

## NOIP2019 模拟赛 day 11（B 组）

比赛时间：2019 年 10 月 06 日 14: 00~17: 30

负责人：长乐一中潘恩宁

题目名称	剑	子段和	收金币	最大值
题目类型	传统型	传统型	传统型	传统型
提交程序名	sword.cpp	subsegment.cpp	gold.cpp	max.cpp
输入文件名	sword.in	subsegment.in	gold.in	max.in
输出文件名	sword.out	subsegment.out	gold.out	max.out
每个测试点时限	1s	1s	1s	4s
空间限制	512MB	512MB	512MB	512MB
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
是否有 Special Judge	否	否	否	否

### 【注意事项】

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C++中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
3. 提交时请将提交程序直接放在选手目录下，无需建立子文件夹。
4. 评测系统为 `lemon`，结果比较方式为忽略行末空格、文末回车后的全文比较。
5. 栈空间大小与题目的内存限制相同。
6. 题面中的样例以及附加的样例文件在下发的 `down` 文件夹中。

## 剑(sword,1s,512MB)

### 【题目描述】

两把剑进行比试。由于短剑武力值总比长剑高，所以每回合比试后，长剑总会被截去一段，截去的长度恰为短剑长度。当两把剑长度相等时，由于武力值相等，比试无法继续。

给出两把剑的初始长度，求两把剑的比试回合数。

### 【输入格式】

第一行包含一个整数 $T$ ，表示数据组数。

每组数据包含一行一个整数描述，表示两把剑的初始长度。

### 【输出格式】

输出一行一个整数表示总回合数。

### 【样例输入】

```
3
1 8
3 7
6 6
```

### 【样例输出】

```
7
4
0
```

### 【数据规模及约定】

对于40%的数据， $1 \leq T \leq 20$ ， $1 \leq a, b \leq 10^3$ 。

对于100%的数据， $1 \leq T \leq 10^3$ ， $1 \leq a, b \leq 10^{18}$ 。

## 子段和 (subsegment,1s,512MB)

### 【题目描述】

求环形序列的最大子段和。

### 【输入格式】

第一行一个整数 $n$ ，表示序列长度。

接下来 $n$ 行，第 $i$ 行包含一个整数 $a_i$ 。

### 【输出格式】

输出一行一个整数，表示最大子段和。

### 【样例输入】

```
5
1
2
-3
-4
3
```

### 【样例输出】

```
6
```

### 【数据规模及约定】

对于50%的数据， $n \leq 5 \times 10^3$ 。

对于100%的数据， $n, a_i \leq 10^6$ 。

## 收金币 (gold,1s,512MB)

### 【题目描述】

在一张 $n \times m$ 的网格中，分布着若干金币。

你希望通过一次从 $(1,1)$ 走到 $(n,m)$ （只能向右、向下走），能顺走网格中的所有金币。

求路径方案数，方案数对 $10^8$ 取模。

### 【输入格式】

第一行包含三个整数 $n, m, k$ ，表示网格大小和金币数。

接下来 $k$ 行，每行两个整数 $x, y$ ，表示一枚金币所在的位置。

### 【输出格式】

输出一行一个整数表示总方案数，方案数对 $10^8$ 取模。

### 【样例输入】

5 4 1

2 2

### 【样例输出】

20

### 【数据规模及约定】

对于30%的数据， $1 \leq n, m \leq 30$ ；

对于100%的数据， $1 \leq n, m \leq 3 \times 10^4$ ， $0 \leq k \leq 10^4$ ， $1 \leq x \leq n$ ， $1 \leq y \leq m$ 。

## 最大值 (max,4s,512MB)

### 【题目描述】

给你一个长度为 $n$ 的序列，有 $Q$ 个形如 $[l, r]$ 询问，表示求区间 $[l, r]$ 的最大值。

【输入格式】

第一行包含两个整数 $n, Q$ ，表示序列长度和询问个数。

第二行包含三个整数 $A, B, C$ 。

序列及询问生成方式如下：（序列为数组 $a[]$ ）

```
1.  int n, m, A, B, C;
2.  int read() {
3.      A = (1LL * A * B) ^ C;
4.      return (A & (n - 1)) + 1;
5.  }
6.  int main() {
7.      scanf("%d%d", &n, &m);
8.      scanf("%d%d%d", &A, &B, &C);
9.      for (int i = 1; i <= n; ++i)
10.         a[i] = read();
11.     for (int i = 1; i <= m; ++i) {
12.         int l = read(), r = read();
13.         if (l > r) swap(l, r);
14.     }
15. }
```

【输出格式】

假设第 $i$ 个询问的答案为 $ans_i$ ，那么你的输出即为：

$$\sum_{i=1}^Q ans_i \times B \bmod C$$

其中， $B, C$ 已在输入中给出。

【样例输入 1】

4 5  
32 17 19

【样例输出 1】

17

【样例输入 2】

8388608 8000000  
95 1071 1989

【样例输出 2】

153

【数据规模及约定】

数据编号	$n$	$Q$	数据编号	$n$	$Q$
1	8	8	6	65536	65000
2	64	50	7	262144	260000
3	128	100	8	1048576	1000000
4	1024	1000	9	8388608	7000000
5	4096	4800	10	8388608	8000000