

# Wybór zadań (A)

Limit pamięci: 512 MB

Limit czasu: 1.00 s

Kenarf i Kenaj przygotowali  $N$  zadań na tegoroczne warsztaty przygotowujące do II etapu Bitockiej Olimpiady Informatycznej Juniorów. Każdemu z zadań przyporządkowali jako trudności liczby całkowite  $t_1, \dots, t_N$ . Jednakże, okazało się, że przygotowali o  $K$  za dużo zadań, zatem z niektórych z nich chcieliby zrezygnować. Chcieliby wybrać  $K$  zadań, których się pozbędą, w taki sposób, żeby różnica między najtrudniejszym, a najłatwiejszym zadaniem, które zostaną, była najmniejsza możliwa.

Obaj panowie są bardzo zarobieni przygotowywaniem paczek, dlatego poprosili Ciebie, żebyś pomógł im wybrać zadania.

## Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite  $N$  oraz  $K$  oznaczające odpowiednio liczbę zadań oraz liczbę zadań, z których należy zrezygnować. W następnym wierszu następuje ciąg  $N$  liczb całkowitych  $t_1, \dots, t_N$ , gdzie  $t_i$  oznacza trudność  $i$ -tego zadania.

## Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczbę, oznaczającą najmniejszą możliwą różnicę między najtrudniejszym a najprostszym zadaniem, po zrezygnowaniu z pewnych  $K$  zadań.

## Ograniczenia

$1 \leq N \leq 1\,000\,000$ ,  $0 \leq K < N$ ,  $1 \leq t_i \leq 10^9$ .

## Podzadania

Podzadanie	Warunki	Punkty
1	$N \leq 20$ .	22
2	$N \leq 2\,000$ .	19
3	$t_i = 1$ lub $t_i = 2$ dla wszystkich $i = 1, \dots, N$ .	24
4	Brak dodatkowych ograniczeń.	35

## Przykład

### Wejście

7 1  
3 7 1 3 8 2 10

### Wyjście

7

# Zepsuty wyświetlacz kontratakuje (B)

Limit pamięci: 512 MB

Limit czasu: 1.00 s

Wielki wyświetlacz, przygotowany przez komitet główny Bitockiej Olimpiady Informatycznej Juniorów (ten sam, co w zadaniu *Zepsuty wyświetlacz*), jest bardziej wadliwy, niż początkowo przypuszczano. Nie dość, że źle prezentuje wyniki podzielne przez pewną liczbę, to w dodatku resetowanie go jest niesamowitą udręką! Za każdym razem, aby zmienić to, co w danym momencie się wyświetla, należy najpierw wyczyścić całą matrycę, a dopiero później rysować to, co powinno być w tym momencie wyświetlane.

Wyświetlacz składa się z matrycy  $N \times N$  pikseli. Każdy piksel może przybrać dwa kolory, biały (oznaczony znakiem kropki `.`) lub czarny (oznaczone znakiem krzyżyka `#`). Resetowanie wyświetlacza polega na zmianie wszystkich pikseli tak, żeby były czarne. Jednakże, nie da się tego osiągnąć w bezpośredni sposób. Jedyna możliwość, żeby zmienić to, jak aktualnie świecą się piksele, jest wybranie pewnego wiersza pikseli oraz pewnej kolumny, a następnie przerysowanie tego wiersza w danej kolumnie. Rozważmy poniższy przykład wyświetlacza rozmiaru  $3 \times 3$ :

```
.##  
#. #  
#..
```

W tej sytuacji, gdybyśmy wybrali trzeci wiersz, po czym przerysowali go w kolumnie pierwszej, wyświetlacz zmieniłby swój stan do poniższego:

```
###  
..#  
...
```

Gdyby teraz wybrać pierwszy wiersz, a następnie przerysować go w każdej z kolumn, otrzymalibyśmy wyświetlacz cały ustawiony na czarno:

```
###   ###   ###   ###  
..# -> #.# -> ### -> ###  
...   #..   ##.   ###
```

Okazuje się, że za czyszczenie wyświetlacza (30 razy w każdej sekundzie!) odpowiedzialny jest krasnal Bitek, ukryty pod podłogą sali zawodów (podobnie, jak krasnale sterujące światłami drogowymi, które, jak każdy wie, są ukryte pod jezdnią). Krasnal Bitek ledwo daje radę i bardzo przydałby mu się program, który pomoże mu w czyszczeniu wyświetlacza. Pomóż mu i napisz program, który policzy, ile razy Bitek musi wykonać operację przerysowania, żeby zmienić kolor wszystkich pikseli na czarny.

## Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita  $N$  oznaczająca wymiary wyświetlacza. W następnych  $N$  wierszach następuje opis początkowego stanu wyświetlacza,  $i$ -ty wiersz składa się z  $N$  znaków kropki `.` lub krzyżyka `#`,  $j$ -ty znak opisuje stan  $j$ -tego piksela w  $i$ -tym wierszu.

## Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczbę całkowitą, oznaczającą minimalną liczbę operacji, która wystarczy, aby wszystkie piksele miały czarny kolor, lub `-1`, jeżeli nie jest to możliwe.

## Ograniczenia

$1 \leq N \leq 1000$ .

## Podzadania

Podzadanie	Warunki	Punkty
1	W każdej kolumnie jest co najmniej jeden czarny piksel.	21
2	W każdym wierszu jest co najwyżej jeden biały piksel.	23
3	$N \leq 50$ .	18
4	Brak dodatkowych ograniczeń.	38

## Przykład

### Wejście

```
3
.##
#.#
#..
```

### Wyjście

```
4
```

### Wejście

```
6
.....#
#..#.#
....#.
#..#.#
...##.
.#.....
```

### Wyjście

```
9
```

### Wejście

```
2
..
..
```

### Wyjście

```
-1
```

# Hory Portier i Kanciapa Tajemnic (c)

Limit pamięci: 512 MB

Limit czasu: 1.00 s

Dawno, dawno temu, za 111 górami i 111 lasami, w XIV liceum ogólnomagicznym odbywały się warsztaty przygotowujące do II etapu Bajtockiej Olimpiady Bitów i Charów. Na straży szkoły stał Hory Portier, chroniący uczniów przed nadchodzącymi potworami. Do walki z potworami wykorzystuje wyroby magiczne, oferowane mu przez Serwerusa Skype'a.

Hory Portier potrafi zaglądać w przyszłość (dzięki lekcjom z nikim innym jak jedynym Wróżbitą Maciejem), zatem ma dostępny ciąg zdarzeń, które zajdą w przyszłości. Zdarzenia ułożone są chronologicznie od 1 do  $N$ -tego. Każde ze zdarzeń jest jednego z dwóch typów:

- Do szkoły podchodzi potwór rodzaju  $p$ ,
- Serwerus Skype oferuje Horemu Portierowi miksturę rodzaju  $p$ , którą Hory Portier może przyjąć lub nie.

Hory Portier doskonale zdaje sobie sprawę, że do walki z potworem rodzaju  $p$  wystarczy mu dokładnie jedna mikstura tego samego rodzaju, którą zużywa w trakcie walki. Z drugiej strony, wszystkie mikstury przechowuje w swojej Kanciapie Tajemnic, której to pojemność jest ograniczona. Nie chciałby zatem przechowywać mikstur, które mogłyby okazać się nieprzydatne. Innymi słowy, założmy, że w momencie, w którym Hory Portier przechowuje w swojej Kanciapie najwięcej mikstur, jest ich dokładnie  $K$ . Hory Portier chciałby, żeby wśród wszystkich możliwych strategii przyjmowania mikstur wybrać taką, że  $K$  jest możliwe najmniejsze.

Hory Portier potrafi czarować, ale nie programować, zatem poprosił Cię o pomoc! Mając listę przyszłych wydarzeń oblicz, jaki jest minimalny rozmiar Kanciapy Tajemnic, aby Hory Portier zawsze miał dostępną miksturę przed walką z potworem oraz które z mikstur powinien przyjąć od Serwerusa Skype'a.

## Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita  $N$  oznaczająca liczbę przyszłych wydarzeń. W następnych  $N$  wierszach następuje opis kolejnych zdarzeń,  $i$ -te zdarzenie opisane jest przez jeden znak oraz jedną liczbę  $c_i, p_i$ . Jeżeli  $c_i$  jest równe ! to oznacza, że Hory Portier musi w tym momencie walczyć z potworem typu  $p_i$ . Jeżeli  $c_i$  jest równe +, to oznacza, że Serwerus Skype oferuje Horemu Portierowi miksturę na potwory rodzaju  $p_i$ .

## Wyjście

W pierwszym wierszu wyjścia należy wypisać jedną liczbę całkowitą  $K$  oznaczającą minimalny potrzebny rozmiar Kanciapy Tajemnic lub jedną liczbę  $-1$ , jeżeli niemożliwe jest wygranie ze wszystkimi potworami. Jeżeli  $K \geq 0$ , w drugim wierszu należy wypisać ciąg liczb równych 0 lub 1, oznaczający, które z kolejnych mikstur Hory Portier przyjmuje od przez Serwerusa Skype'a według kolejności wydarzeń typu + na wejściu (0 oznacza nieprzyjęcie, a 1 przyjęcie mikstury). Jeżeli istnieje wiele możliwości, należy wpisać dowolne z nich.

## Ograniczenia

$$1 \leq N \leq 200\,000, 1 \leq p_i \leq N.$$

## Podzadania

W każdym zestawie testowym otrzymasz 50% punktów jeżeli pierwszy wiersz wyjścia będzie poprawny.

Podzadanie	Warunki	Punkty
1	Wszystkie zdarzenia typu + następują przed wszystkimi typu !.	21
2	$p_i = 1$ dla każdego $i = 1, \dots, N$ .	26
3	$N \leq 2\,000$ .	23
4	Brak dodatkowych ograniczeń.	30

## Przykład

### Wejście

14  
+ 2  
+ 1  
+ 1  
+ 2  
+ 4  
+ 3  
! 1  
! 3  
+ 4  
! 2  
+ 2  
! 2  
! 4  
! 1

### Wyjście

4  
0 1 1 1 0 1 1 1

### Wyjaśnienie

Hory Portier mógłby wziąć wszystkie pierwsze 6 mikstur napotkanych na swojej drodze i już do końca mieć w swoim orężu potrzebne mikstury do walki z potworami. Jednakże, aby oszczędzić miejsce w Kanciapie Tajemnic, mógłby zrezygnować z którejkolwiek mikstury rodzaju 2 oraz z pierwszej napotkanej mikstury rodzaju 4 i przyjąć dopiero następne, które się pojawią, dzięki czemu w najgorszym momencie musi mieć pod ręką 4 mikstury.

### Wejście

3  
+ 2  
! 1  
+ 1

### Wyjście

-1