

## Задача А. Рекордные ДОД

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

На Физтех-школе прикладной математики и информатики (ФПМИ МФТИ) проводится серия из  $n$  Дней открытых дверей. Известно, что в  $i$ -й день мероприятие посетило  $v_i$  человек.

День называется рекордным, если выполняются оба следующих условия:

1.  $v_i$  строго больше, чем количество посетителей во все предыдущие дни;
2. либо это последний день, либо  $v_i$  строго больше, чем количество посетителей в следующий день.

Заметьте, что первый день также может быть рекордным.

Требуется определить количество рекордных Дней открытых дверей ФПМИ.

### Формат входных данных

Первая строка содержит одно целое число  $n$  — количество Дней открытых дверей; Вторая строка содержит  $n$  целых чисел  $v_1, v_2, \dots, v_n$  ( $0 \leq v_i \leq 2 \times 10^5$ ) — количество посетителей в каждый день.

### Формат выходных данных

Выведите одно целое число — количество рекордных Дней открытых дверей.

### Примеры

стандартный ввод	стандартный вывод
8 1 2 0 7 2 0 2 0	2
5 9 9 9 9 9	0

### Замечание

Рассмотрим первый пример:  $n = 8$ , а количество посетителей по дням равно

1, 2, 0, 7, 2, 0, 2, 0.

Рекордными являются:

- День 2:  $2 > 1$ , и  $2 > 0$  (следующий день);
- День 4:  $7 > 2, 0, 1$ , и  $7 > 2$  (следующий день).

Ответ: 2.

## Задача В. Двухцветный путь

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Задана прямоугольная сетка размера  $n \times m$ , каждая клетка которой помечена символом 'B' (чёрная) или 'W' (белая). Также задано  $q$  запросов. Каждый запрос — это координаты начальной клетки  $(sr_i, sc_i)$  и конечной клетки  $(er_i, ec_i)$ . Требуется найти путь из начальной клетки в конечную, проходящий по равному количеству белых и чёрных клеток. Движение разрешено только в четырёх направлениях: вверх ('U'), вниз ('D'), влево ('L'), вправо ('R'). Посещение одной и той же клетки несколько раз разрешено; если клетка посещается несколько раз, её цвет учитывается соответствующее число раз.

### Формат входных данных

В первой строке заданы целые числа  $n$  ( $1 \leq n \leq 500$ ) и  $m$  ( $1 \leq m \leq 500$ ). Следующие  $n$  строк содержат по  $m$  символов — описание сетки ('B' или 'W').

Затем задано целое число  $q$  ( $1 \leq q \leq 10$ ). Следующие  $q$  строк содержат по четыре целых числа:  $sr_i$  ( $1 \leq sr_i \leq n$ ),  $sc_i$  ( $1 \leq sc_i \leq m$ ),  $er_i$  ( $1 \leq er_i \leq n$ ),  $ec_i$  ( $1 \leq ec_i \leq m$ ).

### Формат выходных данных

Для каждого запроса выведите одну строку: сначала число  $k$  ( $1 \leq k \leq 10^6$ ) — количество посещённых клеток, затем строку из  $k - 1$  символов — последовательность шагов ('U', 'D', 'L', 'R'). Если путь не существует, выведите только число  $-1$ .

### Пример

стандартный ввод	стандартный вывод
5 6 WBBWBW WWWBWW BWWWWB BBWWBB WWBBWW 3 1 1 2 5 4 5 2 1 5 1 5 2	6 RRRRD -1 4 URD

## Задача С. Кубики в коробке

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

У Ромы есть  $n_s$  маленьких кубиков с ребром длины 1 и  $n_b$  больших кубиков с ребром длины  $l$ . Он хочет упаковать все кубики в один прямоугольный параллелепипед так, чтобы рёбра всех кубиков были параллельны рёбрам коробки. Кубики нельзя разрезать или деформировать.

Требуется найти минимально возможный объём коробки, в которую можно упаковать все кубики.

### Формат входных данных

В единственной строке заданы три целых числа:  $n_s$  ( $1 \leq n_s \leq 10^9$ ),  $n_b$  ( $1 \leq n_b \leq 10^6$ ) и  $l$  ( $2 \leq l \leq 10$ ) — количество маленьких кубиков, количество больших кубиков и длина ребра большого кубика соответственно.

### Формат выходных данных

Выведите одно целое число — минимальный объём коробки. Гарантируется, что ответ уместается в знаковый 32-битный тип данных.

### Примеры

стандартный ввод	стандартный вывод
2 2 2	20
19 1 2	27

## Задача D. Наиболее значимый шаблон

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Назовём две строки  $u$  и  $v$ , составленные из строчных латинских букв, *эквивалентными*, если они имеют одинаковую длину  $n$  и существует биективное отображение символов алфавита  $\varphi$ , такое что

$$u = \varphi(v_1) \varphi(v_2) \dots \varphi(v_n).$$

Иными словами, можно переназначить буквы в строке  $v$  так, чтобы получилось слово  $u$ ; при этом разные буквы из  $v$  должны соответствовать разным буквам в  $u$ .

- строки «apple» и «arrow» эквивалентны (например,  $a \mapsto a$ ,  $p \mapsto r$ ,  $l \mapsto o$ ,  $e \mapsto w$ );
- строки «hello» и «motto» не эквивалентны.

Рассмотрим строку  $s$ . Пусть  $q$  — некоторая строка. Будем говорить, что  $q$  *появляется* в  $s$   $a$  раз, если существует  $a$  позиций  $i$ , таких, что подстрока  $s[i..i + |q| - 1]$  эквивалентна строке  $q$ .

Определим *значимость* строки  $q$  по отношению к  $s$  как величину

$$\text{val}(q) = a \cdot |q|,$$

где  $a$  — число её появлений в  $s$ , а  $|q|$  — длина  $q$ .

Требуется по данной строке  $s$  найти такую строку  $q$ , для которой значение  $\text{val}(q)$  максимально. Если существует несколько подходящих вариантов, разрешается вывести любой из них.

### Формат входных данных

Во входном файле дана строка  $s$ , состоящая только из букв латинского алфавита ('a'–'z'). Длина строки не превышает 4000.

### Формат выходных данных

Выведите одну строку — шаблон  $q$  с наибольшей значимостью.

### Пример

стандартный ввод	стандартный вывод
missmississippimissed	ok

### Замечание

ok *появляется* 15 раз в строке.

## Задача Е. Нумерация домов

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В городе бесконечно много домов, пронумерованных последовательно, начиная с 1. Вы имеете коробку с пластиковыми цифрами. Для каждой цифры  $i$  от 0 до 9 в коробке есть  $d_i$  копий.

Чтобы пронумеровать дом с номером  $n$ , нужно использовать соответствующие цифры для записи числа  $n$ . Например, для дома с номером 474 используются две цифры 4 и одна цифра 7.

Найдите максимальное число домов  $x$ , которые можно пронумеровать последовательно с 1 до  $x$ , используя имеющиеся цифры.

### Формат входных данных

Первая строка содержит одно целое число  $t$  ( $1 \leq t \leq 100$ ) — количество тестов.

Далее следуют  $t$  тестов. Каждый тест состоит из одной строки с 10 целыми числами  $d_0, d_1, \dots, d_9$  ( $0 \leq d_i \leq 10^{16}$ ) — количество каждой цифры в коробке.

### Формат выходных данных

Для каждого теста выведите одно целое число — максимальный номер  $x$  дома, который можно пронумеровать.

### Пример

стандартный ввод	стандартный вывод
1 1 3 1 1 2 1 1 2 1 1	10

## Задача F. Выбор команды

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Скоро распределение на команды в МФТИ. Будут участвовать  $n$  студентов. У каждого студента  $i$  есть три характеристики:  $x_i$  — навык придумывания идеи,  $y_i$  — навык реализации,  $z_i$  — удача.

Требуется выбрать трёх студентов так, чтобы для каждого из них нашлась хотя бы одна характеристика, в которой он строго превосходит двух других участников команды. Среди всех таких команд нужно найти ту, у которой сумма максимальных значений по каждой характеристике — наибольшая.

Формально: если выбраны студенты индексами  $i, j, k$ , то для студента  $i$  существует хотя бы одна характеристика (например,  $x_i$ ), такая что  $x_i > x_j$  и  $x_i > x_k$ . То же самое должно выполняться для студентов  $j$  и  $k$ .

Общая сила команды это  $\max(x_i, x_j, x_k) + \max(y_i, y_j, y_k) + \max(z_i, z_j, z_k)$ . Если такой команды не существует — вывести  $-1$ .

### Формат входных данных

В первой строке задано целое число  $n$  ( $3 \leq n \leq 150\,000$ ).

В следующих  $n$  строках заданы по три целых числа:  $x_i$  ( $1 \leq x_i \leq 10^8$ ),  $y_i$  ( $1 \leq y_i \leq 10^8$ ),  $z_i$  ( $1 \leq z_i \leq 10^8$ ) — характеристики  $i$ -го студента.

### Формат выходных данных

Выведите одно целое число — максимальную возможную силу команды, удовлетворяющей условию. Если такой команды не существует — выведите  $-1$ .

### Примеры

стандартный ввод	стандартный вывод
5 3 1 4 2 3 1 1 5 5 4 4 2 5 2 3	13
3 1 2 3 1 2 6 1 2 4	-1

## Задача G. Схождение уровней

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Есть  $2n$  участников и  $n$  уровней, пронумерованных от 1 до  $n$ . В каждом шаге:

- участники образуют  $n$  пар (по одной паре на каждом уровне);
- победитель на уровне  $x$  перемещается на уровень  $\max(1, x - 1)$ ;
- проигравший на уровне  $x$  перемещается на уровень  $\min(n, x + 1)$ .

Два участника начинают на различных уровнях  $a$  и  $b$  ( $1 \leq a < b \leq n$ ). Считается, что они могут произвольно выбирать, выигрывать или проигрывать на каждом шаге.

Найдите минимальное число шагов, через которое участники могут оказаться на одном уровне.

### Формат входных данных

В единственной строке даны три целых числа  $n$ ,  $a$  и  $b$  ( $1 \leq a < b \leq n \leq 10^{18}$ ) — количество уровней и уровни, на которых находятся участники изначально.

### Формат выходных данных

Выведите одно целое число — минимальное количество шагов, после которых участники могут оказаться на одном уровне.

### Примеры

стандартный ввод	стандартный вывод
6 2 3	2
4 2 4	1

## Задача Н. Выбор отрезков из множеств

Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Заданы целое число  $l$  и  $n$  наборов отрезков. Каждый набор состоит из отрезков одинаковой длины.

Необходимо выбрать последовательность отрезков, которая полностью покрывает отрезок  $[0, l)$  без пропусков. При этом:

- отрезки должны образовывать непрерывное покрытие  $[0, l)$ , то есть каждый следующий отрезок начинается не позже, чем заканчивается предыдущий;
- из каждого набора разрешается использовать не более одного отрезка;

Нужно определить минимальное число отрезков, которые необходимо выбрать, чтобы покрыть  $[0, l)$ . Если покрытие невозможно, выведите  $-1$ .

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $l$  ( $1 \leq n \leq 20$ ,  $1 \leq l \leq 10^8$ ).

Далее следуют  $n$  строк. В  $i$ -й строке записаны два числа  $d_i$  и  $c_i$  ( $1 \leq d_i \leq l$ ,  $1 \leq c_i \leq 1000$ ) — длина отрезка и количество доступных отрезков в  $i$ -м наборе. Затем следуют  $c_i$  целых чисел  $s_{i,1}, s_{i,2}, \dots, s_{i,c_i}$  ( $0 \leq s_{i,j} \leq l$ ) — моменты начала отрезков, строго возрастающие и попарно различные. Каждый отрезок имеет вид  $[s_{i,j}, s_{i,j} + d_i)$ .

### Формат выходных данных

Выведите одно целое число — минимальное количество отрезков, необходимых для покрытия  $[0, l)$ . Если покрытие невозможно, выведите  $-1$ .

### Примеры

стандартный ввод	стандартный вывод
4 100 50 3 15 30 55 40 2 0 65 30 2 20 90 20 1 0	3
2 10 4 1 0 5 1 5	-1

### Замечание

В первом примере нужно выбрать отрезок из четвертого набора с началом в 0, отрезок из первого набора с началом в 15, отрезок из второго набора с началом в 65.



## Задача I. Откуда ни стартуй

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Есть граф, состоящий из  $n$  вершин. Из каждой вершины выходит  $m$  ориентированных пронумерованных ребер. Граф задан матрицей  $s$  размера  $n \times m$ , где  $s_{i,j}$  — номер вершины, в которую ведет  $j$ -е ребро из вершины  $i$ .

Необходимо составить последовательность индексов ребер  $i_1, i_2, \dots, i_L$  такую, что для любого начального состояния  $v_0$  выполняется  $v_k = s_{v_{k-1}, i_k}$  и  $v_L = 1$ .

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  — количество вершин и количество ребер в каждой вершине ( $3 \leq n \leq 200$ ,  $1 \leq m \leq 200$ ).

Следующие  $m$  строк содержат по  $n$  целых чисел  $s_{1,j}, s_{2,j}, \dots, s_{n,j}$  — вершины, на которые ведут ребра из каждой вершины ( $1 \leq s_{i,j} \leq n$ ).

Гарантируется, что существует последовательность длины не более 5,000,000, которая удовлетворяет условию.

### Формат выходных данных

Выведите последовательность  $i_1, i_2, \dots, i_L$  индексов стрелок, по одному числу на строку, которая гарантирует, что для любой вершины  $v_0$  после применения всей последовательности выполняется  $v_L = 1$ . Длина последовательности не должна превышать 5 000 000.

### Пример

стандартный ввод	стандартный вывод
4 3	1
4 4 1 3	2
1 3 2 4	1
4 2 3 1	2
	3
	1
	3

### Замечание

Все нумерации с единицы

## Задача J. Волейбол Филиппа

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Даны два целых числа  $n$  и  $a$  — общее число очков в матче и минимальное количество очков, необходимое для победы в одном сете.

Матч играется до трёх выигранных сетов (максимум пять сетов). Сет выигрывает команда, первой набравшая хотя бы  $a$  очков и опережающая соперника минимум на два очка. Счёт в сете не ограничен сверху.

Требуется определить: существует ли валидный матч, в котором суммарное количество очков равно  $n$ . Если существует, вывести любой пример такого матча (последовательность сетов, каждый в отдельной строке, в формате “очкикоманды1 очкикоманды2”). Если не существует — вывести  $-1$ .

### Формат входных данных

В единственной строке заданы два целых числа:  $n$  ( $2 \leq n \leq 10^9$ ) и  $a$  ( $2 \leq a \leq 10^9$ ).

### Формат выходных данных

Если валидный матч существует — выведите от трёх до пяти строк, каждая содержит два целых числа — счёт одного сета. Порядок сетов должен соответствовать хронологии: матч заканчивается, как только одна из команд выигрывает 3 сета.

Если валидного матча не существует, выведите  $-1$ .

### Пример

стандартный ввод	стандартный вывод
87 10	10 2 3 10 12 10 13 15 10 2

## Задача К. Минимальная Галерея

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В рамках учебного проекта студенты ФПМИ МФТИ решили устроить импровизированную выставку на открытом воздухе. В поле расставлены  $n$  интерактивных постов, на которых студенты хотят закрепить информационные панели с демонстрацией своих проектов по алгоритмам и визуализации данных.

Чтобы ограничить пространство, студенты выбирают четыре поста и соединяют их ленточными ограждениями, образуя простой четырёхугольник  $p_1p_2p_3p_4$  (выпуклый или вогнутый). Внутри этого периметра будет находиться зона с экспонатами.

Каждый квадратный метр ограждённой зоны требует обслуживания: студентам нужно периодически проверять и обновлять экспонаты. Обслуживание стоит два арткоина за каждый квадратный метр. Поэтому студенты хотят выбрать четыре поста так, чтобы площадь четырёхугольника была минимальной, минимизируя расходы на обслуживание.

### Формат входных данных

Первая строка содержит одно целое число  $t$  ( $1 \leq t \leq 30$ ) — количество тестов. Далее следуют  $t$  тестов.

Каждый тест начинается с целого числа  $n$  ( $4 \leq n \leq 1200$ ) — количество постов. Следующие  $n$  строк содержат по два целых числа  $x_i$  и  $y_i$  ( $-10^9 \leq x_i, y_i \leq 10^9$ ) — координаты  $i$ -го поста. Посты можно считать точками на плоскости. Никакие три поста не лежат на одной прямой.

### Формат выходных данных

Для каждого теста выведите одно число — минимальное количество арткоинов, равное удвоенной площади минимального простого четырёхугольника с вершинами среди данных точек.

### Пример

стандартный ввод	стандартный вывод
2	200
4	30
-5 5	
-5 -5	
5 5	
5 -5	
5	
-5 5	
-5 -5	
5 5	
5 -5	
4 2	

## Задача L. Робот

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Поверхность планеты представлена сеткой размером  $10^9 \times 10^9$  клеток. Столбцы нумеруются от 1 до  $10^9$  с запада на восток, строки — от 1 до  $10^9$  с севера на юг. Клетка с координатами  $(w, h)$  обозначает клетку в  $w$ -м столбце и  $h$ -й строке. Изначально робот находится в клетке  $(1, 1)$ .

Робот управляется программой, которая представляет собой строку, состоящую из следующих команд:

- N — переместиться на одну клетку на север (уменьшить номер строки на 1);
- S — переместиться на одну клетку на юг (увеличить номер строки на 1);
- E — переместиться на одну клетку на восток (увеличить номер столбца на 1);
- W — переместиться на одну клетку на запад (уменьшить номер столбца на 1).

Кроме того, программа может содержать конструкции повторения вида:

$$k(P)$$

где  $k$  — целое число от 2 до 9 включительно, а  $P$  — непустая подпрограмма (строка, состоящая из допустимых команд и/или вложенных повторений). Конструкция  $k(P)$  эквивалентна выполнению подпрограммы  $P$  последовательно  $k$  раз.

Например:

$$2(\text{NWE}) \equiv \text{NWENWE}, \quad 3(\text{S2(E)}) \equiv \text{SEEESEESEE}, \quad \text{EEEE4(N)2(SS)} \equiv \text{EEEENNNNSSSS}.$$

Передвижения выполняются в порядке следования команд в раскрытой программе. Планета имеет тороидальную (замкнутую) поверхность:

- если робот движется восточнее столбца  $10^9$ , он оказывается в столбце 1;
- если робот движется южнее строки  $10^9$ , он оказывается в строке 1;
- если робот движется западнее столбца 1, он оказывается в столбце  $10^9$ ;
- если робот движется севернее строки 1, он оказывается в строке  $10^9$ .

Требуется определить конечные координаты  $(w, h)$  робота после выполнения всей программы.

### Формат входных данных

Первая строка содержит одно целое число  $t$  ( $1 \leq t \leq 100$ ) — количество тестов.

Далее следуют  $t$  строк, каждая из которых содержит одну строку — программу, которую робот должен выполнить. Строка программы содержит только символы N, S, E, W, цифры от 2 до 9, а также круглые скобки, описывающие повторения. Длина каждой программы не превышает  $10^4$  символов.

### Формат выходных данных

Для каждого теста выведите одну строку, содержащую два целых числа  $(w, h)$  — координаты конечного положения робота.

### Пример

стандартный ввод	стандартный вывод
4 NNWWW E S3(N)S2(W)S 2(3(SE)2(E2(WW)E))	999999998 999999999 2 1 999999999 1 999999999 7

## Задача М. Причем тут Евровидение

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дано  $n$  певцов. У каждого певца  $i$  есть текущее количество баллов  $p_i$ . Все  $p_i$  различны. В финальном раунде судьи должны распределить  $x$  баллов между всеми  $n$  певцами. Каждый певец должен получить целое положительное число баллов  $s_i$ , все  $x$  баллов должны быть распределены.

После распределения баллы  $s_i$  объявляются по возрастанию, при равенстве баллов — в порядке возрастания исходных  $p_i$ . После каждого объявления пересчитываются суммарные баллы каждого певца  $p_i + s_i$ , и определяется лидер — певец с наибольшим числом баллов. Гарантируется, что в каждый момент лидер единственный (нельзя допускать ничьих за первое место по суммарным баллам).

Финал называется захватывающим, если после каждого объявления лидер изменяется, то есть певец, который идёт первым, всегда другой, чем был перед предыдущим объявлением. В том числе лидер должен измениться после первой публикации (по сравнению с исходным лидером до финального раунда).

Требуется определить, сколько различных итоговых порядков певцов по суммарным баллам можно получить, если выбрать распределение  $s_i$ , удовлетворяющее условиям захватывающего финала. Два распределения считаются одинаковыми, если итоговый порядок певцов совпадает, вне зависимости от конкретных значений  $s_i$ .

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $x$  ( $1 \leq n \leq 12$ ,  $1 \leq x \leq 700$ ) — количество певцов и суммарное количество баллов, которое необходимо распределить.

Вторая строка содержит  $n$  различных целых чисел  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq 700$ ) — количество баллов каждого певца до финального раунда.

### Формат выходных данных

Выведите одно целое число — количество различных итоговых порядков певцов, которые могут быть получены при распределении баллов так, чтобы финал был захватывающим.

### Примеры

стандартный ввод	стандартный вывод
3 12 3 1 4	3
4 15 1 3 7 4	0