

## Задача А. Бюджет Флатландии

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Кабинет министров Флатландии планирует бюджет на 2024 год. Всего за 2024 год планируется потратить  $N$  единиц валюты. Кроме того члены кабинета министров уже сейчас знают, что каждый месяц будет расходоваться  $K$  единиц валюты, а каждый третий месяц запланированы дополнительные траты на  $F$  единиц валюты, каждый четвертый — на  $D$ , и, наконец, каждый шестой — на  $S$ . При этом дополнительные траты, запланированные в каком-то месяце, распространяются только на этот месяц и никак не влияют на траты в иных месяцах. Также, если настал неудачный месяц, в котором запланировано несколько дополнительных трат — они суммируются.

Кабинет министров Флатландии обратился к Вам за помощью. Ваша задача — вычислить минимальную сумму, на которую необходимо увеличить первоначальный планируемый бюджет, чтобы единиц валюты хватило на весь год.

### Формат входных данных

В первой строке вводится число  $N$  ( $0 \leq N \leq 10^{16}$ ) — планируемый бюджет на год.

Во второй строке вводится число  $K$  ( $0 \leq K \leq 10^{16}$ ) — траты каждый месяц;

В третьей строке вводится число  $F$  ( $0 \leq F \leq 10^{16}$ ) — дополнительные траты каждый третий месяц;

В четвертой строке вводится число  $D$  ( $0 \leq D \leq 10^{16}$ ) — дополнительные траты каждый четвертый месяц;

В пятой строке вводится число  $S$  ( $0 \leq S \leq 10^{16}$ ) — дополнительные траты каждый шестой месяц.

### Формат выходных данных

Ваша задача — вывести одно число: минимальную сумму, на которую необходимо увеличить начальный планируемый бюджет, чтобы единиц валюты хватило на весь год.

### Примеры

стандартный ввод	стандартный вывод
10 1 2 3 4	27
100 3 4 5 6	0

### Замечание

Разберем первый тестовый пример.

Начальный планируемый бюджет: 10.

Каждый месяц расходуется ровно 1 единица валюты.

Каждый третий месяц расходуется на 2 единицы валюты больше.

Каждый четвертый месяц расходуется на 3 единицы валюты больше.

Каждый шестой месяц расходуется на 4 единицы валюты больше.

Тогда получаем:

1) В первый месяц расходы составят: 1 (ежемесячные траты);

2) Во второй месяц расходы составят: 1 (ежемесячные траты);

3) В третий месяц расходы составят:  $1 + 2 = 3$  (ежемесячные траты + траты каждый третий месяц);

4) В четвертый месяц расходы составят:  $1 + 3 = 4$  (ежемесячные траты + траты каждый четвертый месяц);

5) В пятый месяц расходы составят: 1 (ежемесячные траты);

6) В шестой месяц расходы составят:  $1 + 2 + 4 = 7$  (ежемесячные траты + траты каждый третий месяц + траты каждый шестой месяц);

7) В седьмой месяц расходы составят: 1 (ежемесячные траты);

8) В восьмой месяц расходы составят:  $1 + 3 = 4$  (ежемесячные траты + траты каждый четвертый месяц);

9) В девятый месяц расходы составят:  $1 + 2 = 3$  (ежемесячные траты + траты каждый третий месяц);

10) В десятый месяц расходы составят: 1 (ежемесячные траты);

11) В одиннадцатый месяц расходы составят: 1 (ежемесячные траты);

12) В двенадцатый месяц расходы составят:  $1 + 2 + 3 + 4 = 10$  (ежемесячные траты + траты каждый третий месяц + траты каждый четвертый месяц + траты каждый шестой месяц).

Итого за 12 месяцев расходы составят:  $1 + 1 + 3 + 4 + 1 + 7 + 1 + 4 + 3 + 1 + 1 + 10 = 37$ .

Нехватка бюджета:  $37 - 10 = 27$  (траты за 12 месяцев - начальный планируемый бюджет).

Ответ: 27.

## Задача В. Фонари в Мытицах

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Один из составителей олимпиады по имени Игорь живет в Мытицах. Не так давно он заметил, что расположение фонарей в Мытицах позволяет представить их, расставленных в точках пересечения  $N$  параллельных прямых и перпендикулярных им  $M$  параллельных прямых. Для удобства такую схему можно представить в виде таблицы  $N$  на  $M$ , где в точках пересечения прямых, составляющих таблицу, стоят фонари (Рис. 1).

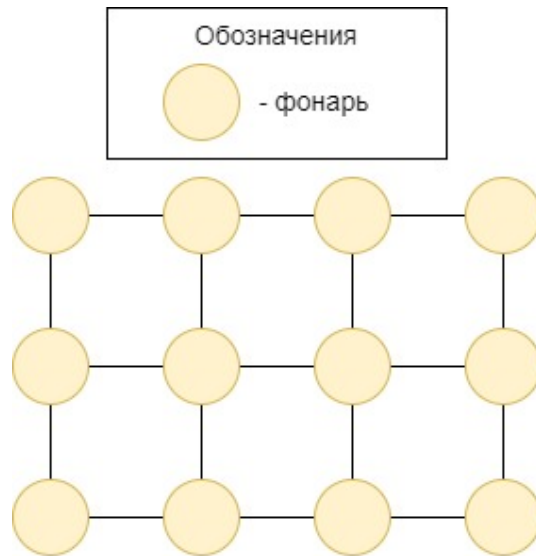


Рис. 1

Для удобства пронумеруем все фонари натуральными числами слева направо, начиная с расположенного в левом верхнем углу (Рис. 2).

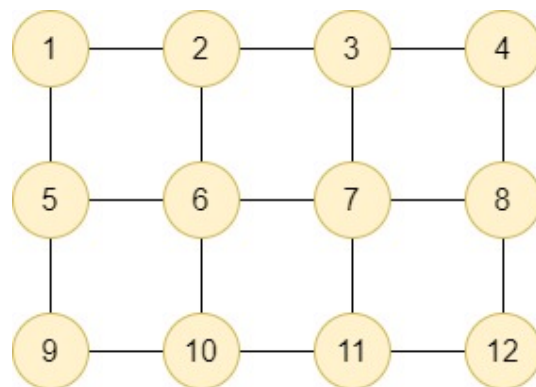


Рис. 2

Недавно, возвращаясь домой, Игорь заметил, что фонари загораются в случае, если пройти под ними. А некоторые фонари также загораются, если загорается какой-то соседний фонарь. Под соседним фонарем будем подразумевать фонарь, расположенный над / под рассматриваемый фонарем и слева / справа от него. У фонарей, расположенных с краю, некоторых соседей нет. Так, например, для фонаря номер 6 с рисунка 2 соседями являются фонари с номерами 2, 5, 7, 10. А для фонаря с номером 1 — фонари с номерами 2, 5.

Потратив не один день на исследование фонарной системы Мытищ, Игорь выяснил, что всего  $K$  фонарей с номерами  $k_1, k_2, k_3, \dots, k_K$  загораются при загорании соседних. Далее Игорю стало

интересно — под сколькими фонарями минимально нужно пройти, чтобы загорелись все фонари в городе. Для простоты задачи предположим, что если какой-то фонарь уже включился, то больше он не гаснет, а изначально все фонари являются выключенными. К сожалению, сейчас у Игоря нет времени для написания программы, решающей эту непростую задачу, поэтому он обратился к Вам с этой задачей.

### Формат входных данных

В первой строке подаются два числа:  $N$  и  $M$ , ( $1 \leq N, M \leq 700$ ).

Во второй строке на вход подается число  $K$  ( $0 \leq K \leq N * M$ ) — количество фонарей, который загораются при зажигании соседних фонарей.

В третьей строке через пробел на вход подуются  $K$  чисел ( $1 \leq k_K \leq N * M$ ) — номера фонарей, которые загораются при зажигании соседних фонарей.

### Формат выходных данных

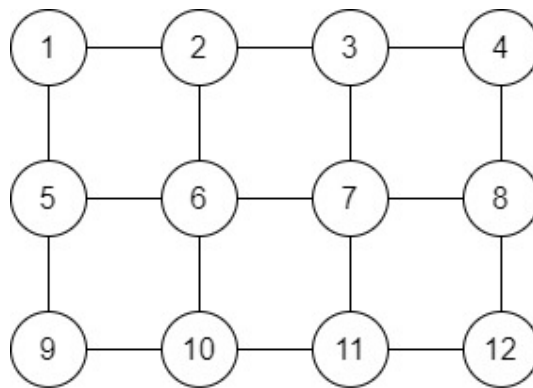
Ваша задача — вывести одно число: минимальное количество фонарей, под которыми необходимо пройти, чтобы все фонари в городе загорелись.

### Примеры

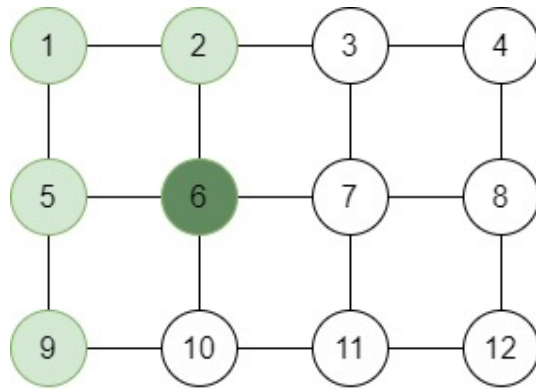
стандартный ввод	стандартный вывод
3 4 8 1 2 4 5 8 9 11 12	4
2 2 4 1 2 3 4	1

### Замечание

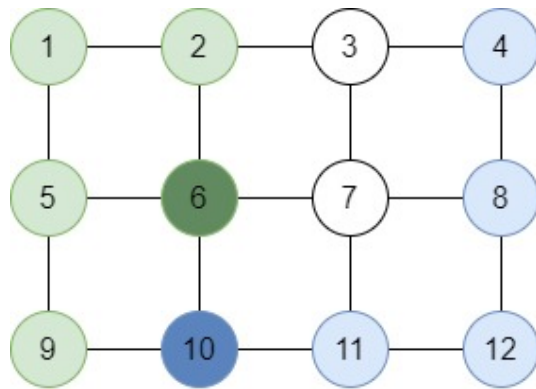
Разберем первый пример.



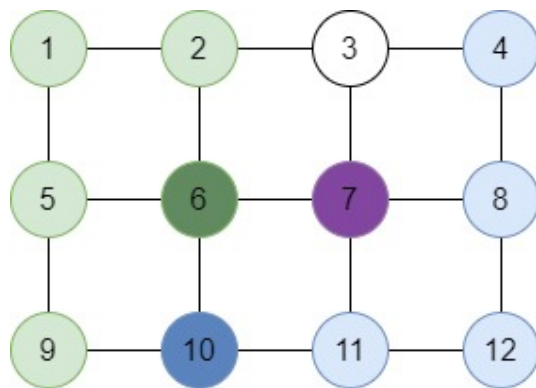
Для начала пройдем под фонарем с номером 6. Фонари 2 и 5 также зажгутся, так как они являются соседями и реагирует на включение соседних фонарей. Фонари 1 и 9 зажгутся так как зажегся фонарь 5, а они реагируют на включение соседних. Таким образом, пройдя под фонарем 6 сразу активировали фонари 1, 2, 5, 6, 9.



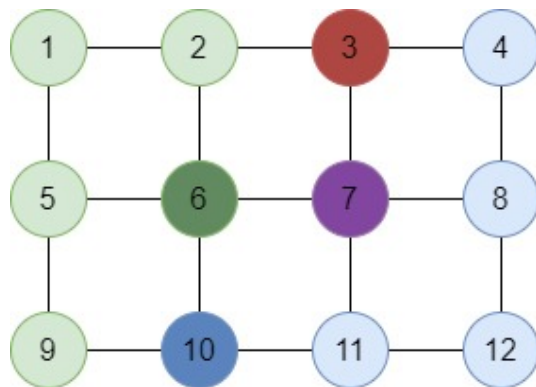
Далее пройдем под фонарем 10. По цепочке активируются фонари 11, 12, 8, 4 (так как они реагируют на включение соседних фонарей).



Далее пройдемся под фонарем 7.



В конце пройдемся под фонарем 3.



В итоге мы прошли под 4 фонарями и зажгли все фонари в городе. Ответ - 4.

## Задача С. Суперкратные числа

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Назовем число суперкратным  $k$ , если оно кратно  $k$ , а так же его позиция в списке всех чисел, кратных  $k$  (список, как и нумерация в нём начинаются с нуля) также кратна  $k$ .

Рассмотрим пример:

Числа, кратные 2: [0, 2, 4, 6, 8, 10, ...].

Числа, кратные 3: [0, 3, 6, 9, 12, ...].

Выделенные числа являются суперкратными 2 и 3 соответственно.

Для данного  $n$  найдите ближайшее меньшее или равное суперкратное  $k$  число. Гарантируется, что  $n \geq k$ .

### Формат входных данных

В первой строке входных данных находится число  $k$ , ( $1 \leq k \leq 2 \times 10^9$ ). Во второй строке находится число  $n$ , ( $k \leq n \leq 2 \times 10^9$ ).

### Формат выходных данных

Выведите одно число — ближайшее суперкратное  $k$  число, меньшее или равное  $n$ .

### Система оценки

Решения, корректно работающие при  $n, k \leq 1000$  будут набирать не менее 30 баллов.  
Решения, корректно работающие при  $n, k \leq 100000$  будут набирать не менее 60 баллов.

### Примеры

стандартный ввод	стандартный вывод
3 11	9
5 16	0

## Задача D. Мультикурсор

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Во многих современных редакторах кода есть возможность использовать несколько курсоров. Такая возможность называется мультикурсором. Это ускоряет процесс написания кода, поскольку можно одновременно внести изменения в несколько строк, исправить ошибки или вставить символы.

Ваша задача — проэмулировать работу текстового редактора, поддерживающего мультикурсор. Вам необходимо обрабатывать следующие действия:

1. Добавить новый курсор на строку номер  $l$  после символа  $i$ . Если  $i$  равен 0, то курсор устанавливается в начало строки.
2. Нажать клавишу «влево». Тогда все курсоры сдвигаются налево. Если какой-то курсор находился в начале текста, то он там и остается. В случае, если курсор находится в начале какой-то строки, то он перемещается в конец предыдущей строки.
3. Нажать клавишу «вправо». Тогда все курсоры сдвигаются направо. Если какой-то курсор находился в конце текста, то он там и остается. В случае, если курсор находится в конце какой-то строки, то он перемещается в начало следующей строки.
4. Нажать клавишу «стереть». Тогда все курсоры стирают тот символ, после которого они находятся. Если курсор находился в начале текста, то он ничего не делает. Если курсор находился в начале строки, то вся строка переходит в конец предыдущей строки.

В данной задаче, для упрощения, будем считать, что новые символы дописывать нельзя.

Обратите внимание, что после некоторых операций, курсоры могут иметь одинаковую позицию в тексте. В таком случае, эти курсоры становятся неотличимы друг от друга и вам следует считать их за один курсор.

Вам дан исходный текст и некоторая последовательность операций. Необходимо определить, какой текст получится после выполнения данных операций.

### Формат входных данных

На первой строке вводится число  $n$ , ( $1 \leq n \leq 1000$ ) — количество строк в исходном тексте.

Далее вводится  $n$  строк — исходный текст. Гарантируется, что никакая строка не пустая. В конце текста нет пустых строк. Строки разделяются символом `'\n'` (ASCII код – 10). Обратите внимание на ограничения в системе оценки. Гарантируется, что суммарный размер текста не превосходит 1000 символов.

Следующая строка содержит число  $q$ , ( $1 \leq q \leq 1000$ ) — количество запросов.

Каждая из следующих строк содержит одну из команд:

1. « + ». После неё идет два числа —  $l$  и  $i$ . Команда означает добавление курсора как описано в условии.
2. « < ». Сдвиг влево.
3. « > ». Сдвиг вправо.
4. « - ». Команда стереть.

### Формат выходных данных

Выведите текст, который будет получен после применения всех операций. Обратите внимание, что некоторые строки могут остаться пустыми. Строки должны разделяются символом `'\n'`, другие разделители не допускаются.



## Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
0	Тесты из условия	0	
1	Только один курсор, нет запросов сдвига. Размер текста $\leq 10, q \leq 10$	6	
2	Один курсор. Размер текста $\leq 100, q \leq 100$	10	1
3	Нет запросов сдвига. Размер текста $\leq 100, q \leq 100$	7	1
4	Все запросы добавления идут до запросов 'стереть'. Размер текста $\leq 100, q \leq 100$	10	
5	Размер текста $\leq 10, q \leq 10$	13	1
6	Размер текста $\leq 100, q \leq 100$	14	1 - 5
7	Размер текста $\leq 1000, q \leq 1000$	40	6

## Примеры

стандартный ввод	стандартный вывод
3 abc de fgh 4 - + 2 2 - -	abc  fgh
3 a bcd efghi 7 + 3 4 + 2 0 < < - - -	a bcdghi

## Замечание

Входные данные не запрещают нажатия клавиш «влево», «вправо» или «стереть» если в тексте нет курсоров. В таком случае просто ничего не нужно делать.