

### Задача 1. Крепостная стена (100 баллов)

Имя входного файла: fortress.in  
Имя выходного файла: fortress.out  
Максимальное время работы на одном тесте: 1 секунда  
Максимальный объем используемой памяти: 64 Кб



Имеется древняя крепостная стена длиной  $L$  из  $N$  прямоугольных камней разного размера. Никакой камень не вылезает за границы стены. Стена очень древняя и часть камней разрушилась и выпала, образовав сквозные пустоты. Археологам необходимо измерить количество камней под заданными  $M$  точками стены. Точки задаются натуральным числом – расстоянием от левого края стены.

От Вас **требуется** написать программу расчета количества камней под заданными точками по заданным координатам всех камней в стене.

#### Формат входного файла

В первой строке входного файла записаны 3 целых числа:

- натуральное число  $L$  – длина стены ( $1 \leq L \leq 10\,000$ );
- натуральное число  $N$  – количество камней в стене ( $1 \leq N < 10\,000$ );
- натуральное число  $M$  – число точек стены, под которыми необходимо определить количество камней ( $1 \leq M < 100\,000$ ).

В следующих  $N$  строках идут пары натуральных чисел  $l$  и  $r$  ( $1 \leq l \leq r \leq L$ ) – левые и правые концы камней стены.

В следующих  $M$  строках идут натуральные числа  $q$  ( $1 \leq q \leq L$ ) – координаты точек стены, под которыми нужно вычислить количество камней

#### Формат выходного файла

В  $M$  строках выходного файла вывести по одному целому числу – количество камней над очередной точкой.

#### Пример входного и выходного файлов

fortress.in	fortress.out
39 4 7	4
3 21	3
3 15	0
2 20	4
3 17	4
4	0
17	0
33	

5 9 25 37	
--------------------	--

**Рисунок, поясняющий пример.**



**Задача 2. Красивые обои (100 баллов)**

Имя входного файла: wallpaper.in  
 Имя выходного файла: wallpaper.out  
 Максимальное время работы на одном тесте: 1 секунда  
 Максимальный объем используемой памяти: 64 КБ

Петя - будущий программист. А его старший брат Володя – дизайнер. Он создает новые рисунки на выпускающем обои предприятии. Заказчик попросил создать различным образом разрисованные прямоугольниками обои. Заказ был мало творческим, скорее механическим, поэтому Володя попросил Петю попробовать написать программу, которая будет рисовать рамки прямоугольников.

Поле для будущей композиции – тоже прямоугольник, который состоит из строк (пронумерованных от 1 до X) и столбцов (пронумерованных от 1 до Y) Нумерация соответственно сверху-вниз и слева-направо. На поле необходимо вывести N рамок прямоугольников, заданных двумя точками  $(w1_i, h1_i)$  и  $(w2_i, h2_i)$  - координаты противоположных углов, лежащих на диагонали i-того прямоугольника.

Для оценки гармонии рисунка программа должна создавать изображение с помощью букв латинского алфавита, i-ый прямоугольник создается i-ой буквой алфавита, а свободные места заполнять символом 0. Прямоугольник, нарисованный позже, может перекрывать ранее нарисованные.

Итак, от Вас **требуется** написать программу, которая будет рисовать рамки прямоугольников на поле обоев.

**Формат входных данных**

Первая строка содержит 3 натуральных числа X, Y и N – размеры композиции и число прямоугольников ( $2 \leq X \leq 100, 2 \leq Y \leq 100, 1 \leq N \leq 26$ ).

Следующие N строк содержащих координаты прямоугольников – натуральные числа  $w1_i, h1_i, w2_i, h2_i$  ( $1 \leq w1_i < w2_i \leq X, 1 \leq h1_i < h2_i \leq Y$ ).

**Формат выходных данных**

X строк композиции, заполненных в соответствии со входными данными

**Пример входного и выходного файлов**

wallpaper.in	wallpaper.out
10 10 3	cccccccc00
2 4 9 7	c00aaaac00
6 6 10 10	cccccccc00
1 1 3 8	000a00a000
	000a00a000
	000a0bbbbbb
	000a0ba00b
	000a0ba00b
	000aaba00b
	00000bbbbbb

**Задача 3. Олимпиада (100 баллов)**

Имя входного файла: olympic.in  
Имя выходного файла: olympic.out  
Максимальное время работы на одном тесте: 1 секунда  
Максимальный объем используемой памяти: 64 Кб

Олимпиада по информатике в школах области проходила два дня, причем в первый день были предложены более простые для решения задачи, а во второй день задачи более высокого уровня сложности. Результаты олимпиады отсортировали по каждому дню в порядке убывания баллов, полученных каждым участником, и разместили в таблице так, что за самым низким баллом первого дня шел самый высокий балл второго дня. Для анализа результатов олимпиады потребовалось определить самое большое различие в баллах двух соседних результатов таблицы. Информация о результатах олимпиады содержала данные о результатах обоих дней, которые приходили постепенно.

От Вас требуется написать программу, определяющую указанное в условии различие.

**Формат входных данных**

Первая строка содержит целое число N – количество участников олимпиады ( $2 \leq N \leq 200$ ).

Следующие N строк два числа  $d_i$  и  $b_i$  – день и балл i-го участника ( $d_i$  равно 1 или 2 и  $0 \leq b_i \leq 100$ )

**Формат выходных данных**

Одно число – максимальная разница в баллах стоящих рядом в отсортированной по дням таблице.

**Пример входного и выходного файлов**

olympic.in	olympic.out
------------	-------------

8	52
1 23	
2 38	
1 33	
1 85	
2 66	
1 22	
2 44	
2 52	

#### Задача 4. Волшебный конь (100 баллов)

Имя входного файла: horse.in  
 Имя выходного файла: horse.out  
 Максимальное время работы на одном тесте: 1 секунда  
 Максимальный объем используемой памяти: 64 Кб

В волшебной стране Дивнопутии жил волшебный конь Бегень. Особенность Бегня заключалась в том, что после долгого застоя, когда конь никуда не скакал, у него происходил упадок сил, и поэтому, отправляясь в очередной путь, Бегень мог не преодолеть весь путь без остановки и подзаправки волшебным сеном и родниковой водой. Но каждая пробежка увеличивала силы коня и после небольшого подкрепления каждый следующий участок пути, который преодолевал Бегень, был больше предыдущего, по крайней мере, на 2 км. Известно также, что каждый участок пути, преодоленный конем, это целое число километров.

Итак, от Вас **требуется** написать программу, которая выведет все возможные для заданного расстояния разбиения пути Бегня на отрезки, соответствующие условию задачи.

##### **Формат входных данных**

Входной файл содержит одно натуральное число  $N$  – количество километров, которое должен проскакать Бегень ( $1 \leq N \leq 150$ ).

##### **Формат выходных данных**

Выходной файл содержит одно натуральное число  $K$  – количество разбиений заданного расстояния на отрезки пути коня.

##### **Пример входного и выходного файлов**

horse.in	horse.out
10	6

**Пояснения к полученному результату – разбиения расстояния 10 км на отрезки пути коня:**

- 10=1+3+6
- 10=1+9
- 10=2+8
- 10=3+7
- 10=4+6
- 10=10