

## 全国信息学奥林匹克联赛 (NOIP2011) 复赛

## 提高组 day2

( 请选手务必仔细阅读本页内容 )

## 一 . 题目概况

中文题目名称	计算系数	聪明的质监员	观光公交
英文题目与子目录名	factor	qc	bus
可执行文件名	factor	qc	bus
输入文件名	factor.in	qc.in	bus.in
输出文件名	factor.out	qc.out	bus.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	10	20	20
每个测试点分值	10	5	5
附加样例文件	有	有	有
结果比较方式	全文比较 ( 过滤行末空格及文末回车 )		
题目类型	传统	传统	传统

## 二 . 提交源程序文件名

对于 C++语言	factor.cpp	qc.cpp	bus.cpp
对于 C 语言	factor.c	qc.c	bus.c
对于 pascal 语言	factor.pas	qc. pas	bus. pas

## 三 . 编译命令 ( 不包含任何优化开关 )

对于 C++语言	g++ -o factor factor.cpp -lm	g++ -o qc qc.cpp -lm	g++ -o bus bus.cpp -lm
对于 C 语言	gcc -o factor factor.c -lm	gcc -o qc qc.c -lm	gcc -o bus bus.c -lm
对于 pascal 语言	fpc factor.pas	fpc qc.pas	fpc bus.pas

## 四．运行内存限制

内存上限	128M	128M	128M
------	------	------	------

## 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/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、全国统一评测时采用的机器配置为：CPU P4 3.0GHz，内存 1G，上述时限以此配置为准。
- 4、特别提醒：评测在 **NOI Linux** 下进行。

# 1 . 计算系数

(factor.cpp/c/pas)

## 【问题描述】

给定一个多项式  $(ax + by)^k$  , 请求出多项式展开后  $x^n y^m$  项的系数。

## 【输入】

输入文件名为 factor.in。

共一行, 包含 5 个整数, 分别为  $a, b, k, n, m$ , 每两个整数之间用一个空格隔开。

## 【输出】

输出文件名为 factor.out。

输出共 1 行, 包含一个整数, 表示所求的系数, 这个系数可能很大, 输出对 10007 取模后的结果。

## 【输入输出样例 1】

factor.in	factor.out
1 1 3 1 2	3

见选手目录下的 factor/factor1.in 和 factor/factor1.ans。

## 【输入输出样例 2】

见选手目录下的 factor/factor2.in 和 factor/factor2.ans。

## 【数据范围】

对于 30% 的数据, 有  $0 \leq k \leq 10$  ;

对于 50% 的数据, 有  $a = 1, b = 1$  ;

对于 100% 的数据, 有  $0 \leq k \leq 1,000, 0 \leq n, m \leq k$ , 且  $n + m = k, 0 \leq a, b \leq 1,000,000$ 。

## 2. 聪明的质监局

(qc.cpp/c/pas)

### 【问题描述】

小 T 是一名质量监督员，最近负责检验一批矿产的质量。这批矿产共有  $n$  个矿石，从 1 到  $n$  逐一编号，每个矿石都有自己的重量  $w_i$  以及价值  $v_i$ 。检验矿产的流程是：

- 1、给定  $m$  个区间  $[L_i, R_i]$ ；
- 2、选出一个参数  $W$ ；
- 3、对于一个区间  $[L_i, R_i]$ ，计算矿石在这个区间上的检验值  $Y_i$ ：

$$Y_i = \sum_j 1 * \sum_j v_j, j \in [L_i, R_i] \text{ 且 } w_j \geq W, j \text{ 是矿石编号}$$

这批矿产的检验结果  $Y$  为各个区间的检验值之和。即： $Y = \sum_{i=1}^m Y_i$

若这批矿产的检验结果与所给标准值  $S$  相差太多，就需要再去检验另一批矿产。小 T 不想费时间去检验另一批矿产，所以他想通过调整参数  $W$  的值，让检验结果尽可能的靠近标准值  $S$ ，即使得  $S-Y$  的绝对值最小。请你帮忙求出这个最小值。

### 【输入】

输入文件 qc.in。

第一行包含三个整数  $n, m, S$ ，分别表示矿石的个数、区间的个数和标准值。

接下来的  $n$  行，每行 2 个整数，中间用空格隔开，第  $i+1$  行表示  $i$  号矿石的重量  $w_i$  和价值  $v_i$ 。

接下来的  $m$  行，表示区间，每行 2 个整数，中间用空格隔开，

第  $i+n+1$  行表示区间  $[L_i, R_i]$  的两个端点  $L_i$  和  $R_i$ 。注意：不同区间可能重合或相互重叠。

### 【输出】

输出文件名为 qc.out。输出只有一行，包含一个整数，表示所求的最小值。

**【输入输出样例 1】**

qc.in	qc.out
5 3 15 1 5 2 5 3 5 4 5 5 5 1 5 2 4 3 3	10

见选手目录下的 qc/qc1.in 和 qc/qc1.ans。

**【输入输出样例 1 说明】**

当 W 选 4 的时候，三个区间上检验值分别为 20、5、0，这批矿产的检验结果为 25，此时与标准值 S 相差最小为 10。

**【输入输出样例 2】**

见选手目录下的 qc/qc2.in 和 qc/qc2.ans。

**【数据范围】**

对于 10%的数据，有  $1 \leq n, m \leq 10$ ；

对于 30%的数据，有  $1 \leq n, m \leq 500$ ；

对于 50%的数据，有  $1 \leq n, m \leq 5,000$ ；

对于 70%的数据，有  $1 \leq n, m \leq 10,000$ ；

对于 100%的数据，有  $1 \leq n, m \leq 200,000, 0 < w_i, v_i \leq 10^6, 0 < S \leq 10^{12}, 1 \leq L_i \leq R_i \leq n$ 。

## 任务(task)

时间限制: 1.000 Sec 内存限制: 128 MB

### 题目描述

现在又两台机器 $A$ 和 $B$ 。有 $n$ 个任务，编号 $1, 2, \dots, n$ 。你必须把每个任务安排到一台机器上处理时需要满足以下一些条件。

- 1.你必须把每个任务安排到任意一台机器上处理。
  - 2.在任何时刻，一台机器只能最多处理一个人任务。
  - 3.任务 $i(0 < i < n)$ 可以被处理当前仅当每个任务 $j(j < i)$ 已经被完成或者正在进行。
  - 4.一个任务如果在一台机器上进行，它是不能被打断的。
- 请你算算最少完成任务的时间。

### 输入

第一行一个整数 $n$ 表示任务的个数。

接下来 $n$ 行，每行两个整数 $t_A, t_B$ 表示完成每个任务在两台机器上花的时间。

### 输出

输出最早完成任务的时间。

### 样例输入

```
2
1 2
90 95

3
1 3
1 3
1 3
```

### 样例输出

```
90

3
```

### 提示

样例解释：

第一组数据：让 $B$ 机器完成任务1同时让 $A$ 机器完成任务2。

第二组数据：让 $A$ 机器完成所有的任务。

对于30%的数据， $n$ 的范围 $[1, 20]$ ， $0 < t_A, t_B < 30$ ；

对于70%的数据， $n$ 的范围 $[1, 200]$ ， $0 < t_A, t_B < 300$ ；

对于100%的数据， $n$ 的范围 $[1, 2000]$ ， $0 < t_A, t_B < 3000$ ；

### 3. 观光公交

( bus.cpp/c/pas )

#### 【问题描述】

风景迷人的小城 Y 市，拥有  $n$  个美丽的景点。由于慕名而来的游客越来越多，Y 市特意安排了一辆观光公交车，为游客提供更便捷的交通服务。观光公交车在第 0 分钟出现在 1 号景点，随后依次前往 2、3、4…… $n$  号景点。从第  $i$  号景点开到第  $i+1$  号景点需要  $D_i$  分钟。任意时刻，公交车只能往前开，或在景点处等待。

设共有  $m$  个游客，每位游客需要乘车 1 次从一个景点到达另一个景点，第  $i$  位游客在  $T_i$  分钟来到景点  $A_i$ ，希望乘车前往景点  $B_i$  ( $A_i < B_i$ )。为了使所有乘客都能顺利到达目的地，公交车在每站都必须等待需要从该景点出发的所有乘客都上车后才能出发开往下一景点。假设乘客上下车不需要时间。一个乘客的旅行时间，等于他到达目的地的时刻减去他来到出发地的时刻。因为只有一

辆观光车，有时候还要停下来等其他乘客，乘客们纷纷抱怨旅行时间太长了。于是聪明的司机 ZZ 给公交车安装了  $k$  个氮气加速器，每使用一个加速器，可以使其中一个  $D_i$  减 1。对于同一个  $D_i$  可以重复使用加速器，但是必须保证使用后  $D_i$  大于等于 0。

那么 ZZ 该如何安排使用加速器，才能使所有乘客的旅行时间总和最小？

#### 【输入】

输入文件名为 bus.in。

第 1 行是 3 个整数  $n, m, k$ ，每两个整数之间用一个空格隔开。分别表示景点数、乘客数和氮气加速器个数。

第 2 行是  $n-1$  个整数，每两个整数之间用一个空格隔开，第  $i$  个数表示从第  $i$  个景点开往第  $i+1$  个景点所需要的时间，即  $D_i$ 。

第 3 行至  $m+2$  行每行 3 个整数  $T_i, A_i, B_i$ ，每两个整数之间用一个空格隔开。第  $i+2$  行表示第  $i$  位乘客来到出发景点的时刻，出发的景点编号和到达的景点编号。

#### 【输出】

输出文件名为 bus.out。

共一行，包含一个整数，表示最小的总旅行时间。

### 【输入输出样例 1】

bus.in	bus.out
3 3 2 1 4 0 1 3 1 1 2 5 2 3	10

见选手目录下的 bus/bus1.in 和 bus/bus1.ans。

### 【输入输出样例 1 说明】

对  $D_2$  使用 2 个加速器，从 2 号景点到 3 号景点时间变为 2 分钟。

公交车在第 1 分钟从 1 号景点出发，第 2 分钟到达 2 号景点，第 5 分钟从 2 号景点出发，第 7 分钟到达 3 号景点。

第 1 个旅客旅行时间  $7-0=7$  分钟。

第 2 个旅客旅行时间  $2-1=1$  分钟。

第 3 个旅客旅行时间  $7-5=2$  分钟。

总时间  $7+1+2=10$  分钟。

### 【输入输出样例 2】

见选手目录下的 bus/bus2.in 和 bus/bus2.ans。

### 【数据范围】

对于 10% 的数据， $k=0$ ；

对于 20% 的数据， $k=1$ ；

对于 40% 的数据， $2 \leq n \leq 50$ ， $1 \leq m \leq 1,000$ ， $0 \leq k \leq 20$ ，

$0 \leq D_i \leq 10$ ， $0 \leq T_i \leq 500$ ；

对于 60% 的数据， $1 \leq n \leq 100$ ， $1 \leq m \leq 1,000$ ， $0 \leq k \leq 100$ ，

$0 \leq D_i \leq 100$ ， $0 \leq T_i \leq 10,000$ ；

对于 100% 的数据， $1 \leq n \leq 1,000$ ， $1 \leq m \leq 10,000$ ， $0 \leq k \leq 100,000$ ，

$0 \leq D_i \leq 100$ ， $0 \leq T_i \leq 100,000$ 。