

## Задача А. Графічний процесор

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	0.5 seconds
Ліміт використання пам'яті:	256 megabytes

Багато років тому персональні комп'ютери мали лише текстовий екран, тобто, могли виводити інформацію на екран лише за допомогою букв, цифр, розділових знаків та інших символів. Але це не означає, що тодішні користувачі відмовилися від зображень!

Виявилося, що за допомогою різних символів можна будувати різні зображення. Оскільки вручну це робити доволі довго, пропонуємо скласти програму, яка, прочитавши код зображення, відтворить його на текстовому «екрані» шириною  $W$  та висотою  $H$  символів, спочатку заповненому символами, які позначають тло.

Код зображення, крім інформації про розміри та тло, міститиме графічні команди.

Загальна структура команди:

<код команди><пропуск><код кольору><пропуск><параметри, відокремлені пропусками>

Тут:

<код команди> — велика латинська буква (H — горизонтальна лінія, V — вертикальна лінія, R — прямокутник, S — заповнений прямокутник, F — заливка)

<код кольору> — символ, з якого будується фігура або яким заливається ділянка зображення. Початок координат — символ з координатами (0, 0) — розташований у лівому верхньому куті. Вісь X напрямлена вправо, вісь Y — вниз.

Наприклад:

– H # 0 0 50 - горизонтальна лінія зі символів '#' від (0,0) вправо довжиною 50

– V & 21 14 60 - вертикальна лінія зі символів '&' від (21,14) вниз довжиною 60

– R # 30 11 8 12 - прямокутник із символів '#' з лівим верхнім кутом (30,11) шириною 8 і висотою 12

– S ! 10 4 6 20 - зафарбований прямокутник із символів '!' з лівим верхнім кутом (10,4) шириною 6 і висотою 20

– F \* 30 11 - заливання символом '\*', починаючи з (30,11). Заливання поширюється вгору, вниз, вліво і вправо, замінюючи такі символи, як той, що був у позиції початку заливання.

### Обмеження

$3 \leq W, H \leq 100$

$1 \leq N \leq 100$

<код команди> - символи проміжку 'A'..'Z'

<код кольору>, <позначка тла> - малі англійські букви, символи '!', '#', '\$', '%', '^', '&', '5', '?', ':', '\*', '!', '!', '=', '+', '%'

### Формат вхідних даних

У першому рядку вхідного потоку містяться два цілі числа  $W$  і  $H$  — ширина та висота екрану.

У другому рядку міститься символ — позначка для тла.

У третьому рядку міститься одне ціле число  $N$  — кількість команд.

У наступних  $N$  рядках містяться коди команд, по одному в рядку.

### Формат вихідних даних

Вивести у вихідний потік  $H$  рядків по  $W$  символів у кожному — зображення на екрані після опрацювання команд.

## Приклади

standard input	standard output
8 5 - 1 V * 5 0 4	-----*-- -----*-- -----*-- -----*-- -----
10 10 . 5 R * 0 4 10 6 S * 3 0 4 3 R & 5 3 1 2 F 5 8 3 H ! 3 6 4	...***555 ...***555 ...***555 .....&5555 *****&***** *.....* *...!!!...* *.....* *.....* *****

## Задача В. Зубчик і гральні кубики

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	0.5 seconds
Ліміт використання пам'яті:	256 megabytes

Великий Зуб маленькому Зубчику подарував набір із  $n$  гральних кубиків. На кожній стороні грального кубика вказано число у вигляді точок від одного до шести. Зубчик, зрадівши, висипав усі кубики на поверхню стола і зразу ж отримав завдання від самого Зуба: а за яку мінімальну кількість перевертань, можна досягти файного стану. Файний стан, на думку Зуба, це таке розташування кубиків, при якому на верхній площині усіх кубиків зображене одне і теж число.

### Формат вхідних даних

Перший рядок стандартного потоку містить натуральне число  $t$  ( $1 \leq t \leq 10^3$ ) — кількість наборів вхідних тестів.

Перший рядок кожного тестового набору містить натуральне число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — кількість кубиків у наборі.

Другий рядок містить  $n$  натуральних чисел  $a_i$  ( $1 \leq a_i \leq 6$ ), кожне із яких вказує на число, що зображене на верхній площині кубика.

Гарантується, що сума  $n$  за всіма наборами тестів не перевищує  $2 \cdot 10^5$ .

### Формат вихідних даних

Для кожного набору вхідних даних виведіть мінімальну кількість перевертань, щоб розташування кубиків на поверхні стола було файним та через пропуск вкажіть число яке буде на поверхні файного розташування.

Якщо варіантів декілька, то вкажіть мінімальне число.

### Приклад

standard input	standard output
4	2 1
4	1 3
1 2 1 2	0 5
2	3 1
6 3	
4	
5 5 5 5	
5	
1 2 3 4 1	

### Зауваження

У першому тестовому наборі можна перевернути два кубика із верхньою стороною 1, отримаємо файне розташування у якому верхня сторона у сіх кубиків буде 2 або перевернути два кубика із двійкою, тоді файне розташування буде містити кубики на верхній стороні яких буде одиничка.

## Задача С. Вартість доставки

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

Байтозавр — власник транспортної компанії. Його компанія існує вже 5 років і держава змушує його налагодити мережу перевезень по всій країні. У країні є  $n$  міст у координатах  $(X_i, Y_i)$ . Вартість сполучення маршрутом пари міст  $a$  і  $b$  —  $\min(|X_a - X_b|, |Y_a - Y_b|)$ . Виведіть мінімальну вартість сполучення усіх міст між собою, тобто щоб між кожною парою міст існувала якась послідовність маршрутів яка дозволяє дістатись з одного міста в інше.

### Формат вхідних даних

У першому рядку дано одне ціле число  $n$  — кількість міст; ( $1 \leq n \leq 10^5$ ) У наступних  $n$  рядках дані пари цілих чисел  $X_i Y_i$  — координати  $i$  — міста; ( $0 \leq X_i, Y_i \leq 10^9$ )

### Формат вихідних даних

Виведіть одне ціле число — мінімальну вартість сполучення усіх міст між собою, тобто щоб між кожною парою міст існувала якась послідовність маршрутів яка дозволяє дістатись з одного міста в інше.

### Приклади

standard input	standard output
3 1 2 0 1 2 2	1
4 0 5 5 0 2 8 2 9	5

## Задача D. Недружні друзі

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

У всіх бувають погані дні. Можна коритись долі, а можна взяти все в свої руки. Саме така ситуація і відбувається зараз, клас вирушив у поїздку до Карпат. Клас розбився на групки, одна з цих групок розміру  $k$  «друзі» на думку класного керівника, але насправді терпіти одне-одного не можуть. Вони мусять зіграти в гру суть якої в тому щоб сидіти на  $k$  різних пеньках і передавати повідомлення по колу (тобто є якийсь порядок гравців починаючи з першого і гравець 1 передає гравцю 2, 2 — 3, ...,  $k - 1 - k$ ). Між пеньками є стежки по яких можна ходити щоб передати повідомлення, усього є  $n$  пеньків і  $m$  стежок, стежки можуть мати різну довжину (час проходження), повідомлення отримуються миттєво, час займає тільки проходження стежки. Потрібно знайти такий порядок пеньків що невдоволеність гравців буде мінімальною. Невдоволеність гравців — визначимо як добуток довжин стежок по яких пройде хоча б один гравець. Між двома сусідніми в порядку пеньками мусить існувати пряма стежка і гравець йтиме від свого пенька до наступного тільки по прямій стежці, якщо існує кілька таких стежок — по стежці мінімальної довжини.

### Обмеження

$$1 \leq n, m \leq 10^3$$

$$1 \leq k \leq 6$$

$$1 \leq u, v \leq n$$

$$1 \leq w \leq 100$$

Усі вхідні дані є цілими числами

### Формат вхідних даних

У першому рядку дані три числа:  $n$   $m$   $k$  - кількість доріг і кількість ребер і розмір групи.

У наступних  $m$  рядках дана інформація про стежки, по три числа:  $u$   $v$   $w$  - двухстороння стежка між пеньками з індексами  $u$  і  $v$  довжини  $w$ .

### Формат вихідних даних

У першому рядку виведіть мінімальну сумму невдоволеностей.

У другому рядку виведіть індекси пеньків в порядку при якому досягається така сума.

Якщо існує кілька правильних відповідей - виведіть будь-яку.

Якщо відповіді не існує — виведіть  $-1$ .

## Приклади

standard input	standard output
3 3 3 1 2 1 2 3 4 3 1 4	4 3 2 1
3 2 4 1 2 1 2 3 4	-1
8 6 4 1 2 1 2 3 4 3 4 5 5 6 2 6 7 2 7 8 2	8 5 6 7 8
6 6 6 1 2 5 2 3 6 3 4 1 4 5 10 5 6 6 6 1 9	1620 4 3 2 1 6 5