

# A 协调巴士

时空限制：1s 256MB

## 文件名

bus.in/bus.out/bus.cpp

## 题意

一场别开生面的牛吃草大会就要在Farmer John的农场举办了！

世界各地的奶牛将会到达当地的机场, 前来参会并且吃草。具体地说, 有  $N$  头奶牛到达了机场 ( $1 \leq N \leq 10^5$ ), 其中奶牛  $i$  在时间  $t_i$  ( $0 \leq t_i \leq 10^9$ ) 到达。Farmer John安排了  $M$  ( $1 \leq M \leq 10^5$ ) 辆大巴来机场接这些奶牛。每辆大巴可以乘坐  $C$  头奶牛 ( $1 \leq C \leq N$ )。Farmer John正在机场等待奶牛们到来, 并且准备安排到达的奶牛们乘坐大巴。当最后一头乘坐某辆大巴的奶牛到达的时候, 这辆大巴就可以发车了。

Farmer John想要做一个优秀的主办者, 所以并不想让奶牛们在机场等待过长的时间。如果Farmer John合理地协调这些大巴, 等待时间最长的奶牛等待的时间的最小值是多少? 一头奶牛的等待时间等于她的到达时间与她乘坐的大巴的发车时间之差。

输入保证  $MC \geq N$ 。

## 输入格式

输入的第一行包含三个空格分隔的整数  $N$ ,  $M$  和  $C$ 。第二行包含  $N$  个空格分隔的整数, 表示每头奶牛到达的时间。

## 输出格式

输出一行, 包含所有到达的奶牛中的最大等待时间的最小值。

## 输入样例

```
6 3 2
1 1 10 14 4 3
```

## 输出样例

```
4
```

## 提示

如果两头时间1到达的奶牛乘坐一辆巴士, 时间2和时间4到达的奶牛乘坐乘坐第二辆, 时间10和时间14到达的奶牛乘坐第三辆, 那么等待时间最长的奶牛等待了4个单位时间 (时间10到达的奶牛从时间10等到了时间14)。

# B 吃草

时空限制：1s 256MB

文件名

grass.in/grass.out/grass.cpp

题意

虽然在接机上耽误了挺长时间, Farmer John为吃草爱好牛们举行的大会至今为止都非常顺利。大会吸引了世界各地的奶牛。

然而大会的重头戏看起来却给Farmer John带来了一些新的安排上的困扰。他的农场上的一块非常小的牧草地出产一种据某些识货的奶牛说是世界上最美味的品种的草。因此, 所有参会的  $N$  头奶牛 ( $1 \leq N \leq 10^5$ ) 都想要品尝一下这种草。由于这块牧草地小到仅能容纳一头奶牛, 这很有可能会导致排起长龙。

Farmer John知道每头奶牛计划到达这块特殊的牧草地的时间  $a_i$ , 以及当轮到她时, 她计划品尝这种草花费的时间  $t_i$ 。当奶牛  $i$  开始吃草时, 她会在离开前花费全部  $t_i$  的时间, 此时其他到达的奶牛需要排队等候。如果这块牧草地空出来的时候多头奶牛同时在等候, 那么资历最深的奶牛将会是下一头品尝鲜草的奶牛。需要说明的是, 恰好在另一头奶牛吃完草离开时到达的奶牛被认为是“在 等待的”。类似地, 如果当没有奶牛在吃草的时候有多头奶牛同时到达, 那么资历最深的奶牛是下一头吃草的奶牛。

请帮助FJ计算所有奶牛中在队伍里等待的时间 ( $a_i$  到这头奶牛开始吃草之间的时间) 的最大值。

输入格式

输入的第一行包含  $N$ 。以下  $N$  行按资历顺序给出了  $N$  头奶牛的信息 (资历最深的奶牛排在最前面)。每行包含一头奶牛的  $a_i$  和  $t_i$ 。所有的  $t_i$  为不超过  $10^4$  的正整数, 所有的  $a_i$  为不超过  $10^9$  的正整数。

输出格式

输出所有奶牛中的最长等待时间。

输入样例

```
5
25 3
105 30
20 50
10 17
100 10
```

输出样例

```
10
```

## 样例解释

在这个例子中，我们有5头奶牛（按输入中的顺序编号为1..5）。奶牛4最先到达（时间10），在她吃完之前（时间27）奶牛1和奶牛3都到达了。由于奶牛1拥有较深的资历，所以她先吃，从她到达开始共计等待了2个单位时间。她在时间30结束吃草，随后奶牛3开始吃草，从她到达开始共计等待了10单位时间。在一段没有奶牛吃草的时间过后，奶牛5到达，在她正在吃草的时间里奶牛2也到达了，在5个单位时间之后能够吃到草。相比于到达时间等待最久的奶牛是奶牛3。

## C 棋盘游戏

时空限制：1s 256MB

### 文件名

mooyo.in/mooyo.out/mooyo.cpp

### 题意

由于手上（更确实的，蹄子上）有大把的空余时间, Farmer John的农场里的奶牛经常玩电子游戏消磨时光。她们最爱的游戏之一是基于一款人类中流行的电子游戏Puyo Puyo的奶牛版; 名称当然叫做Mooyo Mooyo。 Mooyo Mooyo是在一块细长的棋盘上进行的, 高  $N(1 \leq N \leq 100)$  格, 宽10格。这是一个  $N = 6$  的棋盘的例子:

```
0000000000
0000000300
0054000300
1054502230
2211122220
1111111223
```

每个格子中或者是空的（用0表示），或者是九种颜色之一的干草捆（用字符1..9表示）。重力会使得干草捆下落，所以没有干草 捆的下方是 0。

如果两个格子水平或垂直方向直接相邻, 并且为同一种非0颜色, 那么这两个格子就属于同一个连通区域。任意时刻出现至少  $K$  个格子构成的连通区域, 其中的干草捆就会全部消失, 变为 0。如果同时出现多个这样的连通区域, 它们同时消失。随后, 重力可能会导致干草捆向下落入某个变为 0 的格子。由此形成的新的布局中, 又可能出现至少  $K$  个格子构成的连通区域。若如此, 它们同样也会消失（如果又有多个这样的区域, 则同时消失）, 然后重力又会使得剩下的方块下落, 这一过程持续进行, 直到不存在大小至少为  $K$  的连通区域为止。

给定一块Mooyo Mooyo棋盘的状态, 输出这些过程发生之后最终的棋盘的图案。

### 输入格式

输入的第一行包含  $N$  和  $K(1 \leq K \leq 10N)$ 。以下  $N$  行给出了棋盘的初始状态。

## 输出格式

输出  $N$  行, 描述最终的棋盘状态。

## 输入样例

```
6 3
0000000000
0000000300
0054000300
1054502230
2211122220
1111111223
```

## 输出样例

```
0000000000
0000000000
0000000000
0000000000
1054000000
2254500000
```

## 样例解释

在上面的例子中, 如果  $K = 3$ , 那么存在一个大小至少为  $K$  的颜色1的连通区域, 同样有一个颜色2的连通区域。当它们同时被移除 之后, 棋盘暂时成为了这样:

```
0000000000
0000000300
0054000300
1054500030
2200000000
0000000003
```

然后, 由于重力效果, 干草捆下落形成这样的布局:

```
0000000000
0000000000
0000000000
0000000000
1054000300
2254500333
```

再一次地, 出现了一个大小至少为KK的连通区域（颜色3）。移除这个区域就得到了最终的棋盘布局:

0000000000  
0000000000  
0000000000  
0000000000  
1054000000  
2254500000