

## Задача А. Episodes

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

Кот Miston поставил на скачивание сериал «Котокоп» («Police Cat»). В сериале  $n$  серий,  $i$ -я из которых сейчас скачивается со скоростью  $v_i$ , и браузер показывает, что  $i$ -я серия скачается ровно через  $t_i$  минут, что не совсем верно.

Браузер рассчитывает время, исходя из предположения, что скорость скачивания  $i$ -й серии не изменится. Однако когда какая-то серия оказывается полностью скачанной, скорость скачивания других серий увеличивается (других причин для изменения скорости скачивания той или иной серии и правда нет). При этом соотношения скоростей скачивания еще не скачанных серий неизменны. Иными словами, если  $v_i^t$  — скорость скачивания  $i$ -й серии в момент времени  $t$ , то для любых  $i$  и  $j$  верно, что при всех  $t$ , меньших момента окончания скачивания  $i$ -й или  $j$ -й серии, величина  $\frac{v_i^t}{v_j^t}$  одинакова.

Никакой другой нагрузки на сеть, кроме скачивания  $n$  серий, нет. То есть суммарная скорость скачивания  $\sum_{i=1}^n v_i^t$  постоянна (одинакова при любом  $t$ ).

Определите для каждой серии реальный момент времени, в который завершится ее загрузка.

### Формат входных данных

В первой строке ввода содержится целое число  $n$  — количество серий в сериале ( $1 \leq n \leq 3 \cdot 10^5$ ).

Каждая из следующих  $n$  строк содержит два целых числа  $v_i$  и  $t_i$  — текущая скорость скачивания и предполагаемое браузером время скачивания  $i$ -й серии ( $1 \leq v_i, t_i \leq 10^{18}$ ;  $\sum_{i=1}^n v_i \leq 10^{18}$ ).

### Формат выходных данных

Для каждой серии выведите настоящее время ее скачивания.

Ваш ответ будет засчитан, если его абсолютная или относительная ошибка не превосходит  $10^{-6}$ .

Формально, пусть ваш ответ равен  $a$ , а ответ жюри равен  $b$ . Ваш ответ будет засчитан, если  $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$ .

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	5	$n \leq 2$		полная
2	5	$n \leq 3$	0, 1	первая ошибка
3	15	$n, v_i, t_i \leq 3000$	0	первая ошибка
4	20	$n \leq 3000$	0 – 3	первая ошибка
5	20	$v_i = v_j$ для всех $i, j$		первая ошибка
6	15	$n \leq 30\,000$	0 – 4	первая ошибка
7	5	$t_i \leq 10^9$ для всех $i$ ; $\sum_{i=1}^n v_i \leq 10^9$	0, 3	первая ошибка
8	15	без дополнительных ограничений	0 – 7	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
2	4
2 4	4.6
3 5	
3	2.5
1 4	2
5 2	2.375
2 3	

## Задача В. Board Game

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

Студенты Иннополиса придумали новую настольную игру. Игра кооперативная, то есть игроки должны играть вместе, чтобы добиться определенной цели. Одной из фаз игры является распределение ресурсов.

Игра рассчитана ровно на  $n$  игроков, и на фазе распределения ресурсов между игроками по кругу распределяются  $n$  карточек с целыми числами, **упорядоченными по неубыванию**. Пусть  $i$ -му игроку досталось число  $a_i$  (тогда  $a_i \leq a_{i+1}$  для всех  $i \leq n - 1$ ). Назовем *опциями*  $i$ -го игрока тройку чисел  $\{a_{i-1}, a_i, a_{i+1}\}$ , то есть число, написанное на его карточке, и два числа его соседей. Игроки сидят по кругу, то есть для первого игрока вместо  $a_{i-1}$  будет опция  $a_n$ , а для  $n$ -го игрока вместо  $a_{i+1}$  будет опция  $a_1$ .

Каждый игрок должен выбрать ровно одну из доступных ему опций и получить ровно столько ресурсов из банка. После чего от двух оставшихся опций считается побитовое «И» (операция '&'), и такое количество очков добавляется к результату игроков в этом раунде. Например, если у игрока есть опции  $\{2, 5, 11\}$  и игрок выбирает взять 5 ресурсов, он получает  $2 \& 11 = 0010_2 \& 1011_2 = 2$  очка. Для успешного перехода к следующей фазе игры

1. суммарное количество взятых игроками ресурсов не должно превосходить  $\left(\sum_{i=1}^n a_i\right) + X$ , то есть должно не больше чем на  $X$  превышать количество ресурсов в случае, когда каждый игрок выбирает свое  $a_i$ ;
2. суммарное количество набранных очков должно быть максимально возможным.

Помогите игрокам определиться с выбором и успешно перейти к следующей фазе. Обратите внимание, что не требуется максимизировать количество взятых ресурсов при данном ограничении сверху, необходимо максимизировать только количество очков.

## Формат входных данных

В первой строке ввода через пробелы даны два целых числа  $n$  и  $X$  — количество игроков и ограничение на «дополнительное» число ресурсов ( $3 \leq n \leq 300$ ;  $0 \leq X \leq 10^5$ ).

Во второй строке через пробел перечислены  $n$  целых чисел  $a_i$  — значения, написанные на выданных игрокам карточках ( $0 \leq a_i \leq 10^5$ ;  $a_i \leq a_{i+1}$ ).

## Формат выходных данных

Выведите одно целое число — максимальное количество очков, которое могут набрать игроки за эту фазу.

## Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	13	$n \leq 14$	0	полная
2	8	$1 \leq a_i \leq 2$ для всех $i$		первая ошибка
3	10	$a_i \leq 3$ для всех $i$	2	первая ошибка
4	12	$a_n - a_1 \leq 1$	2	первая ошибка
5	17	$a_i \& a_{i+1} = a_i$ для всех $i$		первая ошибка
6	16	$a_i \leq 200$ для всех $i$	0, 2, 3	первая ошибка
7	24	без дополнительных ограничений	0 – 6	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
4 0 1 2 4 8	0
4 0 3 6 12 24	22
7 3 1 2 3 4 5 6 7	26

## Замечание

В первом примере  $a_i \& a_j = 0$  при  $i \neq j$ , поэтому набрать очки не получится.

Во втором примере игрокам следует взять ресурсы в количестве 24, 3, 6 и 3, соответственно. Тогда количество очков равно  $(3 \& 6) + (6 \& 12) + (12 \& 24) + (12 \& 24) = 2 + 4 + 8 + 8 = 22$ .

## Задача C. Helicopter

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Компании VK понадобилось срочно переместить сервера из одного дата-центра в другой. Это очень важная и срочная задача, потому что во время их перемещения не все сервисы будут полноценно функционировать, однако на дорогах сегодня почему-то пробки, поэтому сервера будут перемещаться на вертолете.

Карта области, над которой может перемещаться вертолет, представляет из себя поле, состоящее из  $n$  строк и  $m$  столбцов. Нумерация строк идет от 1 до  $n$  сверху-вниз, нумерация столбцов от 1 до  $m$  слева-направо. Каждая клетка представляет из себя участок одинаковой высоты. Будем характеризовать клетку в строке  $i$  и столбце  $j$  как  $(i, j)$ , а минимальную высоту, на которой может безопасно находиться вертолет на соответствующем участке, как  $a_{i,j}$ .

Сервера надо переместить из клетки  $(1, 1)$  в клетку  $(n, m)$ . Будем считать, что вертолет уже поднялся в воздух, и находится на высоте  $a_{1,1}$ . При попадании вертолета в клетку  $(n, m)$  на любой разрешенной высоте маршрут считается завершенным.

В любой момент времени высота, на которой находится вертолет, должна быть не меньше минимальной высоты полета для текущей клетки, то есть если обозначить за  $h$  высоту, на которой сейчас находится вертолет в клетке  $(i, j)$ , должно выполняться  $a_{i,j} \leq h$ . При этом из позиции  $(i, j)$  на высоте  $h$  можно:

- Перелететь в соседнюю по стороне клетку, потратив на такой перелет  $h$  топлива. При этом высота не изменится, а такое перемещение можно сделать только если  $a$  клетки, в которую совершается перелет, не больше  $h$ .
- Увеличить свою высоту на произвольное  $x \geq 0$ . На такой подъем тратится  $x$  топлива.
- Опуститься на любую высоту не меньше  $a_{i,j}$ . На это топливо не тратится.

Из-за погодных условий вертолет не всегда может лететь во все стороны. Доступные направления задаются строкой  $\mathbf{s}$ . Каждое возможное направление полета задается следующими буквами:

- 'R' — разрешено движение вправо из клетки  $(i, j)$  в клетку  $(i, j + 1)$ .
- 'L' — разрешено движение влево из клетки  $(i, j)$  в клетку  $(i, j - 1)$ .
- 'U' — разрешено движение вверх из клетки  $(i, j)$  в клетку  $(i - 1, j)$ .
- 'D' — разрешено движение вниз из клетки  $(i, j)$  в клетку  $(i + 1, j)$ .

Увеличивать и уменьшать высоту можно всегда. Направления перемещения 'R' и 'D' всегда разрешены. Перемещаться за границу поля, разумеется, нельзя.

Программисты из VK быстро справились определить оптимальный по затратам топлива маршрут для вертолета, даже при таком большом количестве условий на его перемещения. А справитесь ли вы?

### Формат входных данных

В первой строке даны два целых числа  $n$  и  $m$  — размеры поля, над которым перемещается вертолет ( $1 \leq n, m \leq 1000$ ).

Во второй строке дана строка  $\mathbf{s}$ , состоящая из различных символов 'L', 'R', 'U' и 'D' — разрешенные направления полета ( $2 \leq |\mathbf{s}| \leq 4$ ). Гарантируется, что 'R', 'D'  $\in \mathbf{s}$ .

Далее идут  $n$  строк по  $m$  чисел — описание поля,  $i$ -я строка содержит числа  $a_{i,j}$  (значения минимальной высоты полета), описывающие  $i$ -ю строку поля ( $0 \leq a_{i,j} \leq 10^9$ ).

## Формат выходных данных

Выведите единственное число — минимальное количество топлива, чтобы добраться из положения на высоте  $a_{1,1}$  в клетке  $(1, 1)$  до клетки  $(n, m)$ .

## Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены. Далее за  $A$  обозначено  $\max_{i,j} a_{i,j}$ .

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	5	$n, m, A \leq 10$	0	полная
2	5	$n, m \leq 10^3, A \leq 1$		первая ошибка
3	5	$n = 1, m \leq 10^3$		первая ошибка
4	7	$n, m, A \leq 500, s = \text{«RD»}$		первая ошибка
5	15	$n, m \leq 10^3, s = \text{«RD»}$	4	первая ошибка
6	5	$n, m \leq 10^3$ , и $a_{i-1,j}, a_{i,j-1} \leq a_{i,j}$		первая ошибка
7	15	$n, m \leq 10^3, s = \text{«LRD»}$	5	первая ошибка
8	5	$n, m, A \leq 100$	0, 1	первая ошибка
9	13	$n, m, A \leq 500$	0, 1, 4, 8	первая ошибка
10	25	без дополнительных ограничений	0 – 9	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
<pre>3 3 LRDU 1 2 4 4 4 4 1 1 1</pre>	13
<pre>5 5 LRD 0 10 0 0 0 0 0 0 10 0 10 10 10 10 0 9 5 3 2 0 3 3 2 5 0</pre>	30

## Замечание

В первом примере выгодно лететь вниз до упора, затем вправо до упора. Первые два перелета придется совершить на высоте 4, последние два можно совершить на высоте 1.

Во втором примере оптимальный путь проходит по клеткам с  $a_{i,j} = 0$  и стоит 0 топлива, однако для этого надо перемещаться вверх, а перемещения вверх запрещены. В таком случае можно пролететь по «нулевым» клеткам через клетку  $(1, 2)$  или  $(2, 4)$  с  $a_{1,2} = a_{2,4} = 10$ . Для перелета через любую из них понадобится подняться на высоту 10, после совершить два перелета на этой высоте.

## Задача D. Bill Restoration

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

**Это задача с двойным запуском.** Ваше решение будет запущено два раза.

Члены жюри олимпиады совершили в Иннополисе крупную покупку. Для удобства весь чек оплатил один человек, а остальные должны были после вернуть свою долю. Чек, разумеется, сразу был потерян, но сохранилась сделанная для истории фотография, к сожалению, довольно плохого качества.

На ней не только не видны отдельные позиции, вошедшие в чек, но и в строке с суммой оказались смазаны  $k$  цифр подряд. Тем не менее, члены жюри, хоть и с большим трудом, справились восстановить информацию о покупке. Вам предстоит проэмулировать этот сценарий.

При **первом запуске** вам дается целое число  $A$ , состоящее ровно из  $n$  цифр — суммарная стоимость покупки, а также число  $k$  — количество подряд идущих цифр, которые необходимо будет «замазать». Гарантируется, что  $n \geq 10^k + 10$ . Вы должны выбрать ровно  $k$  подряд идущих цифр числа и заменить их на символы '#’.

При **втором запуске** ваша программа получает на вход строку, которая была получена при первом запуске заменой  $k$  подряд идущих цифр на '#’. Ваша цель — целиком восстановить исходное число  $A$ , то есть вывести, какие  $k$  цифр «спрятаны» за символами решетки.

### Формат входных данных

В этой задаче используется мультитест, каждый тест содержит несколько наборов входных данных.

При первом запуске первая строка ввода содержит через пробел число 1 и два целых числа  $t$  и  $k$  — количество наборов входных данных и количество цифр, которые необходимо закрыть в каждом из них ( $1 \leq k \leq 6$ ). В  $i$ -й из следующих  $t$  строк дана строка  $A$  для  $i$ -го набора входных данных, состоящая из цифр (символы от '0' до '9') и задающая сумму чека ( $10^k + 10 \leq |A| \leq 10^6 + 20$ ).

Гарантируется, что сумма  $|A|$  по всем наборам входных данных в одном тесте не превосходит  $2 \cdot 10^6 + 10^5$ .

При втором запуске первая строка содержит число 2 и целое число  $t$  — количество наборов входных данных. В следующих  $t$  строках даны строки  $A'$  для всех наборов входных данных. Каждая  $A'$  состоит из символов от '0' до '9' или символов '#’ и была получена при первом запуске.

**Между двумя запусками наборы входных данных могут быть перемешаны.**

### Формат выходных данных

При первом запуске для каждого набора входных данных на отдельной строке выведите целое число  $l$  от 1 до  $n - k + 1$  — позицию начала отрезка цифр длины  $k$ , который будет заменен на '#’.

При втором запуске для каждого набора входных данных выведите в отдельной строке строку длины ровно  $k$ , состоящую из цифр — последовательность, которая была спрятана за отрезком из '#’. Ответы во втором запуске следует выводить в том же порядке, в котором строки следуют во входных данных для второго запуска.

### Протокол взаимодействия

Во избежание получения некорректных вердиктов вроде `Idleness Limit Exceeded` или `Security Violation` заканчивайте вывод каждой строки символом перевода строки ('`\n`') и сбросом буфера потока вывода (`cout.flush()` в C++, `System.out.flush()` в Java и `sys.stdout.flush()` в Python).

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	10	$k = 1, t = 1,$ '1' $\leq A_i \leq$ '2' для всех $i$ и $A$ сгенерирована равномерно		полная
2	13	$k = 1,$ '0' $\leq A_i \leq$ '2' для всех $i$	1	первая ошибка
3	17	$k = 1,$ '0' $\leq A_i \leq$ '7' для всех $i$	1, 2	первая ошибка
4	17	$k \leq 2,$ '0' $\leq A_i \leq$ '7' для всех $i$	1 – 3	первая ошибка
5	21	'0' $\leq A_i \leq$ '7' для всех $i$	1 – 4	первая ошибка
6	22	без дополнительных ограничений	0 – 5	первая ошибка

### Примеры

стандартный ввод	стандартный вывод	Пояснение
1 2 1 12345678901234567890 11111111111111111111	20 11	Первый запуск
2 2 1234567890123456789# 1111111111#111111111	0 1	Второй запуск

стандартный ввод	стандартный вывод	Пояснение
1 1 2 100000000020000000003000000000 400000000050000000005999999999 499999999939999999992999999999 19999999995555555555	51	Первый запуск
2 1 100000000020000000003000000000 40000000005000000000##99999999 499999999939999999992999999999 19999999995555555555	59	Второй запуск

## Задача E. Innopolis Data Center

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	1024 мегабайта

В дата-центре Университета Иннополис есть  $n$  серверов, занумерованных последовательными целыми числами от 1 до  $n$ . Сервера соединены  $n - 1$  оптоволоконными кабелями таким образом, что между любыми двумя серверами возможна передача данных по одному или нескольким кабелям, иными словами, связи между серверами образуют дерево.

Каждый оптоволоконный кабель передает данные с задержкой в 1 единицу времени, а задержка между двумя серверами равна расстоянию в дереве между этими серверами.

Требуется выбрать несколько серверов для системы распределенного хранения данных таким образом, чтобы задержка между любыми двумя серверами, входящими в систему, не превышала  $d$ . Пусть выбраны сервера  $s_1, s_2, \dots, s_k$ , тогда общая задержка системы равна сумме задержек между всеми  $\frac{k(k-1)}{2}$  парами серверов  $s_i$ . Формально:

1. должно выполняться  $\text{dist}(s_i, s_j) \leq d$  для любых  $i$  и  $j$  (где  $\text{dist}$  — расстояние в дереве);
2. и тогда общая задержка системы равна  $\sum_{i < j} \text{dist}(s_i, s_j)$ .

Ваша задача — ответить на  $q$  запросов. Каждый запрос задается номером одного сервера  $u$  и состоит в следующем: в случае, если конкретный сервер  $u$  будет включен в систему хранения данных, какова будет **максимальная** возможная общая задержка системы?

### Формат входных данных

Первая строка входных данных содержит два целых числа  $n$  и  $d$  — количество серверов в дата-центре и максимальная разрешенная задержка между двумя серверами ( $2 \leq n \leq 5 \cdot 10^5$ ;  $0 \leq d \leq n - 1$ ).

Каждая из следующих  $n - 1$  строк содержит по два целых числа  $a_i$  и  $b_i$  — номера двух серверов, соединенных  $i$ -м оптическим кабелем напрямую ( $1 \leq a_i, b_i \leq n$ ). Гарантируется, что между любыми двумя серверами существует возможность передать данные по одному или нескольким последовательным кабелям.

Следующая строка содержит одно целое число  $q$  — количество запросов ( $1 \leq q \leq 10$ ). В  $i$ -й из следующих  $q$  строк дано единственное целое число  $u_i$  — номер сервера из  $i$ -го запроса ( $1 \leq u_i \leq n$ ).

### Формат выходных данных

Для каждого запроса выведите в отдельной строке одно целое число — максимальную общую задержку системы в случае, когда соответствующий сервер входит в систему хранения данных.

### Система оценки

Баллы за каждую подзадачу с 1 по 4 начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Каждый тест в подзадачах 5 и 6 оценивается независимо и стоит 3 балла в подзадаче 5 или 2 балла в подзадаче 6.



Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	12	$n \leq 15$	0	первая ошибка
2	11	$d = n - 1$		первая ошибка
3	13	$n \leq 300$	0, 1	первая ошибка
4	16	$n \leq 5000$	0, 1, 3	первая ошибка
5	$8 \times 3$	$n \leq 10^5$	0, 1, 3, 4	полная
6	$12 \times 2$	без дополнительных ограничений	0 – 6	полная

## Примеры

стандартный ввод	стандартный вывод
7 2 2 1 1 3 2 4 5 4 4 7 6 5 7 7 6 5 4 3 2 1	9 4 9 9 4 9 4
2 0 1 2 1 1	0

## Замечание

В первом примере ответы на запросы соответствуют следующим множествам выбранных серверов:

- для сервера 1 — множество  $\{1, 2, 3\}$  или  $\{1, 2, 4\}$ ;
- для серверов 2, 4, 5 и 7 — множество  $\{2, 4, 5, 7\}$ ;
- для сервера 3 —  $\{1, 2, 3\}$ ;
- для сервера 6 —  $\{4, 5, 6\}$ .