

# 2015 年《程序设计导论》期末考试

(考试时间：2015 年 12 月 31 日)

1. 本次测试的时间为 150 分钟；编程结果采用机器自动评测。
2. 本次考试共 6 题，完成 5 题即可满分。注意：其中 1-4 题为必做题目。得分从高到低每题的分值分别为 30、25、20、15、10、附加 10 分。
3. 程序设计语言选用 C 或 C++。
4. 所有题目的时间限制均为 1 秒。

## 题目索引

1. 统计时间 .....	2
2. 数据挖掘 .....	3
3. 矩阵搜索 .....	4
4. 链表操作 .....	5
5. 数据表排序 .....	6
6. 数列合并 .....	8

## 1. 统计时间

### 【问题描述】

信息学院大一学生都要学习《程序设计导论》，但是，每人每天用在学习程序设计的时间不尽相同，编写一个程序统计每天用于学习编程的最长时间、最短时间和平均时间。

### 【输入格式】

输入 2 行，第一行为输入  $n$ ，表示学生的总人数。 $n$  为整数，取值范围是： $1 \leq n \leq 200$ 。

第二行共  $n$  个数，数据之间用一个空格分隔，最后以回车结尾。数据为学生每天用于学习程序设计的时间  $t$ 。 $t$  为整数，以分钟为单位。

### 【输出格式】

输出一行，共有 3 个数，依次是**最长时间**、**最短时间**和**平均时间**，两个整数之间以一个空格分隔，回车结束。时间都是以分钟为单位，输出时间数据按四舍五入保留整数。

### 【样例输入 1】

```
3
100 50 300
```

### 【样例输出 1】

```
300 50 150
```

### 【样例输入 2】

```
5
123 200 30 360 170
```

### 【样例输出 2】

```
360 30 177
```

## 2. 数据挖掘

### 【问题描述】

新生聚会上，要求每名学生按照例句用英文介绍个人情况，例句如下：

My name is Li Ming, born in 1995, from Yunnan Province, Shangri La county.  
My major is computer science and technology. I like swimming.

也就是说用个人的信息替换例句中划线的内容。现在请你编程从每人的自我介绍中截取每人的生日与主修专业。

### 【输入格式】

1 行，为一名学生的英文自我介绍，长度不超 200 个字符。

### 【输出格式】

1 行，包含两个数据：生日和主修专业，之间由一个逗号和一个空格分隔，回车结尾。

### 【样例输入】

My name is Li Ming, born in 1995, from Yunnan Province, Shangri La country. My major is computer science and technology. I like swimming.

### 【样例输出】

1995, computer science and technology

### 【注意】

文中的标点符号后有一个空格，如：“Li Ming, born”中在逗号与其后的单词“born”之间有一个空格。

### 3. 矩阵搜索

#### 【问题描述】

有一个  $m$  行  $n$  列的正整数矩阵，请你找出一个位置，满足下面的条件：该位置上的数字与其相邻位置上的数字之和最大。注意：相邻的位置是指有公共边的 2 个位置。如果有多个位置可以得到同一个最大值，则按顺序输出这多个位置。

例如：下面的矩阵中，与黄色格子相邻的 4 格子已经用特殊的颜色标出。注意，靠边和角落上格子的相邻格子分别是 3 个和 2 个。

1	7	4	0
9	4	8	8
2	4	5	5
1	7	1	1

#### 【输入格式】

第 1 行，两个整数  $m$ 、 $n$ ，分别表示矩阵的行、列数。其中， $1 \leq m, n \leq 100$ 。

从第 2 行起，按行输入矩阵。

以空格分隔两个整数，以回车结束一行。

#### 【输出格式】

输出若干行，每行两个数，以空格分隔。

第 1 行的第 1 个数为最大值；第 2 个数为最大值个数。

从第 2 行起，每行表示一个位置，表示符合条件的位置的行、列坐标（行、列均从 0 开始编号），请按照位置的字典序输出。

#### 【样例输入 1】

```
4 4
1 7 4 0
9 4 8 8
2 4 5 5
1 7 1 1
```

#### 【样例输出 1】

```
32 1
1 1
```

#### 【样例输入 2】

```
4 6
1 7 4 0 9 4
0 7 0 4 5 5
1 7 1 1 0 0
0 7 2 0 0 0
```

#### 【样例输出 2】

```
23 2
1 4
2 1
```

## 4. 链表操作

### 【问题描述】

编写程序完成单链表的三种基本操作，三种操作的命令如下：

**C n m**

在链表第  $n$  个节点之后创建一个值为  $m$  的链表节点，如果  $n$  为 0 则表示在表头创建，显然第一次操作的  $n=0$ ；

**D n**

在链表中删除第  $n$  个节点，数据保证该节点是存在的；

**P n**

输出链表中第  $n$  个节点的值，数据保证该节点是存在的。

**注意：**本题为完成函数题，需要完成 2 个函数，分别是用于插入节点和删除节点。打印的节点的函数已给出。

### 【输入格式】

第一行，包含一个正整数  $K$ ，表示操作的个数。

接下来的  $K$  行，每行表示一个操作，按照描述的格式给出。

### 【输出格式】

若干行，每一次  $P$  操作，输出一行。

### 【样例输入】

```
5
C 0 1
C 1 2
P 1
D 1
P 1
```

### 【样例输出】

```
1
2
```

### 【数据规模】

$K \leq 1000$

## 5. 数据表排序

### 【问题描述】

有一个二维数据表，表中均为整数，表中的第一列的数字没有重复，可以用来唯一标识一行数据。

现在希望你以行为单位对这个二维数据表进行排序，排序的依据是某一列或者某几列。

列 1	列 2	列 3	列 4	列 5
1	1	7	4	0
2	9	4	8	8
3	2	4	5	5
4	1	7	1	1
5	2	2	7	6

### 【输入格式】

第 1 行，两个整数  $m$ 、 $n$  ( $2 \leq m, n < 100$ )，分别表示数据表的行、列数。

第 2 行，若干个整数，其中第 1 个整数  $k$  表示作为排序依据的列的个数。接下来  $k$  个整数，依次是作为排序依据的列的列号（列号从 1 开始），按照优先级从高到低的顺序给出。例如 2 2 3 表示按照第 2 列、第 3 列进行排序，也就是说第 2 列上数字相等的时候，再按照第 3 列上数字进行排序。**特别注意：输入的排序依据列号不会包括 1，但当按照给出的排序依据列无法确定数据表中两行的大小关系的时候，默认按照第 1 列排序。**

第 3 行起的  $m$  行，每行  $n$  个整数，表示二维数据表中一行的数据，每 2 个整数之间用一个空格隔开。

### 【输出格式】

$m$  行，每行  $n$  个整数，为排序后的  $m$  行  $n$  列数据表。

### 【样例输入 1】

```
5 5
1 2
1 1 7 4 0
2 9 4 8 8
3 2 4 5 5
4 1 7 1 1
5 2 2 7 6
```

### 【样例输出 1】

```
1 1 7 4 0
4 1 7 1 1
3 2 4 5 5
5 2 2 7 6
2 9 4 8 8
```

**【样例输入 2】**

5 5  
2 2 3  
1 1 7 4 0  
2 9 4 8 8  
3 2 4 5 5  
4 1 7 1 1  
5 2 2 7 6

**【样例输出 2】**

1 1 7 4 0  
4 1 7 1 1  
5 2 2 7 6  
3 2 4 5 5  
2 9 4 8 8

**【样例说明】**

两个样例的实际排序字段列表分别为 2、1 和 2、3、1。也就说当第 2 行所有指定的排序字段的值都相等时，再按第 1 列的值排序。

**【数据说明】**

有 50% 的测试数据，作为排序依据的列只有 1 个（不含第 1 列，下同）。

有 75% 的测试数据，作为排序依据的列不超过 3 个。

## 6. 数列合并

### 【问题描述】

给定一个  $1\sim N$  的排列，每次将相邻两个数相加，得到新序列，再对新序列重复这样的操作，显然每次得到的序列都比上一次的序列长度少 1，最终只剩一个数字。

例如：

3 1 2 4

4 3 6

7 9

16

现在如果知道  $N$  和最后得到的数字  $sum$ ，请求出最初的序列。若有多种答案，则输出字典序最小的那一个。数据保证有解。

### 【输入格式】

第 1 行为两个正整数，分别表示  $N$  ( $0 < N \leq 10$ ) 和  $sum$ 。

### 【输出格式】

一行，即所求的复合条件的  $1\sim N$  的一个排列，每 2 个整数之间用一个空格隔开。

### 【样例输入】

4 16

### 【样例输出】

3 1 2 4

### 【说明】

$0 < n \leq 10$