

## Задача А. Масив і ще кілька масивів

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	2 seconds
Ліміт використання пам'яті:	256 megabytes

Задано  $k$  масивів цілих чисел  $a_1, a_2, \dots, a_k$ . Масив з номером  $i$  містить  $l_i$  елементів. Нехай  $n = l_1 + l_2 + \dots + l_k$ .

Вам потрібно знайти  $k$  цілих чисел  $d_1, d_2, \dots, d_k$  таких, що числа  $(a_{i,j} + d_i)$  попарно різні, а також  $1 \leq a_{i,j} + d_i \leq n$ .

### Формат вхідних даних

У першому рядку задано два цілі числа  $n, k$  ( $1 \leq n \leq 10^4, 1 \leq k \leq 5$ ) — загальна кількість елементів у масивах та кількість масивів відповідно.

У наступних  $k$  рядках задано масиви. В  $i$ -му рядку задано ціле число  $l_i$  ( $1 \leq l_i \leq n$ ) та  $l_i$  цілих чисел  $a_{i,1}, a_{i,2}, \dots, a_{i,l_i}$  ( $1 \leq a_{i,j} \leq n$ ) — довжина  $i$ -го масиву та його елементи відповідно.

Гарантується, що  $n = l_1 + l_2 + \dots + l_k$ .

### Формат вихідних даних

Якщо шуканих значень  $d$  не існує, виведіть єдиний рядок «No».

Інакше у першому рядку виведіть «Yes».

У другому рядку виведіть  $k$  цілих чисел  $d_1, d_2, \dots, d_k$  — значення, які потрібно додати до елементів масивів, щоб загалом утворилось  $n$  різних цілих чисел від 1 до  $n$ .

Якщо існує кілька правильних відповідей, дозволяється вивести будь-яку з них.

### Система оцінки

- (8 балів):  $k = 1$ ;
- (9 балів):  $a_{i,j} + 1 = a_{i,j+1}$  для  $1 \leq i \leq k, 1 \leq j < l_i$ ;
- (15 балів):  $k \leq 2$ ;
- (21 бал):  $k \leq 3$ ;
- (10 балів):  $a_{i,j} + 2 = a_{i,j+1}$  для  $1 \leq i \leq k, 1 \leq j < l_i$ ;
- (10 балів):  $(\max_{j \in [1;l_i]} a_{i,j}) - (\min_{j \in [1;l_i]} a_{i,j}) = (n - k)$  для  $1 \leq i \leq k$ ;
- (10 балів):  $n \leq 30$ ;
- (17 балів): без додаткових обмежень.

## Приклади

standard input	standard output
5 5 1 1 1 2 1 3 1 4 1 5	Yes 0 0 0 0 0
6 4 2 2 3 1 6 1 4 2 1 5	Yes 1 -5 1 1
7 2 4 1 4 5 6 3 1 2 6	Yes 0 1
4 2 2 2 3 2 2 4	No

## Зауваження

У першому прикладі  $d = [0, 0, 0, 0, 0]$  задовільняє умові, адже після додавання відповідних значень утворюються масиви  $[1]$ ,  $[2]$ ,  $[3]$ ,  $[4]$ ,  $[5]$ .

У другому прикладі  $d = [1, -5, 1, 1]$  задовільняє умові, адже після додавання відповідних значень утворюються масиви  $[3, 4]$ ,  $[1]$ ,  $[5]$ ,  $[2, 6]$ .

У третьому прикладі  $d = [0, 1]$  задовільняє умові, адже після додавання відповідних значень утворюються масиви  $[1, 4, 5, 6]$  та  $[2, 3, 7]$ .

## Задача В. Масив монет і запити зважування

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	1 second
Ліміт використання пам'яті:	256 megabytes

*Це інтерактивна задача.*

Є  $n$  монет, розташованих в ряд та пронумерованих зліва направо цілими числами від 1 до  $n$ .

Рівно  $k$  ( $k < n$ ) з цих монет є фальшивими, інші  $(n - k)$  монет — справжні. Фальшиві монети легші за справжні. Всі справжні монети мають однакову вагу, в той час як фальшиві монети можуть мати різну вагу. Також відомо, що фальшиві монети **розташовані поспіль**, тобто мають номери  $p, p + 1, \dots, (p + k - 1)$ .

Вам потрібно знайти номер найлівоїшої фальшивої монети. Ви можете користуватись зважуванням, що аналогічне зважуванню на двочашкових терезах: вибрати дві множини монет, що не перетинаються, і дізнатись, яка з множин важить більше, або що множини важать однаково.

### Формат вхідних даних

У першому рядку задано три цілі числа  $n, k, g$  ( $1 \leq k < n \leq 10^4, 0 \leq g \leq 6$ ) — загальна кількість монет, кількість фальшивих монет та номер блоку тестів відповідно.

### Формат вихідних даних

Для того, щоб дати відповідь, виведіть єдиний рядок у форматі «!  $p$ », де  $p$  ( $1 \leq p \leq n$ ) — номер найлівоїшої фальшивої монети.

### Протокол взаємодії

Щоб виконати запит зважування виведіть «?  $s_1 s_2 a_1 a_2 \dots a_{s_1} b_1 b_2 \dots b_{s_2}$ », де  $s_1$  та  $s_2$  позначають розміри множин які зважуються, а масиви  $a$  та  $b$  позначають номери монет, що належать першій та другій множинам відповідно.

У відповідь на запит програма журі виведе одне ціле число  $x$  ( $x \in \{0, 1, 2\}$ ). Якщо  $x = 1$ , то перша множина важча за другу; якщо  $x = 2$ , то друга множина важча за першу; якщо  $x = 0$ , то множини мають однакову вагу.

Якщо запит невалідний (тобто перевищено максимальну кількість запитів або параметри запиту є невалідними), програма журі виведе -1 та припинить роботу. У такому випадку завершіть роботу програми, щоб отримати вердикт **Неправильна відповідь**.

Подбайте про виклик методу `flush` після виводу кожного рядка. Для цього можна використовувати:

- `fflush(stdout)`, `cout << endl` або `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `sys.stdout.flush()` в Python;
- дивіться документації для інших мов програмування.

### Система оцінки

Визначимо  $q$  як максимальну кількість запитів зважування, яку ви можете виконати у тестах певного блоку.

1. (5 балів):  $n \leq 16, q = 16$ ;
2. (9 балів):  $k = 1, q = 16$ ;
3. (7 балів):  $k = 1, q = 11$ ;

4. (16 балів):  $k \leq 16$ ,  $q = 11$ ;
5. (9 балів): ваги всіх фальшивих монет однакові,  $q = 11$ ;
6. (до 54 балів):  $q = 300$ . Нехай максимальна кількість використаних зважувань дорівнює  $c$ . Якщо  $c \leq 9$ , то ви отримаєте 54 бали, а інакше ви отримаєте  $\lfloor 54 \cdot \max(-0.0004 \cdot c + 0.3134, 0.018 + \frac{9.0773}{c}) \rfloor$  балів.

Код на мові C++, який обчислює кількість балів за останній блок тестів залежно від кількості використаних зважувань:

```
((c <= 9) ? 54 : int(54 * (max((-0.0004 * c + 0.3134), (0.018 + 9.0773 / c))))))
```

Таблиця розподілу балів

$c \leq 17$	Кількість балів	$18 \leq c \leq 27$	Кількість балів	$28 \leq c \leq 300$	Кількість балів
$\leq 9$	54	18	28	28	18
10	49	19	26	29-30	17
11	45	20	25	31-42	16
12	41	21	24	43-89	15
13	38	22	23	90-135	14
14	35	23	22	136-181	13
15	33	24	21	182-227	12
16	31	25	20	228-274	11
17	29	26-27	19	275-300	10

### Приклад

standard input	standard output
4 1 0	? 1 1 1 2
0	? 1 1 2 4
0	? 1 1 3 4
2	! 3

## Задача С. Масив і додавання на відрізку

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: 2 seconds  
Ліміт використання пам'яті: 256 megabytes

Задано масив цілих чисел  $a$  довжини  $n$ .

Ви можете змінювати масив, використовуючи *операцію додавання*. Для того, щоб застосувати *операцію додавання*, необхідно виконати три послідовні дії:

1. вибрати довільне ціле число  $x$ ;
2. вибрати довільний підвідрізок масиву  $[l; r]$ ;
3. до кожного елемента вибраного підвідрізка додати  $x$  (виконати операцію присвоєння  $a_i \leftarrow (a_i + x)$  для  $l \leq i \leq r$ ).

Знайдіть мінімальну кількість *операцій додавання*, необхідну для того, щоб усі елементи масиву  $a$  стали попарно різними.

### Формат вхідних даних

У першому рядку задано одне ціле число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — довжина масиву.

У другому рядку задано  $n$  цілих чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — елементи масиву.

### Формат вихідних даних

Виведіть одне ціле число — мінімальну кількість *операцій додавання*, необхідну для того, щоб усі елементи масиву  $a$  стали попарно різними.

### Система оцінки

1. (9 балів): усі елементи масиву  $a$  рівні 1;
2. (15 балів):  $1 \leq a_i \leq 2$  для  $1 \leq i \leq n$ ;  $a_i \leq a_{i+1}$  для  $1 \leq i < n$ ;
3. (14 балів):  $n \leq 8$ ;
4. (17 балів):  $a_1 = a_n$ ;
5. (12 балів):  $n \leq 2000$ ;
6. (12 балів):  $1 \leq a_i \leq 100$  для  $1 \leq i \leq n$ ;
7. (21 бал): без додаткових обмежень.

### Приклади

standard input	standard output
3 1 2 3	0
5 2 3 2 3 2	2
9 2 3 1 1 3 2 1 3 3	2

### Зауваження

У першому прикладі усі елементи масиву  $a$  є попарно різними.

У другому прикладі після застосування двох *операцій додавання* з параметрами  $x = -3$ ,  $l = 1$ ,  $r = 2$  та  $x = -1$ ,  $l = 1$ ,  $r = 3$  масив  $a$  стає рівним  $[-2, -1, 1, 3, 2]$ .

У третьому прикладі після застосування двох *операцій додавання* з параметрами  $x = -3$ ,  $l = 4$ ,  $r = 8$  та  $x = -10$ ,  $l = 7$ ,  $r = 9$  масив  $a$  стає рівним  $[2, 3, 1, -2, 0, -1, -12, -10, -7]$ .

## Задача D. Масив і часткові суми

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	2 seconds
Ліміт використання пам'яті:	256 megabytes

Масивом префіксних сум масиву цілих чисел  $a$  довжини  $n$  називається такий масив  $b$  довжини  $n$ , що  $b_i = a_1 + a_2 + \dots + a_i$ .

Масивом суфіксних сум масиву цілих чисел  $a$  довжини  $n$  називається такий масив  $b$  довжини  $n$ , що  $b_i = a_n + a_{n-1} + \dots + a_i$ .

Назвемо *нормалізацією* масиву цілих чисел  $a$  довжини  $n$  виконання операції присвоєння  $a_i \leftarrow \max(\min(a_i, 10^{18}), -10^{18})$  для  $1 \leq i \leq n$ .

Задано масив цілих чисел  $a$  довжини  $n$ .

Дозволяється виконувати операції трьох типів:

1. замінити кожен елемент масиву  $a$  на протилежний (виконати операцію присвоєння  $a_i \leftarrow (-a_i)$  для  $1 \leq i \leq n$ );
2. вибрати довільний підвідрізок масиву  $a$  та замінити його на масив його префіксних сум, після чого *нормалізувати* масив  $a$ ;
3. вибрати довільний підвідрізок масиву  $a$  та замінити його на масив його суфіксних сум, після чого *нормалізувати* масив  $a$ .

Знайдіть найкоротшу послідовність операцій, необхідних для того, щоб усі елементи масиву  $a$  стали невід'ємними.

Зверніть увагу, що для деяких блоків тестів дозволяється знайти послідовність операцій, що не є найкоротшою з можливих.

### Формат вхідних даних

У першому рядку задано два цілі числа  $n, g$  ( $1 \leq n \leq 2 \cdot 10^5, 0 \leq g \leq 8$ ) — довжина масиву та номер блоку тестів відповідно.

У другому рядку задано  $n$  цілих чисел  $a_1, a_2, \dots, a_n$  ( $-1 \leq a_i \leq 1$ ) — елементи масиву.

### Формат вихідних даних

У першому рядку виведіть одне ціле число  $m$  — мінімальну кількість операцій, необхідних для того, щоб усі елементи масиву  $a$  стали невід'ємними.

У наступних  $m$  рядках виведіть опис операцій. Описи операцій першого типу виводіть у форматі «1». Описи операцій другого та третього типів виводіть у форматах «2 l r» та «3 l r» відповідно, де  $l$  та  $r$  ( $1 \leq l \leq r \leq n$ ) позначають ліву та праву межі підвідрізка чергової операції.

Якщо існує кілька правильних відповідей, дозволяється вивести будь-яку з них.

### Система оцінки

Нехай для певного тесту мінімальна кількість операцій, необхідних для того, щоб усі елементи масиву  $a$  стали невід'ємними, рівна  $m_{ans}$ , а ваше рішення використовує  $m_{user}$  операцій.

1. (14 балів):  $m_{ans} \leq 1$ ;
2. (17 балів): ваш розв'язок вважатиметься правильним, якщо  $m_{user} \leq 100$ . Можна довести, що завжди існує послідовність не більше ніж з 100 операцій при заданих обмеженнях;
3. (18 балів): ваш розв'язок вважатиметься правильним, якщо  $m_{user} \leq m_{ans} + 3$ ;
4. (7 балів): ваш розв'язок вважатиметься правильним, якщо  $m_{user} \leq m_{ans} + 1$ ;
5. (7 балів):  $n \leq 3000$ ; гарантується, що **усі** найкоротші послідовності операцій містять **лише** операції другого типу;

6. (19 балів): гарантується, що **усі** найкоротші послідовності операцій містять **лише** операції другого типу;
7. (17 балів):  $n \leq 3000$ ;
8. (1 бал): без додаткових обмежень.

### Приклад

standard input	standard output
7 0	2
0 0 1 -1 -1 -1 1	3 1 3
	2 1 7

### Зауваження

У першому прикладі масив  $a$  змінюється двічі:

1. після виконання операції третього типу з параметрами  $l = 1$ ,  $r = 3$  масив  $a$  стає рівним  $[1, 1, 1, -1, -1, -1, 1]$ ;
2. після виконання операції другого типу з параметрами  $l = 1$ ,  $r = 7$  масив  $a$  стає рівним  $[1, 2, 3, 2, 1, 0, 1]$ .