

## Общая информация по задачам второго тура

Задача	Тип задачи	Ограничения
5. Гирлянда	Стандартная	2 с, 512 МБ
6. Классные парты	Стандартная	2 с, 512 МБ
7. Робогольф	Стандартная	3 с, 512 МБ
8. Поиск идеи	Стандартная	5 с, 1024 МБ

Необходимо считывать данные из стандартного потока ввода. Выходные данные необходимо выводить в стандартный поток вывода.

Во всех задачах баллы за подзадачу начисляются, только если все тесты для подзадачи пройдены.

## Задача 5. Гирлянда

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Гирлянда — это цепочка из элементов: флажков и шариков, содержащая хотя бы один флажок. Длинной гирлянды называется количество элементов, из которых она состоит.

Гирлянда считается красивой, если для каждого флажка количество подряд идущих шариков слева от него до ближайшего слева флажка или до начала гирлянды равно количеству подряд идущих шариков справа от него до ближайшего справа флажка или до конца гирлянды.

Обозначим шарик символом «0», а флажок — символом «1». Например, гирлянда «0001000» является красивой, а гирлянда «001010» — нет, поскольку слева от первого флажка два шарика, а справа от него — один шарик. Отметим, что цепочка «000» не является гирляндой, так как не содержит ни одного флажка.

Задана гирлянда, разрешается удалить некоторые её элементы.

Требуется написать программу, находящую по описанию гирлянды самую длинную красивую гирлянду, которую можно получить из заданной удалением элементов.

### Формат входных данных

Первая строка содержит целое число  $n$  — количество элементов в исходной гирлянде ( $1 \leq n \leq 500\,000$ ).

Вторая строка содержит описание гирлянды в виде строки из  $n$  символов «0» и «1». Строка содержит хотя бы один символ «1».

### Формат выходных данных

В первой строке выведите число  $m$  — длину получившейся красивой гирлянды ( $1 \leq m \leq n$ ).

Во второй строке выведите получившуюся красивую гирлянду.

Если существует несколько решений, выведите любое из них.

### Система оценки

Обозначим через  $k$  количество флажков в исходной гирлянде.

Подз.	Баллы	Ограничения		Необх. подзадачи	Результаты во время тура
		$n$	$k$		
1	20	$n \leq 15$		У	потестовые
2	20	$n \leq 1000$	$k \leq 2$	—	потестовые
3	20	$n \leq 1000$	$k \leq 15$	У, 1, 2	потестовые
4	16	$n \leq 1000$		У, 1–3	потестовые
5	10	$n \leq 100\,000$	$k \leq 50$	У, 1–3	первая ошибка
6	14	$n \leq 500\,000$		У, 1–5	первая ошибка

### Примеры

стандартный ввод	стандартный вывод
10 0100100000	7 0001000
3 111	3 111
7 0100101	5 01010

## Задача 6. Классные парты

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Для нового кабинета школы Иннополиса требуется купить  $n$  двухместных парт.

Парты бывают  $k$  типов, которые задают их размер. Парта типа  $i$  подходит школьникам, рост которых находится в диапазоне от  $L_i$  до  $R_i$  включительно. Остальным школьникам сидеть за такой партой неудобно, при этом *величиной неудобства* школьника, если он сидит за такой партой, будем называть модуль разности его роста и ближайшей границы диапазона этой парты. Если парта школьнику подходит, то для него величина неудобства равна нулю.

Например, если  $L_i = 100$  и  $R_i = 120$ , то неудобство для школьника с ростом 80 равно 20, для школьника с ростом 130 равно 10, а для школьника с ростом 105 равно 0.

В кабинете по очереди будут заниматься  $m$  групп школьников, каждая из которых состоит из  $2n$  человек. Известен рост каждого школьника в каждой из групп. Закупленные парты будут расставлены в классе, и в каждой группе за каждой партой будут сидеть ровно два школьника. Необходимо купить  $n$  парт и рассадить за ними школьников каждой группы таким образом, чтобы суммарное неудобство для всех школьников, занимающихся в этом кабинете, было минимальным.

Требуется написать программу, которая по информации о каждом из  $k$  типов парт и известным значениям роста каждого школьника в каждой группе определяет, какого минимального суммарного значения неудобства школьников можно достичь, купив парты и рассадив за них школьников в каждой группе оптимальным образом.

### Формат входных данных

В первой строке входных данных находятся три целых числа  $m$ ,  $n$  и  $k$  ( $1 \leq m, n \leq 200\,000$ ;  $1 \leq m \cdot n \leq 200\,000$ ;  $2 \leq k \leq 200\,000$ ) — количество групп школьников, которые будут заниматься в кабинете, количество парт, которые необходимо купить, и количество типов парт соответственно.

В каждой из следующих  $k$  строк находятся по два целых числа  $L_i$  и  $R_i$  ( $1 \leq L_i \leq R_i \leq 10^9$ ), характеризующие диапазон роста школьников, для которых подходят парты типа  $i$ .

В каждой из следующих  $m$  строк находится описание группы. Каждое описание состоит из  $2n$  целых чисел  $h_1, h_2, \dots, h_{2n}$ , задающих значение роста каждого из  $2n$  школьников группы ( $1 \leq h_i \leq 10^9$ ).

### Формат выходных данных

В единственной строке выходных данных выведите  $P$  — минимальную величину суммарного неудобства, которую можно достичь при оптимальной покупке парт.

## Система оценки

Подз.	Баллы	Ограничения				Необх. подзадачи	Результаты во время тура
		$m$	$n$	$k$	Доп. огр.		
1	10	$m \leq 100$	$n = 1$	$k \leq 50$	—	—	потестовые
2	10	$m = 1$	$n \leq 1000$	$k \leq 50$	—	—	потестовые
3	10	$m \leq 50$	$n \leq 5$	$k \leq 3$	—	—	потестовые
4	10	$m \leq 100$	$n \leq 1000$	$k = 2$	—	—	потестовые
5	10	$m \leq 100$	$n \leq 1000$	$k \leq 3$	—	3, 4	потестовые
6	10	$m \leq 100$	$n \leq 1000$	$k \leq 50$	$L_i = R_i$	—	первая ошибка
7	10	$m \leq 100$	$n \leq 1000$	$k \leq 50$	—	У, 1–6	первая ошибка
8	8				$L_i = R_i$	6	первая ошибка
9	8	$m \leq 100$			—	У, 1–7	первая ошибка
10	10		$n \leq 100$		—	У, 1, 3	первая ошибка
11	4				—	У, 1–10	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
1 2 2 5 25 50 90 60 5 10 40	10
2 3 3 200 400 300 500 100 600 300 330 440 40 30 300 150 250 350 450 550 300	130
1 3 4 10 100 200 200 10 100 300 1000 5 10 20 15 200 90	105

## Пояснение к примеру

В первом примере есть только одна группа школьников, занимающаяся в классе. Следует купить по одной парте каждого вида и посадить школьников с ростами 5 и 10 за парту первого типа, а школьников с ростами 40 и 60 за парту второго типа. В таком случае неудобно сидеть будет только школьнику с ростом 40 и соответствующая величина неудобства будет равна 10.

## Задача 7. Робогольф

Ограничение по времени: 3 секунды  
Ограничение по памяти: 512 мегабайт

На всемирной олимпиаде роботов проводятся соревнования по робоголфу. Поле, на котором происходит игра, имеет вид прямоугольника, состоящего из  $n \times m$  единичных квадратов. Строки поля пронумерованы числами от 1 до  $n$ , а столбцы — от 1 до  $m$ . Таким образом, каждый квадрат характеризуется двумя положительными целыми числами  $r$  и  $c$  — номерами строки и столбца, на пересечении которых он находится.

Два робота по очереди перемещают специальную фишку по полю, в некоторых квадратах которого находятся ловушки. Если фишка находится в квадрате  $(r, c)$ , то робот, выполняющий очередной ход, может переместить её в квадрат  $(r + 1, c)$  или в квадрат  $(r, c + 1)$ . Если соответствующего квадрата не оказалось, поскольку фишка находится в последней строке или в последнем столбце поля, робот может переместить её за границу поля. Игра заканчивается в том случае, когда фишка оказывается либо за границей поля, либо в квадрате с ловушкой.

Каждой ловушке соответствует некоторое целое число  $v_i$  — её цена. Результат игры равен значению цены ловушки, в которой закончилась игра, или равен 0, если фишка оказалась за границей поля. Робот, который делает ход первым, стремится минимизировать результат игры, а робот, который делает ход вторым — максимизировать.

Пусть фишка вначале находится в квадрате  $(r, c)$ . Гарантированный результат игры  $g(r, c)$  для этого квадрата равен минимальному возможному результату игры, которого может добиться делающий первый ход робот вне зависимости от действий соперника. Поскольку стартовый квадрат до начала матча неизвестен, разработчики роботов хотят вычислить сумму значений  $g(r, c)$  для всех квадратов поля.

Требуется написать программу, которая по заданным расположениям и ценам ловушек определит сумму гарантированных результатов игры по всем квадратам поля.

### Формат входных данных

Первая строка входных данных содержит три целых числа  $n$ ,  $m$  и  $k$  ( $1 \leq n, m \leq 10^9$ ;  $1 \leq k \leq \min(n \cdot m, 10^5)$ ) — количество строк, количество столбцов и количество ловушек, соответственно.

Следующие  $k$  строк содержат по три целых числа  $r_i$ ,  $c_i$  и  $v_i$  ( $1 \leq r_i \leq n$ ,  $1 \leq c_i \leq m$ ,  $-10^9 \leq v_i \leq 10^9$ ) — номер строки, номер столбца и цену  $i$ -й ловушки соответственно. Никакие две ловушки не находятся в одном и том же квадрате.

### Формат выходных данных

Выведите одно целое число — остаток от деления суммы гарантированных результатов игры по всем квадратам поля на число 998 244 353.

Остаток от деления  $a$  на  $b$ , где  $a$  может быть отрицательным, равен числу  $r = a \bmod b$ , лежащему в диапазоне от 0 до  $b-1$ , такому что  $a = qb+r$ , где  $q$  — целое. Например,  $13 \bmod 4 = 1$ ,  $-13 \bmod 4 = 3$ ,  $12 \bmod 4 = 0$ ,  $-12 \bmod 4 = 0$ .

### Примеры

стандартный ввод	стандартный вывод
3 3 3 2 3 -2 3 1 3 1 2 1	998244352
2 4 3 1 2 2 2 4 -3 2 1 1	998244348

## Пояснение к примеру

В первом примере гарантированные результаты игр для всех квадратов выглядят так (квадраты с ловушками выделены):

	1	2	3
1	1	1	-2
2	0	-2	-2
3	3	0	0

Сумма результатов равна:  $1 + 1 - 2 + 0 - 2 - 2 + 3 + 0 + 0 = -1$ . Ответ равен  $(-1) \bmod 998\,244\,353 = -1 + 998\,244\,353 = 998\,144\,352$ .

В втором примере гарантированные результаты игр для всех квадратов выглядят так (квадраты с ловушками выделены):

	1	2	3	4
1	1	2	0	-3
2	1	0	-3	-3

Сумма результатов равна:  $1 + 2 + 0 - 3 + 1 + 0 - 3 - 3 = -5$ . Ответ равен  $(-5) \bmod 998\,244\,353 = 998\,244\,348$ .

## Система оценки

Подз.	Баллы	Ограничения			Необх. подзадачи	Результаты во время тура
		$n, m$	$k$	дополнительно		
1	20	$n, m \leq 1\,000$			У	первая ошибка
2	14	$n \leq 5, m \leq 10^9$			У	первая ошибка
3	14	$n, m \leq 100\,000$	$k = n + m - 1$	для любого $i$ , $r_i = n$ или $c_i = m$	—	первая ошибка
4	10			для любого $i$ , $r_i \geq n - 1\,000$ и $c_i \geq m - 1\,000$	У, 1	первая ошибка
5	6	$n, m \leq 100\,000$		для любого $i$ , $v_i = 1$	—	первая ошибка
6	6			для любого $i$ , $v_i = 1$	5	первая ошибка
7	10	$n, m \leq 100\,000$			У, 1, 3, 5	первая ошибка
8	10		$k \leq 1\,000$		У	первая ошибка
9	10				У, 1–8	первая ошибка

## Задача 8. Поиск идеи

Ограничение по времени: 5 секунд  
Ограничение по памяти: 1024 мегабайта

3019 год. При раскопках древнего города Иннополис археологи обнаружили артефакт — жёсткий диск, на котором находится файл, предположительно содержащий тексты всех задач всероссийских олимпиад по информатике.

После исследования файла было выяснено, что информация в нём закодирована таким образом, что записанный в файле текст представляет собой строку  $t$  из букв английского алфавита. Текст с задачами оказался довольно длинным и содержал много повторений, поэтому файл хранился на диске в сжатом виде. Для его распаковки используется следующий алгоритм.

В процессе распаковки формируется строка  $t$  из строчных букв английского алфавита. Исходно строка пуста. Сжатый файл состоит из  $n$  блоков, которые должны быть обработаны в порядке следования. Каждый блок имеет один из двух типов.

- Блок «1  $w$ », где  $w$  — строка. При обработке такого блока в конец строки  $t$  дописывается строка  $w$ .
- Блок «2  $pos\ len$ », где  $pos$  и  $len$  — положительные целые числа. Пусть символы строки  $t$  пронумерованы с 1. При обработке такого блока в конец строки  $t$  по очереди приписываются  $len$  подряд идущих символов строки  $t$ , начиная с позиции  $pos$ . При этом, если значение  $len$  достаточно велико, некоторые только что выписанные символы могут быть снова использованы при обработке того же блока.

Ученые решили выяснить, сколько раз некоторая идея встречалась в олимпиадах. Для этого они сформировали строку  $p$  из строчных букв английского алфавита и хотят найти количество вхождений строки  $p$  как подстроки в полученную после распаковки файла строку  $t$ .

Строка  $p$  длины  $m$  входит в строку  $t$  как подстрока с позиции  $i$ , если  $m$  следующих подряд символов строки  $t$ , начиная с  $i$ -го, представляют собой строку  $p$ . Например, строка «aba» входит как подстрока в строку «ababaaba» три раза: с позиций 1, 3 и 6.

Требуется написать программу, которая определяет количество вхождений заданной строки  $p$  в полученную после распаковки файла строку  $t$ .

### Формат входных данных

В первой строке находятся натуральные числа  $m$  и  $n$  — длина строки  $p$  и количество блоков в сжатом тексте ( $1 \leq m \leq 2 \cdot 10^5$ ,  $1 \leq n \leq 10^4$ ).

Во второй строке входных данных задана непустая строка  $p$ , состоящая из строчных букв английского алфавита.

В следующих  $n$  строках находятся описания блоков в описанном в условии формате. Для блоков первого типа приписываемая строка  $w$  непуста, сумма длин всех строк  $w$  в блоках первого типа не превышает  $2 \cdot 10^5$ . Для блоков второго типа в строке  $t$  к моменту обработки этого блока находится не менее  $pos$  символов. Длина распакованного текста не превышает  $10^{15}$  символов.

### Формат выходных данных

Выведите одно целое число — количество вхождений строки  $p$  в текст.

### Пример

стандартный ввод	стандартный вывод
3 4 aba 1 ab 2 1 3 2 3 3 2 1 8	6

## Пояснение к примеру

При распаковке файла в примере последовательно получаются следующие строки:

«» → «ab» → «ababa» → «ababaaba» → «ababaabaababaaba».

Строка «aba» входит как подстрока в результирующую строку «ababaabaababaaba» 6 раз.

## Система оценки

Обозначим длину текста  $t$  после обработки  $i$  блоков как  $L_i$ .

Обозначим тип  $i$ -го блока как  $type_i$ . Если  $type_i = 2$ , то обозначим параметры этого блока как  $pos_i$  и  $len_i$ .

Подз.	Баллы	Ограничения			Необх. подзадачи	Результаты во время тура
		$m, n$	$L_n$	дополнительно		
1	6	$m \leq 2000$ $n = 1$	$L_n \leq 1000$		—	первая ошибка
2	10	$m \leq 10^5$ $n \leq 2000$	$L_n \leq 10^6$		У, 1	первая ошибка
3	10	$m \leq 2000$ $n \leq 2000$	$L_n \leq 10^{10}$	кроме первого блока $type_i = 2, pos_i = 1$ $len_i$ делится на $L_1$	1	первая ошибка
4	10	$m \leq 2000$ $n \leq 2000$	$L_n \leq 10^{10}$	$pos_i = L_{i-1}$	1	первая ошибка
5	10	$m \leq 20$ $n \leq 2000$	$L_n \leq 10^{10}$	$pos_i = 1, len_i \leq 10^7$	—	первая ошибка
6	4	$m \leq 2000$ $n \leq 2000$	$L_n \leq 10^{10}$	$pos_i = 1, len_i \leq 10^7$	1, 5	первая ошибка
7	10	$m \leq 20$ $n \leq 20$	$L_n \leq 10^{10}$	$p$ состоит из букв «а» $pos_i + len_i - 1 \leq L_{i-1}$	—	первая ошибка
8	6	$m \leq 20$ $n \leq 20$	$L_n \leq 10^{10}$	$pos_i + len_i - 1 \leq L_{i-1}$	7	первая ошибка
9	2	$m = 1$ $n \leq 2000$	$L_n \leq 10^{10}$	$p$ состоит из букв «а»	—	первая ошибка
10	4	$m \leq 20$ $n \leq 2000$	$L_n \leq 10^{10}$	$p$ состоит из букв «а»	7–9	первая ошибка
11	5	$m \leq 20$ $n \leq 2000$	$L_n \leq 10^{10}$		У, 5, 7–10	первая ошибка
12	14	$m \leq 10^5$ $n \leq 2000$	$L_n \leq 10^{10}$		У, 1–11	первая ошибка
13	9	$m \leq 2 \cdot 10^5$ $n \leq 10000$	$L_n \leq 10^{15}$		У, 1–12	первая ошибка