

Задача А. Отель в Олимп-Сити

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вы — сотрудник компании «Interplanetary Software, Inc.». Недавно Вы поехали в командировку в Олимп-Сити, один из крупнейших городов Марса. По приезду в город Вы заселились в одном из многочисленных отелей города.

В отеле всего n этажей, на каждом из которых находится m окон. Таким образом, его можно представить как прямоугольную матрицу размером $n \times m$. Отель разделен на комнаты, каждая из которых находится ровно на одном этаже и содержит одно или более соседних окон. То есть, каждая комната представляет собой некоторый отрезок на этой матрице. Известно также, что каждое окно принадлежит ровно одной комнате.

По вечерам Вам понравилось смотреть на отель, наблюдая, как в его окнах включается и выключается свет. Всего Вы провели k наблюдений, причем для каждого наблюдения Вам известно про каждое окно, включен в нем свет или нет.

Вы понимаете, что свет либо горит во всех окнах комнаты, либо не горит. Иными словами, не может быть такого, что в одном окне комнаты свет горит, а в другом — нет.

По результатам наблюдений Вы хотите узнать: а какое наименьшее количество комнат может быть в отеле? Так как у Вас скопилось много результатов наблюдений, то Вы решили написать программу, чтобы выяснить ответ на этот вопрос.

Формат входных данных

В первой строке входных данных находится три целых числа k , n и m ($1 \leq k \cdot n \cdot m \leq 10^6$) — количество наблюдений, количество этажей в отеле и количество окон на каждом этаже.

Далее во входных данных находится k блоков — описание наблюдений.

Каждый блок состоит из n строк по m символов каждая. Если j -й символ i -й строки равен «0», то в j -м окне i -го этажа не горит свет, а если он равен «1», то свет горит.

Формат выходных данных

Выведите одно целое число — минимально возможное количество комнат в отеле.

Система оценки

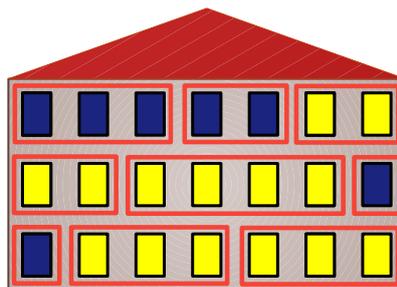
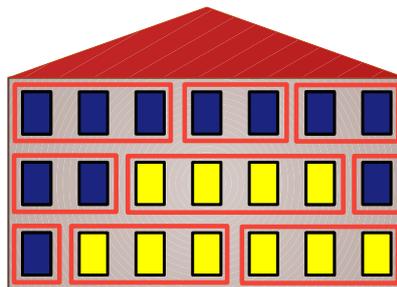
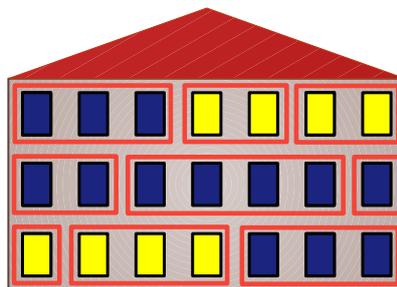
№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$m = 1$	9	
2	$m \leq 2, k = 1$	5	
3	$m \leq 2$	7	1 - 2
4	$k = 1, n \cdot m = 1000$	21	
5	$k = 1$	15	2, 4
6	$k \cdot n \cdot m = 1000$	17	4
7	Нет дополнительных ограничений	26	1 - 6

Примеры

стандартный ввод	стандартный вывод
1 5 8 01010101 10101010 01010101 10101010 01010101	40
1 1 10 1111111111	1
3 3 7 0001111 0000000 1111000 0000000 0011110 0111111 0000011 1111110 0111111	9

Замечание

Рассмотрим третий пример из условия. Оптимальное разбиение на девять комнат выглядит следующим образом:



Можно показать, что на меньшее число комнат разбить отель невозможно.

Задача В. Игра с цифрами

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

Петя и его друг, робот Petya++, любят играть в различные игры. Одна из любимых забав у ребят — выписывать на доске последовательности из цифр и проделывать над ними всякие операции.

Сегодня Петя выписал на доске n двоичных цифр и предложил Petya++ проделать q операций следующего вида:

1. Выбрать отрезок $[l; r]$ и заменить все цифры на позициях с l по r на противоположные, т. е. заменить все нули на единицы и наоборот.
2. Выполнить операцию *смешивания* k раз на отрезке $[l; r]$. Операция смешивания выглядит следующим образом. Пусть a — позиция самого левого нуля в отрезке, а b — позиция самой правой единицы в отрезке, а w — длина отрезка (т. е. $w = r - l + 1$). Если отрезок состоит целиком из нулей, то $b = -\infty$, а если целиком из единиц — то $a = \infty$. Далее, если $a \leq w - b + 1$, то мы заменяем ноль на a -й позиции на единицу, иначе заменяем единицу на b -й позиции на ноль. Например, строка 11010010 станет равна 11010000 после однократного применения операции смешивания, т. к. в этом случае $a = 3$ и $b = 7$, откуда $w - b + 1 = 2$ и, поскольку $a > w - b + 1$, то мы заменяем самую правую единицу на ноль. А строка 0000 превратится в 1000, т. к. в этом случае $a = 1$ и $w - b + 1 = \infty$, и тогда мы заменяем самый левый ноль на единицу.
3. Сообщить количество единиц на отрезке $[l; r]$.

Petya++ справился с этими операциями довольно быстро. Теперь Петя должен проверить, что его друг все правильно сделал. Для этого ему понадобится программа, которая умеет выполнять все вышеперечисленные операции.

Формат входных данных

В первой строке входных данных находится два целых числа n и q ($1 \leq n, q \leq 2 \cdot 10^5$) — количество цифр на доске и количество операций.

Во второй строке входных данных находится строка, состоящая из n цифр «0» и «1» — цифры, записанные на доске.

В каждой из следующих q строк расположено описание операции. Оно состоит из разделенных пробелами чисел. Первое число в строке t ($1 \leq t \leq 3$) обозначает тип операции. В зависимости от типа операции, строка может иметь следующий вид:

- $1 \ l \ r$ ($1 \leq l \leq r \leq n$)
- $2 \ l \ r \ k$ ($1 \leq k \leq 10^9$, $1 \leq l \leq r \leq n$)
- $3 \ l \ r$ ($1 \leq l \leq r \leq n$)

Формат выходных данных

Для каждой операции третьего типа в отдельной строке выведите целое число — количество единиц на отрезке. Числа необходимо выводить в том же порядке, в котором операции расположены во входных данных.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n = 1, q \leq 300, k_i \leq 300$	2	
2	$n, q, k \leq 300$	19	1
3	$n, q \leq 3000$	12	1 - 2
4	$k = 1$	9	
5	не более одного события с $t = 2$	17	
6	$2 \leq t \leq 3$ и если $t = 2$, то $l = 1$ и $r = n$	14	
7	$n, q \leq 10^5$	11	1 - 3
8	Нет дополнительных ограничений	16	1 - 7

Пример

стандартный ввод	стандартный вывод
8 7	3
00111010	2
3 2 5	4
1 1 7	
2 1 8 3	
3 3 6	
1 1 2	
2 1 7 2	
3 1 8	

Замечание

Рассмотрим пример из условия.

После первой операции необходимо вывести 3, поскольку цифры на доске — 00111010.

После второй операции на доске записано 11000100.

Чтобы выполнить третью операцию, необходимо выполнить три смешивания на всей строке (т. е. на 11000100). После первого смешивания получится 11100100, поскольку $a = 3, w - b + 1 = 3$ и, раз $a = w - b + 1$, то надо заменять ноль на единицу. После второго смешивания мы заменим единицу на ноль и получим 11100000. После третьего смешивания мы снова заменим ноль на единицу и получим 11110000. Таким образом, после данной операции на доске будет записано 11110000.

На четвертой операции необходимо вывести 2.

После пятой операции на доске записано 00110000.

На шестой операции мы произведем два смешивания на отрезке $[1; 7]$, т. е. на строке 0011000. Каждое смешивание заменит самый левый ноль на единицу. Получится 1111000, т. е. в итоге на доске будет записано 11110000.

На седьмой операции необходимо вывести 4.

Задача С. Научные эксперименты

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

Как вы уже знаете, марсианские ученые экспериментируют с элементарными частицами, чтобы усовершенствовать ракетные двигатели для полета на дальние планеты.

Недавно их коллеги, исследователи с Венеры, открыли новый вид элементарных частиц — ϕ -нейтрино. Когда об открытии узнали на Марсе, то сразу же начались эксперименты с новыми частицами.

Один из экспериментов заключается в следующем. n частиц выстроены в ряд, i -я из которых имеет заряд c_i .

Ученые умеют выполнять операцию *нейтрализации* над частицами. Заключается она в следующем. Выберем l_1 и l_2 — позиции двух самых левых частиц с ненулевыми зарядами, причем $l_1 < l_2$. Среди двух частиц на этих позициях выберем ту, у которой заряд меньше и перенесем его заряд на другую частицу, т. е. заряд частицы с большим зарядом станет равен сумме зарядов, а заряд другой частицы станет равен нулю. Если заряды частиц равны, то тогда заряд переносится с более левой частицы на более правую. Данная операция применяется до тех пор, пока не останется одна частица с ненулевым зарядом.

Рассмотрим пример. Пусть заряды частиц равны 1 4 2 8 5 7, в порядке слева направо. Тогда нейтрализация частиц будет происходить следующим образом:

- Выбираем частицы на первой и второй позициях. Так как у частицы на второй позиции заряд больше, то на нее переносится заряд первой. Получаем заряды 0 5 2 8 5 7.
- Выбираем частицы на второй и третьей позициях. Так как у частицы на второй позиции заряд больше, то на нее переносится заряд третьей. Получаем заряды 0 7 0 8 5 7.
- Выбираем частицы на второй и четвертой позициях. Так как у частицы на четвертой позиции заряд больше, то на нее переносится заряд второй. Получаем заряды 0 0 0 15 5 7.
- Выбираем частицы на четвертой и пятой позиции. Получаем 0 0 0 20 0 7.
- Выбираем частицы на четвертой и шестой позиции. Получаем 0 0 0 27 0 0.
- Осталась ровно одна частица с ненулевым зарядом, поэтому окончательно последовательность частиц после нейтрализации выглядит как 0 0 0 27 0 0.

В ходе экспериментов будут выполнены следующие действия:

1. Заряд k -й частицы становится равным v .
2. Заряды всех частиц становятся равными нулю, кроме тех, которые расположены на отрезке от l до r включительно. После этого проводится нейтрализация и измеряется позиция частицы с ненулевым зарядом. После этого частицам возвращаются те заряды, которые у них были до начала данного действия.

Ученые хотят убедиться, что экспериментальные данные не расходятся с теорией, поэтому они попросили Вас написать программу, которая будет моделировать ход эксперимента.

Формат входных данных

В первой строке входных данных находится два целых числа n и q ($1 \leq n, q \leq 2 \cdot 10^5$) — количество частиц и количество действий.

Во второй строке входных данных находится n целых чисел c_i ($1 \leq c_i \leq 10^9$) — заряды частиц.

В каждой из следующих q строк расположено описание действия. Оно состоит из разделенных пробелами чисел. Сначала записано целое число t_i ($1 \leq t_i \leq 2$) — тип действия. Если $t_i = 1$, то далее записаны два целых числа k_i и v_i ($1 \leq k_i \leq n, 1 \leq v_i \leq 10^9$). Если $t_i = 2$, то далее записаны два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

Формат выходных данных

Для каждого действия второго типа выведите целое число — позиция заряженной частицы после нейтрализации. Числа необходимо выводить в том же порядке, в котором действия расположены во входных данных.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n, q \leq 300$	7	
2	$n, q \leq 3000$	9	1
3	$q = 1, t_i = 2, r_i = n$	13	
4	$c_i, v_i \leq 2$	8	
5	$c_i, v_i \leq 8$	6	4
6	$t_i = 2, r_i = n$	23	3
7	$t_i = 2$	12	3, 6
8	$n, q \leq 10^5$	9	1 - 2
9	Нет дополнительных ограничений	13	1 - 8

Пример

стандартный ввод	стандартный вывод
6 4	4
1 4 2 8 5 7	3
2 1 6	
1 3 4	
1 4 7	
2 2 4	

Замечание

Рассмотрим пример из условия.

Первое действие подробно разбирается выше. После нейтрализации заряды частиц равны 0 0 0 27 0 0, а частица с ненулевым зарядом находится на позиции 4, поэтому необходимо вывести 4. Обратите внимание, что по окончании данного действия

После второго и третьего действий заряды частиц равны 1 4 4 7 5 7.

Четвертое действие применяет нейтрализацию к последовательности частиц 0 4 4 7 0 0:

- Выбираем частицы на второй и третьей позициях. Так как у этих частиц заряды равны, то на заряд переносится на третью частицу. Получаем заряды 0 0 8 7 0 0.
- Выбираем частицы на третьей и четвертой позициях. Так как у частицы на третьей позиции заряд больше, то на нее переносится заряд четвертой. Получаем заряды 0 0 15 0 0 0.

В конце частица с ненулевым зарядом остается на позиции 3, поэтому необходимо вывести 3.

Задача D. Игра в слова

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Марсианская компания «Interplanetary Software, Inc.» решила выпустить новую игру в слова. Игра предназначена для жителей Плутона. Как известно, жители этой планеты обходятся всего двумя буквами, которые мы обозначим цифрами 0 и 1.

Заключается игра в следующем. Загадывается слово a из n букв. Игрок делает попытки отгадать это слово. Каждую попытку он должен называть перестановку p из n чисел. Напоминаем, что перестановкой из n чисел называется последовательность, содержащая каждое число от 1 до n ровно по одному разу.

После этого игра временно применяет перестановку p к исходному слову, т. е. составляет новое слово $(a_{p_1}, a_{p_2}, \dots, a_{p_n})$. Затем она вычисляет количество инверсий в слове и сообщает ответ участнику. Обратите внимание, что после попытки слово никак не изменяется.

Количеством инверсий в последовательности s длины n называется количество таких пар $1 \leq i < j \leq n$, что $s_i > s_j$. Например, в последовательности (1 0 1 1 0) четыре инверсии, поскольку подходят пары (1, 2), (1, 5), (3, 5) и (4, 5).

После того, как игрок назовет некоторое количество перестановок, он должен отгадать исходное слово.

Вам поручили протестировать новую игру. Вас очень заинтересовало: а какая же стратегия позволит быстро отгадать любое слово? Поэтому Вы решили написать программу для решения этой незамысловатой игры.

Протокол взаимодействия

Это интерактивная задача.

Сначала на вход подается целое число n ($2 \leq n \leq 4000$) — количество букв в слове.

Чтобы назвать перестановку, необходимо вывести в одной строке через пробел сначала символ «?», затем — n целых чисел p_i ($1 \leq p_i \leq n$). В ответ на вопрос программы-решения программа жюри подает ей на вход целое число x — количество инверсий в слове после применения перестановки p .

Вы можете назвать не более 450 перестановок.

Если программа-решение определила ответ на задачу, то она должна вывести в одной строке через пробел сначала символ «!», а затем строку, состоящую из n символов «0» или «1» — отгаданное слово. После вывода этой строки программа-решение должна немедленно завершиться.

После каждого действия Вашей программы Вы должны делать перевод строки и сбрасывать поток вывода. Если Вы используете «writeln» в Pascal, «cout << .. << endl;» в C++, «print» в Python, то сброс потока вывода происходит автоматически, ничего дополнительно делать не требуется. Если Вы используете другой способ вывода, рекомендуется делать сброс потока вывода. Обратите внимание, что перевод строки надо выводить в любом случае. Для сброса потока вывода можно использовать «fflush(stdout)» в C и C++, «flush(output)» в Pascal, «sys.stdout.flush()» в Python.

Если Вы получаете вердикт «Idleness Limit Exceeded» (IL), то это обозначает, что Ваша программа ожидает ввода, но данных в стандартном потоке ввода нет. Это может произойти, например, когда Ваша программа ошибочно ожидает ввода, а она должна вывести информацию для программы жюри, либо завершиться. Если Вы забыли вывести перевод строки либо сбросить поток вывода, то Вы также можете получить этот вердикт.

Гарантируется, что слово содержит хотя бы одну букву 0 и хотя бы одну букву 1.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n \leq 4$	5	
2	$n \leq 20$	4	1
3	$n \leq 100$	3	1 - 2
4	$n \leq 200$	9	1 - 3
5	$n \leq 450$	11	1 - 4
6	$n \leq 750$	13	1 - 5
7	$n \leq 1000$	9	1 - 6
8	$n \leq 1500$	8	1 - 7
9	$n \leq 2000$	8	1 - 8
10	$n \leq 2500$	7	1 - 9
11	$n \leq 3000$	7	1 - 10
12	$n \leq 3500$	7	1 - 11
13	$n \leq 3800$	7	1 - 12
14	Нет дополнительных ограничений	2	1 - 13

Пример

стандартный ввод	стандартный вывод
6	
	? 1 3 5 2 6 4
3	
	? 3 2 6 4 5 1
7	
	! 011010

Замечание

В первом запросе после применения перестановки получается слово 011100, которое содержит три инверсии: подходят пары (1, 2), (1, 3) и (1, 4).

В первом запросе после применения перестановки получается слово 110010, которое содержит семь инверсий: подходят пары (1, 3), (1, 4), (2, 3), (2, 4), (1, 6), (2, 6) и (5, 6).