

# 2018 年“创新杯”小小程序员技能展示活动

## 第二场（4 月 1 日 9:00~11:00）

（请选手务必仔细阅读本页内容）

### 一.题目概况

中文题目名称	买苹果	扫雷	数字变换	跳一跳
英文题目名称	apple	mine	number	jump
可执行文件名	apple	mine	number	jump
输入文件名	apple.in	mine.in	number.in	jump.in
输出文件名	apple.out	mine.out	number.out	jump.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
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	传统

### 二.提交源程序文件名

对于 C 语言	apple.c	mine.c	number.c	jump.c
对于 C++语言	apple.cpp	mine.cpp	number.cpp	jump.cpp

### 三.编译命令（不含任何优化开关）

对于 C 语言	gcc -o apple apple.c -lm	gcc -o mine mine.c -lm	gcc -o number number.c -lm	gcc -o jump jump.c -lm
对于 C++语言	g++ -o apple apple.cpp -lm	g++ -o mine mine.cpp -lm	g++ -o number number.cpp -lm	g++ -o jump jump.cpp -lm

### 四.运行内存限制

运行内存上限	128M	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、评测在 windows 下用 lemon 进行，不需要建立子文件夹。
- 4、提交时将所写程序放入一个文件夹内，文件夹命名为“编号+姓名”，如“01+张三”。

# 1.买苹果

(apple.cpp/c)

## 【问题描述】

假设商店有三种昂贵的苹果，价格分别为 15 元每斤、20 元每斤和 35 元每斤，并且该商店卖东西不会找零，多的都算小费。若用一张面额为 N 元的钞票来买苹果，那么可能会有很多种购买的方案，请找出在这些方案中给出小费最少的情况，并输出此时小费的值。

**钞票面额为 40 元时，有如下购买组合：**

- ① 35 元每斤的买 1 斤，此时商品价格为 35 元，小费为 5 元；
- ② 20 元每斤的买 1 斤，此时商品价格为 20 元，小费为 20 元；
- ③ 15 元每斤的买 1 斤，此时商品价格为 15 元，小费为 25 元；
- ④ 15 元每斤的买 1 斤，20 元每斤的买 1 斤，此时商品价格为 35 元，小费为 5 元；
- ⑤ 15 元每斤的买 2 斤，此时商品价格为 30 元，小费为 10 元；
- ⑥ 20 元每斤的买 2 斤，此时商品价格为 40 元，小费为 0 元。

所以钞票面额为 40 元时，选用第六种购买方案，小费为 0。

**钞票面额为 25 元时，有如下购买方案：**

- ① 15 元每斤的买 1 斤，此时商品价格为 15 元，小费为 10 元；
- ② 20 元每斤的买 1 斤，此时商品价格为 20 元，小费为 5 元。

所以钞票面额为 25 元时，选用第二种购买方案，小费为 5。

## 【输入格式】

从文件 apple.in 中读入数据。

输入的第一行是一个整数 T，代表测试数据的组数。接下来是 T 行测试数据，每行测试数据包含一个正整数 N，N 代表钞票的面值。

## 【输出格式】

输出到文件 apple.out 中。

对于每组测试数据，输出最小的费用数；如果完全买不起，输出-1。

## 【样例输入】

2  
40  
25

## 【样例输出】

0  
5

## 【数据规模】

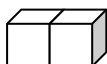
$1 \leq T \leq 100$ ,  $1 \leq N \leq 100000$ 。

## 2.扫雷

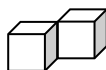
(mine.cpp/c)

### 【问题描述】

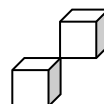
经典的扫雷游戏升级了。在  $n$  行  $m$  列  $s$  层的雷区中有  $k$  个地雷，我们把存在地雷的格子称为地雷格，剩下的称为非地雷格。若两格共点或者共线或者共面，则这两格互为相邻格。（即在一个  $3*3*3$  的区域中，对于正中间的格子而言，其余格均是它的相邻格）



(共面)



(共线)



(共点)

玩家在翻开非地雷格时，该格会出现一个字符来提示相邻格中有多少个是地雷格，字符 '0'、'a'~'z' 分别对应数字 0、1~26。若一个非地雷格上的字符为 'a'，则表示该格的相邻格中存在 1 个地雷格；若一个非地雷格上的字符为 'z'，则表示该格的相邻格中存在 26 个地雷格；若一个非地雷格上的字符为 '0'，则表示该格的相邻格中没有地雷格。

现在给出雷区中所有地雷格的坐标，要求计算出每个非地雷格周围的地雷格数，并从第一层开始，用字符逐层描述每一层的状态。

### 【输入格式】

从文件 mine.in 中读入数据。

输入的第一行包含四个空格隔开的正整数  $n$ 、 $m$ 、 $s$  和  $k$ ，表示雷区的行数、列数、层数和地雷个数。接下来  $k$  行，每行三个空格隔开的正整数  $x$ 、 $y$  和  $z$ ，表示地雷的坐标。

### 【输出格式】

输出到文件 mine.out 中。

输出包括  $n*s$  行，每行包括  $m$  个字符，描述整个雷区。用 '\*' 表示地雷格，用周围的地雷个数表示非地雷格。相邻字符之间无间隔符。

其中，第  $1\sim n$  行描述的第 1 层，第  $n+1\sim n+n$  行描述的第 2 层，以此类推。

### 【样例输入】

```
3 3 2 1
1 1 1
```

### 【样例输出】

```
*a0
aa0
000
aa0
aa0
000
```

### 【数据规模】

对于 30% 的数据， $s=1$ ；对于 100% 的数据， $1\leq n,m\leq 100$ ， $1\leq s\leq 20$ ， $1\leq k\leq 10000$ 。

### 3.数字变换

(number.cpp/c)

#### 【问题描述】

有这样一类特殊的数，这类数满足以下特征：

①数字 1 属于这类数；

②若  $P$  属于这类数，则  $A * P + B, C * P + D$  也属于这类数（这里的  $A、B、C、D$  均为 1~9 之间的整数）。

现在将这类数中最小的  $K$  个数从小到大依次相连凑成一个数  $x$ 。若  $x$  的位数为奇数，则从第 1 位开始（包括第 1 位），每隔一个数字删掉一个数字；若  $x$  的位数为偶数，则从第 2 位开始（包括第 2 位），每隔一个数字删掉一个数字。

请问最后剩下的数是多少？

说明：这里所指的第一位、第二位是从数的最高位数起，即数字 789 的第 1 位是 7，第 2 位是 8。

（假设这类数中的数从小到大排列依次为 1,5,7,13,17,……，我们要在这类数中取最小的 5 个数相连，便会生成数字 1571317，该数字长度为 7，是奇数。因此删掉第 1、3、5、7 位上的数，剩下的数为 511。）

#### 【输入格式】

从文件 number.in 中读入数据。

输入内容为 5 个正整数，依次为  $K、A、B、C、D$ 。

#### 【输出格式】

输出到文件 number.out 中。

输出按照题目要求求得的数字。

#### 【样例输入】

5 3 4 2 3

#### 【样例输出】

511

#### 【数据规模】

对于 10% 的数据， $A=B=C=D=1$ ；

对于 30% 的数据， $A=C, B=D$ ；

对于 100% 的数据， $0 < A, B, C, D < 10, K \leq 100000$ 。

数据保证这类数中的前  $K$  小的数一定都小于  $2^{63}-1$ 。

## 4.跳一跳

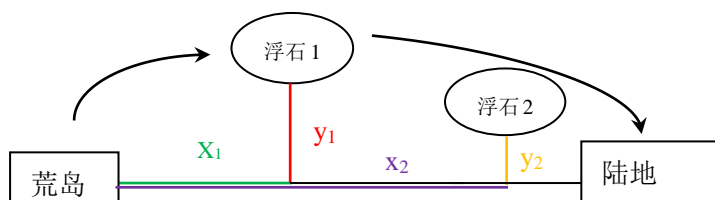
(jump.cpp/c)

### 【问题描述】

一款名叫“跳一跳”的游戏上线了！

游戏开始时，玩家在一个荒岛上，我们把荒岛的位置记为出发点。玩家的目的地是陆地，陆地到荒岛的距离为  $5 \times 10^5 + 1$ 。荒岛和陆地相对于海平面的高度均为 0。现在，荒岛和陆地之间有一条由空中浮石组成的笔直通道，玩家在从荒岛回到陆地的过程中，可以跳到这些浮石上停留，每个浮石最多只能停留一次。第一次停留在浮石上记 1 分，之后每次停留在浮石上的得分均为上次得分的 2 倍。跳到陆地时没有新的加分（即如果只跳到过一个浮石，最终得分为 1 分）。

每块浮石具有两个关键信息，分别用  $x$  和  $y$  表示， $x$  代表该浮石到荒岛的水平距离， $y$  代表该浮石到海平面的垂直距离，如图所示：



（此情况下两种跳法：荒岛-浮石 1-陆地；荒岛-浮石 2-陆地。得分最多为 1 分。）

玩家的跳跃能力十分卓越，所以可以从一个位置准确地跳到任意其他位置，且跳跃过程中不会受到干扰。游戏规定：

- ①只能从荒岛向陆地方向跳；
- ②如果前一次是往高处跳，下一次就只能往低处跳；
- ③如果前一次是往低处跳，下一次就只能往高处跳。

（若现在向高处跳到了浮石  $i$  上，那么下一个位置的浮石  $j$  必须满足  $x_j > x_i$ ,  $y_j < y_i$ ；若现在向低处跳到了的浮石  $i$  上，那么下一个位置的浮石  $j$  必须满足  $x_j > x_i$ ,  $y_j > y_i$ ；第一次跳一定是向高处跳；最后一次跳一定是向低处跳。）

现给定所有浮石的信息，请你计算出可以得到的最高分数是多少。由于答案可能很大，所以只需要输出答案对 1000007 取模的结果。

### 【输入格式】

从文件 jump.in 中读入数据。

第一行为一个正整数  $n$  ( $0 < n \leq 5 \times 10^5$ )，代表浮石的个数。接下来的  $n$  行每行有 2 个整数，分别表示浮石的  $x$  值 ( $0 < x \leq 5 \times 10^5$ ) 与  $y$  值 ( $0 < y < 100$ )。

### 【输出格式】

输出到文件 jump.out 中。

输出一个整数，表示最终的结果。

**【样例输入】**

5  
4 1  
3 2  
1 5  
5 2  
2 3

**【样例输出】**

7

**【数据规模】**

对于 10% 的数据，输入的  $x$  按照从小到大的顺序排列；

对于 30% 的数据， $0 < n \leq 20$ ；

对于 100% 的数据， $0 < n \leq 5 \times 10^5$ ,  $0 < x \leq 5 \times 10^5$ ,  $0 < y < 100$ 。