 10
10 15 20

Wyjście

45

Wejście

2 5
15 25

Wyjście

0

Gra w skakanie (c)

Limit pamięci: 512 MB

Limit czasu: 1.00 s

Jasio lubi grać w grę planszową o nazwie *Gra w skakanie*. Plansza do gry składa się z $N + 1$ pól ponumerowanych od 0 do N . Ponadto w zestawie jest również kość M -ścienna, z liczbami od 1 do M . Początkowo pionek Jasio znajduje się w polu o numerze 0. Celem gry jest dojście do ostatniego, N -tego pola. W każdej rundzie Jasio rzuca kością i przemieszcza się w przód o taką liczbę pól, jaka wypadła w rzucie kością. Jednakże, niektóre pola są zabronione i jeżeli w jakimkolwiek momencie Jasio miałby stanąć na takim polu, to musi wrócić się na sam początek planszy.

Jasio zaczął zastanawiać się, ile minimalnie rzutów kością musiałby wykonać, żeby bezpiecznie, bez żadnego cofnięcia, dotrzeć do N -tego pola? Jasio jest mały, żeby samemu odpowiedź na to pytanie, dlatego poprosił Cię o pomoc. Napisz program, który wyznaczy minimalną liczbę rzutów kością oraz jakie wartości musiałby wypaść, aby Jasio dotarł do końca planszy!

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite N oraz M oznaczające odpowiednio długość planszy oraz rozmiar kości. W następnym wierszu znajduje się ciąg $N + 1$ znaków `.` oraz `X` oznaczające odpowiednio wolne i zablokowane pole. Pierwszy znak odpowiada polu zerowemu, a ostatni ($N + 1$ -wszy) polu N -temu.

Wyjście

Jeżeli nie istnieje ciąg ruchów, który pozwoli Jasiowi dotrzeć do końca planszy, należy wypisać `-1`. W przeciwnym wypadku, należy wypisać dwa wiersze. W pierwszym wierszu wyjścia należy wypisać jedną liczbę całkowitą, oznaczającą minimalną liczbę rzutów potrzebną do wykonania, aby dojść do ostatniego pola planszy. W następnym wierszu należy wypisać **najmniejszą leksykograficznie** serię rzutów, która osiągnęłaby ten cel. Mówimy, że k -elementowy ciąg a_1, \dots, a_k jest mniejszy leksykograficznie od ciągu b_1, \dots, b_k , gdy $a_1 = b_1, \dots, a_i = b_i$ oraz $a_{i+1} < b_{i+1}$ dla pewnego $1 \leq i < k$.

Ograniczenia

$1 \leq N \leq 1\,000\,000$, $1 \leq M \leq N$. Możesz założyć, że pola o numerach 0 i N są zawsze wolne.

Podzadania

Twój program otrzyma 50% punktów za dany test, jeżeli pierwszy wiersz wyjścia będzie poprawny.

Podzadanie	Warunki	Punkty
1	$N \leq 2\,000$	15
2	$M \leq 20$	15
3	Liczba pól wolnych <code>.</code> nie przekracza 1 000.	20
4	Brak dodatkowych ograniczeń.	50

Przykład

Wejście

5 2
. X . . X .

Wyjście

3
2 1 2

Wejście

5 2
.....

Wyjście

3
1 2 2

Wejście

5 2

.XXXX.

Wyjście

-1