

## Задача А Черепаха

|                                           |              |
|-------------------------------------------|--------------|
| Максимальное время работы на одном тесте: | 2 секунды    |
| Максимальный объем используемой памяти:   | 64 мегабайта |

Домик черепахи расположен в начале прямой узкой грядки, на которой должны прорасти одуванчики – ее любимое лакомство. И вот черепахе приснился вещий сон. Из него она узнала, что наконец-то после полуночи начнут расти одуванчики. Ей даже приснилось, в какой момент времени, и в какой точке грядки вырастет каждый одуванчик. Ровно в полночь черепаха выползла из домика, чтобы съесть все одуванчики и до следующей полуночи вернуться домой.

Черепаха может ползти со скоростью, не превосходящей величины  $v_{\max}$ . Одуванчик она съедает, остановившись на время  $d$ . Если одуванчик начать есть, но не доест до конца, то он засыхает, поэтому его надо съедать за один прием. Одуванчики прорастают тем позже, чем дальше они расположены от начала грядки. В одной точке не могут прорасти несколько одуванчиков, а также несколько одуванчиков не могут прорасти в один момент времени.

Требуется определить, в какой момент времени черепаха сможет вернуться домой, съев все одуванчики и затратив на путешествие наименьшее время.

### Входные данные

В 1-й строке входного файла находятся 2 целых числа, разделенные пробелом:  $v_{\max}$  (в см/мин) и  $d$  (в минутах),  $0 < v_{\max} \leq 200$ ,  $0 \leq d \leq 500$ .

Во 2-й строке находится число  $N$  – количество одуванчиков (в штуках).  $0 \leq N \leq 1400$  при  $d = 0$ , в противном случае  $0 \leq N \leq 200$ .

В каждой из последующих  $N$  строк расположены: целое число  $x_i$  – расстояние от одуванчика до начала грядки (в сантиметрах),  $0 \leq x_i \leq 32767$ , и через пробел  $t_i$  – момент прорастания одуванчика (в формате hh:mm). Пары приведены в порядке возрастания расстояний.

### Выходные данные

Выходной файл должен содержать момент времени возвращения черепахи домой (в формате hh:mm), округленный до целых минут в большую сторону.

### Примечания

1. В часе – 60 минут, в сутках – 24 часа.
2. Время в сутках изменяется от 00:00 до 23:59.
3. Можете считать, что черепаха не меняет направления движения до тех пор, пока не доползет до последнего одуванчика.

### Пример

| Входные данные        | Выходные данные |
|-----------------------|-----------------|
| 3 1<br>1<br>100 00:01 | 01:08           |

## Задача В Забавный конфуз

|                                           |              |
|-------------------------------------------|--------------|
| Максимальное время работы на одном тесте: | 2 секунды    |
| Максимальный объем используемой памяти:   | 64 мегабайта |

Пусть  $A$  — массив, состоящий из  $N$  элементов  $A_1, \dots, A_N$ . Обозначим его максимальное и минимальное значение как  $\max(A)$  и  $\min(A)$  соответственно. Вычислим сумму элементов  $S$ ,  $S = A_1 + A_2 + \dots + A_N$ . Заменяем каждый элемент массива на разницу  $S$  и этого элемента:  $A_i := S - A_i$ ,  $1 \leq i \leq N$ . Такое преобразование массива  $A$  назовем операцией *Confuse*. Напишите программу, которая по массиву  $B$ , полученному в результате  $K$ -кратного применения операции *Confuse* к некоторому массиву  $A$ , вычислит разность  $\max(A) - \min(A)$ .

### Входные данные

Первая строка входного файла содержит целые числа  $N$  и  $K$ , где  $N$  — количество элементов массива  $B$  ( $2 \leq N \leq 10000$ ), а  $K$  — количество применений операции *Confuse* к начальному массиву  $A$ ,  $1 \leq K \leq 100$ . Вторая строка файла содержит  $N$  элементов массива  $B$ . Элементы массива  $B$  — целые числа, принадлежащие диапазону от  $-2\,000\,000\,000$  до  $2\,000\,000\,000$ .

### Выходные данные

Единственная строка выходного файла должна содержать целое число - разность  $\max(A)$  и  $\min(A)$ .

### Пример

| Входные данные     | Выходные данные |
|--------------------|-----------------|
| 4 2<br>45 52 47 46 | 7               |

## Задача С Перекраска

|                                           |              |
|-------------------------------------------|--------------|
| Максимальное время работы на одном тесте: | 2 секунды    |
| Максимальный объем используемой памяти:   | 64 мегабайта |

Дано прямоугольное поле, каждая клетка которого покрашена в какой-то цвет. За один ход необходимо перекрасить все клетки одного цвета в другой цвет. Каждая клетка перекрашивается один раз. Стоимость перекраски одной клетки зависит от номера хода и задается функцией:  $F(i) = (A * F(i-1) + B) \bmod C + 1$ ,  $F_1$  – известная стоимость первого хода.

Необходимо перекрасить все поле в один цвет так, чтобы общая стоимость перекраски была бы максимальной. При этом каждая клетка хотя бы один раз должна быть перекрашена в другой цвет.

### Входные данные

В первой строке натуральные задаются числа  $F_1$ ,  $A$ ,  $B$  и  $C$  ( $1 \leq F_1, A, B, C \leq 10000$ ) – параметры функции  $F$ . Во второй строке задаются два натуральных числа  $M$  и  $N$  ( $1 \leq N, M \leq 50$ ) – размеры поля. В последующих  $M$  строках записано по  $N$  натуральных чисел, не превосходящих  $2^{31}$  – цвета клеток.

### Выходные данные

В первую строку выведите минимальное число ходов. Во вторую строку выведите в каком порядке будут перекрашиваться цвета, встречающиеся в таблице.

### Пример

| Входные данные                                             | Выходные данные |
|------------------------------------------------------------|-----------------|
| 1 3 1 5<br>4 4<br>1 2 3 6<br>2 1 1 2<br>3 1 1 3<br>2 2 2 2 | 4<br>6 2 3 1    |

## Задача D Ожерелье

|                                           |              |
|-------------------------------------------|--------------|
| Максимальное время работы на одном тесте: | 1 секунда    |
| Максимальный объем используемой памяти:   | 64 мегабайта |

В витрине ювелирного магазина стоит манекен, на шею которого надето ожерелье. Оно состоит из  $N$  колечек, нанизанных на замкнутую нить. Все колечки имеют разные размеры. В зависимости от размера колечки пронумерованы числами от 1 до  $N$ , начиная с самого маленького и до самого большого. Колечки можно передвигать вдоль нити и протаскивать одно через другое, но только в том случае, если номера этих колечек отличаются более чем на единицу.

Продавец хочет упорядочить колечки так, чтобы они располагались по возрастанию номеров вдоль нити по часовой стрелке. Снимать ожерелье с манекена нельзя.

Требуется написать программу, которая по заданному начальному расположению колечек находит последовательность протаскиваний колечек одно через другое, приводящую исходное расположение колечек в желаемое.

### Входные данные

В первой строке входного файла записано число  $N$  ( $2 \leq N \leq 50$ ).

Во второй строке через пробел следуют  $N$  различных чисел от 1 до  $N$  — номера колечек, расположенных вдоль нити по часовой стрелке.

### Выходные данные

Выходной файл должен содержать описание процесса упорядочения.

В каждой строке, кроме последней, должны быть записаны через пробел два числа, указывающие номера колечек, протаскиваемых друг через друга. В последней строке должен стоять ноль.

Количество строк выходного файла не должно превышать 50000.

Если требуемого упорядочения колечек достичь не удастся, в выходной файл нужно вывести одно число  $-1$ .

### Пример

| Входные данные | Выходные данные |
|----------------|-----------------|
| 4              | 1 3             |
| 3 2 4 1        | 2 4             |
|                | 1 4             |
|                | 0               |