

## Условия задач 9-11 класс

Решением для задач является консольное приложение!

Данные можно считывать с консоли или из текстового файла `input.txt`. Результат можно выводить на консоль или в текстовый файл `output.txt`.

Каждая строка во входных данных завершается символом конца строки. Последняя строка во входном файле пустая.

### Система оценивания

Если не пройдены все тесты из условия, то баллы за задачу не начисляются.

Баллы начисляются за каждый пройденный тест (кроме тестов из условия). Если все тесты для данной подзадачи пройдены, то начисляется указанное для этой подзадачи количество баллов. Если тесты пройдены не все, то начисляется меньшее количество баллов. Решение получает максимальное количество баллов в случае, когда все тесты будут пройдены.

Максимальное время тестирования всех задач на любом тесте – 1 секунда (если в условии не оговорено иное).

### 1. «Прогулка по магазинам» (100 баллов)

В городе, где живет Галя, вдоль одной прямой улицы расположено  $N$  самых интересных и посещаемых магазинов. Ей известно, что первый магазин находится на расстоянии  $a_0$  метров от начала улицы, а остальные магазины находятся на расстоянии  $a_1, \dots, a_{N-1}$  метров друг от друга, последний магазин находится на расстоянии  $a_N$  метров от конца улицы. В выходной день Галя решила прогуляться по этой улице и посетить магазины. Но у Гали нет много времени гулять, ей еще нужно делать уроки. Она хочет выйти сразу к одному из магазинов на улице, посетить  $K$  магазинов, расположенных подряд, и пойти домой. Помогите Гале определить, какое наименьшее количество метров ей придется пройти по улице с магазинами.

#### Входные данные:

В первой строке записано одно натуральное число  $N$  – количество магазинов ( $1 \leq N \leq 10^5$ ).

В следующих  $N+1$  строках записано по одному натуральному числу  $a_i$  ( $a_0$  – расстояние от начала улицы до первого магазина,  $a_1, \dots, a_{N-1}$  – расстояния между соседними магазинами,  $a_N$  – расстояние от последнего магазина до конца улицы ( $1 \leq a_i \leq 10^3$ )).

В последней строке записано одно натуральное число  $K$  – количество магазинов, которое Галя хочет посетить ( $2 \leq K \leq N$ ).

#### Выходные данные:

Одно целое число – минимальное количество метров, которое Гале нужно пройти по улице с магазинами. Обратите внимание, что расстояние, которое она пройдет от дома до улицы, а потом от улицы до дома, учитывать не нужно.

#### Пример:

Входные данные	Выходные данные
4 10 27 67 35 20 3	94

#### Описание подзадач:

Подзадача	Баллы	Дополнительные ограничения
1	60	$1 \leq N \leq 10^3$
2	20	$1 \leq N \leq 10^4, 1 \leq K \leq 10^2$
3	20	Ограничения из условия

## 2. «Букеты в подарок» (100 баллов)

В цветочном магазине продаются  $N$  штук роз и  $M$  штук хризантем. Дима хочет составить очень красивый букет для своей девушки. Он знает, что у девушки любимые числа  $a$  и  $b$ , поэтому он хочет, чтобы в букете количество цветов одного вида было кратно одному из чисел  $a$  или  $b$ , а количество цветов другого вида было кратно другому числу из данной пары. Также Дима обязательно хочет, чтобы в букете присутствовали и розы, и хризантемы. Поскольку букет предназначен в подарок, то в нем должно быть нечетное количество цветов. Конечно, чтобы удивить девушку, букет должен оказаться максимально большим. Помогите Диме определить, сколько роз и сколько хризантем будет в букете.

### Входные данные:

В первой строке записано одно натуральное число  $N$  – количество роз, которое продается в магазине ( $1 \leq N \leq 10^9$ ).

Во второй строке записано одно натуральное число  $M$  – количество хризантем, которое продается в магазине ( $1 \leq M \leq 10^9$ ).

В третьей строке записано одно натуральное число  $a$  – первое любимое число девушки ( $1 \leq a \leq 10^9$ ).

В четвертой строке записано одно натуральное число  $b$  – второе любимое число девушки ( $1 \leq b \leq 10^9$ ).

### Выходные данные:

Два натуральных числа через пробел: первое число – количество роз в букете, второе число – количество хризантем в букете.

Если подходящих вариантов составить букетов несколько, то нужно вывести один любой вариант.

Если букет составить нельзя, то нужно вывести пару 0 0.

### Пример:

Входные данные	Выходные данные
24 12 2 3	21 12
3 6 2 4	0 0

### Описание подзадач:

Подзадача	Баллы	Дополнительные ограничения
1	10	$N, M$ кратны числам $a$ и $b$
2	50	$N, M \leq 10^3$
3	40	Ограничения из условия

## 3. «Проверка последовательности» (100 баллов)

Алексею дали в школе задание: запомнить квадраты всех однозначных чисел. Но сидеть и зубрить табличку это так скучно. Он решил попрактиковаться не только в умении возводить в квадрат, но еще и в сложении.

Алексей стал записывать последовательность цифр по следующему правилу. Он взял две цифры (это первые два элемента последовательности  $a$  и  $b$ ) и нашел сумму их квадратов. У полученной суммы оставил только последнюю цифру. Таким образом он получил третий элемент последовательности. Чтобы получить следующие элементы, нужно взять два предыдущих элемента, найти сумму их квадратов и записать последнюю цифру полученной суммы.

Алексею очень понравилось находить значения таким образом. Он так увлекся, что исписал несколько листов бумаги, но потом остановился и задумался, а не ошибся ли он где-то. Пересчитывать еще раз ему совсем не хочется, поэтому он просит вас помочь и написать программу, которая по заданным двум первым элементам последовательности  $a$  и  $b$  будет находить цифру, которая стоит на месте с номером  $N$  в полученной им последовательности.

### Входные данные:

В первой строке записано натуральное число  $a$  – первый элемент последовательности ( $0 \leq a \leq 9$ ).

В второй строке записано натуральное число  $b$  – второй элемент последовательности ( $0 \leq b \leq 9$ ).

В третьей строке записано натуральное число  $N$  – номер элемента последовательности, который нужно найти ( $3 \leq N \leq 10^{18}$ ).

**Выходные данные:**

Одно целое число –элемент последовательности, который стоит на месте с номером  $N$ .

**Примеры:**

Входные данные	Выходные данные
3 5 3	4
8 5 6	5

**Пояснение к примерам**

В первом примере последовательность будет такой: 3, 5, 4.

В втором примере последовательность будет такой: 8, 5, 9, 6, 7, 5.

**Обратите внимание, что в данной задаче необходимо использовать 64-битный тип данных, например longlong в C++, int64 в Free Pascal, long в Java.**

**Описание подзадач:**

Подзадача	Баллы	Дополнительные ограничения
1	60	$1 \leq N \leq 10^6$
2	20	$1 \leq N \leq 10^9$
3	20	Ограничения из условия

**4. «Яся продаёт рыбок»(100 баллов)**

Максимальное время тестирования на одном тесте – 2 с

Яся решила заняться крупным бизнесом, поэтому начала продавать рыбок. Рыбок у неё неограниченное количество, в то время как продавцов всего  $n$  человек. Продавец с номером  $i$  обладает пропускной способностью  $a_i$ , то есть способен обслуживать  $a_i$  покупателей одновременно.

Всего за рыбками придут  $m$  покупателей. Первый покупатель придёт в секунду 1, второй – в секунду 2 и так далее. Яся должна назначить каждому  $j$ -тому покупателю ( $1 \leq j \leq m$ ) некоторого продавца  $c_j$ , который в данный момент обслуживает **строго меньше**  $a_{c_j}$  покупателей. Этот покупатель будет покупать рыбку у выбранного продавца **ровно**  $k$  секунд, после чего уйдёт домой.

Яся знает, сколько покупателей к ней придут. Помогите ей избавиться от **максимального** количества ненужных продавцов и распределить покупателей между оставшимися так, чтобы каждый смог купить рыбку.

Если решения не существует, выведите -1.

**Входные данные:**

В первой строке записаны через пробел целые числа  $n, m, k$  – количество продавцов, количество покупателей и время, которое тратит каждый покупатель на покупку рыбки ( $1 \leq n, m, k \leq 2 \cdot 10^5$ ).

Во второй строке записаны через пробел  $n$  целых чисел  $a_1, a_2, \dots, a_n$  – пропускные способности продавцов ( $1 \leq a_i \leq 2 \cdot 10^5$ ).

**Выходные данные:**

Если решения не существует, вывести -1.

В противном случае:

В первой строке вывести одно число  $p$  – **максимальное** количество продавцов, от которых Яся избавится ( $0 \leq p < n$ ).

В следующей строке вывести  $p$  попарно различных целых чисел через пробел:  $b_1, b_2, \dots, b_p$  – номера продавцов, от которых Яся избавится ( $1 \leq b_i \leq n$ ).

В следующей строке вывести  $m$  целых чисел через пробел:  $c_1, c_2, \dots, c_m$  – номера продавцов, к которым Яся отправит соответствующих покупателей ( $1 \leq c_j \leq n$ ).

Если решений несколько, можно вывести любое.

### Пример:

Входные данные	Выходные данные
10 10 5	8
2 1 3 3 1 1 2 1 2 2	1 2 5 6 7 8 9 10
	3 3 3 4 4 3 3 3 4 4

### Примечание

Если вы сдаёте решение на **Python**, то

- сдавайте решение на PyPy;
- вставьте в начало файла `from sys import stdin`;
- используйте `stdin.readline()` вместо `input()`;
- минимизируйте количество вызовов `print`.

Если вы сдаёте решение на **Pascal**, то сдавайте решение на Free Pascal.

Если вы сдаёте решение на **C++**, то

- используйте `\n` вместо `std::endl`
- В самом начале функции `main` пропишите строчку  
`std::ios_base::sync_with_stdio(false);`  
`std::cin.tie(nullptr);`

Это ускорит ввод/вывод чисел.

### Описание подзадач

Подзадача	Баллы	Дополнительные ограничения
1	25	$1 \leq n, m, k \leq 15$
2	10	$n = 1$
3	20	$a_i = 1$
4	45	Ограничения из условия

## 5. «Горилла и скобочная последовательность» (100 баллов)

Максимальное время тестирования на одном тесте – 3 с

Однажды одна дружелюбная горилла нашла в джунглях скобочную последовательность  $s$  длины  $n$ . Эта скобочная последовательность ей очень понравилась, поэтому она решила с ней поиграть.

Игра состоит из  $m$  запросов. Каждый запрос задаёт подотрезок  $s_l, s_{l+1}, \dots, s_r$  скобочной последовательности  $s$ . Выполнение этого запроса означает замену всех скобок с данного подотрезка на противоположные, то есть '(' заменяется на ')', а ')' заменяется на '(' . Например, если строка равна «( ) ) ( ( ( ) )», после выполнения запроса  $l = 4, r = 8$ , строка превратится в «( ) ( ( ) ) ( )».

Чтобы ответить на запрос, горилла должна сначала произвести вышеописанную замену, а затем сообщить, какое **минимальное** количество скобок надо вставить в последовательность  $s$ , чтобы она стала *почти правильной*. Горилла вольна вставлять любые скобки в любые позиции.

*Почти правильной* скобочной последовательностью называется скобочная последовательность, в которой количество открывающихся и закрывающихся скобок совпадает. Например, скобочные последовательности «( ) ( ( ) )», «( ) ( )» и «( ) ( ) ( )» являются *почти правильными*, а скобочные последовательности «( ) ( ( )» и «( ) ( ) ( )» – нет.

Обратите внимание, что изменения сохраняются, то есть следующий запрос применяется уже к изменённой последовательности. В то же время, при ответе на запрос горилла не вставляет скобки, а только сообщает их количество.

### Входные данные

В первой строке записаны через пробел два целых числа  $n, m$  – длина скобочной последовательности и количество запросов ( $1 \leq n, m \leq 2 \cdot 10^5$ ).

Во второй строке записана скобочная последовательность  $s$  длины  $n$ , состоящая из символов '(' и ')'

В последних  $m$  строках записаны через пробел по два целых числа  $l, r$  – описание запросов ( $1 \leq l \leq r \leq n$ ).

### Выходные данные

$m$  целых чисел, каждое в отдельной строке – ответы на запросы.

### Пример:

Входные данные	Выходные данные
10 10	0
) ( ) ) ( ( ( )	2
7 7	0
4 8	4
3 9	0
6 7	4
1 6	4
4 9	6
7 10	8
8 8	2
10 10	
3 7	

### Примечание

Если вы сдаёте решение на **Python**, то

- сдавайте решение на PyPy;
- вставьте в начало файла `from sys import stdin`;
- используйте `stdin.readline()` вместо `input()`;
- минимизируйте количество вызовов `print`.

Если вы сдаёте решение на **Pascal**, то сдавайте решение на Free Pascal.

Если вы сдаёте решение на **C++**, то

- используйте `\n` вместо `std::endl`
- В самом начале функции `main` пропишите строчку  
`std::ios_base::sync_with_stdio(false);`  
`std::cin.tie(nullptr);`

Это ускорит ввод/вывод чисел.

### Описание подзадач

Подзадача	Баллы	Дополнительные ограничения
1	12	$1 \leq n, m \leq 250$
2	20	$1 \leq n \cdot m \leq 10^6$
3	8	$\min(n, m) = 1$
4	20	$l = r$
5	40	Ограничения из условия