

## 南京市小学生信息学竞赛初赛编程模拟卷（二）

姓名：

成绩：

### 1. 接水问题 (water.cpp)

#### 【问题描述】

学校里有一个水房，水房里一共装有  $m$  个龙头可供同学们打开水，每个龙头每秒的供水量相等，均为 1。

现在有  $n$  名同学准备接水，他们的初始接水顺序已经确定。将这些同学按接水顺序从 1 到  $n$  编号， $i$  号同学的接水量为  $w_i$ 。接水开始时，1 到  $m$  号同学各占一个水龙头，并同时打开水龙头接水。当其中某名同学  $j$  完成其接水量要求  $w_j$  后，下一名排队等候接水的同学  $k$  马上接替  $j$  同学的位置开始接水。这个换人的过程是瞬间完成的，且没有任何水的浪费。亦即， $j$  同学第  $x$  秒结束时完成接水，则  $k$  同学第  $x+1$  秒立刻开始接水。若当前接水人数  $n'$  不足  $m$ ，则只有  $n'$  个龙头供水，其他  $m - n'$  个龙头关闭。

现在给出  $n$  名同学的接水量，按照上述接水规则，问所有同学都接完水需要多少秒。

#### 【输入】

输入文件名为 `water.in`。

第 1 行 2 个整数  $n$  和  $m$ ，用一个空格隔开，分别表示接水人数和龙头个数。

第 2 行  $n$  个整数  $w_1, w_2, \dots, w_n$ ，每两个整数之间用一个空格隔开， $w_i$  表示  $i$  号同学的接水量。

#### 【输出】

输出文件名为 `water.out`。

输出只有一行，1 个整数，表示接水所需的总时间。

#### 【输入输出样例1】

water.in	water.out
5 3	4
4 4 1 2 1	

#### 【样例1说明】

第1秒，3人接水。第1秒结束时，1、2、3号同学每人的已接水量为1，3号同学接完水，4号同学接替3号同学开始接水。

第2秒，3人接水。第2秒结束时，1、2号同学每人的已接水量为2，4号同学的已接水量为1。

第3秒，3人接水。第3秒结束时，1、2号同学每人的已接水量为3，4号同学的已接水量为2。4号同学接完水，5号同学接替4号同学开始接水。

第4秒，3人接水。第4秒结束时，1、2号同学每人的已接水量为4，5号同学的已接水量为1。1、2、5号同学接完水，即所有人完成接水。

总接水时间为4秒。

#### 【输入输出样例2】

water.in	water.out
8 4	163

23 71 87 32 70 93 80 76	
-------------------------	--

**【数据范围】**

$1 \leq n \leq 10000, 1 \leq m \leq 100$  且  $m \leq n$ ;

$1 \leq w_i \leq 100$ 。

## 2. 字符串解压缩 (string.cpp)

**【问题描述】**

观察这个字符串 'AAAABCCCCDDDD'，它均由拉丁大写字母组成，其长度为 14，我们可以把这个字符串压缩成 '4AB5C4D'，压缩后长度为 7，压缩规则是：假设某连续出现的同一字母的数量为  $n$ ，则其在压缩字符串中为 ' $n$  字母'，若  $n=1$  则  $n$  必须省略。因为倘若 '1' 不省略，则字符串 'AABCDE' 会被压缩成 '2A1B1C1D1E'，长度反而变大了。

编一程序：

1. 从文件 STRING.IN 中读入被压缩的字符串；
2. 求出原来的字符串；
3. 把结果写入文件 STRING.OUT 中。

**【输入】** (string.in)

输入文件仅有一行，为一个压缩过的字符串。

**【输出】** (string.out)

输出原字符串（字符串可能会很长，每 40 个字符换行）。

**【输入样例】**

3A4B7D

**【输出样例】**

AAABBBBDDDDDDDD

## 3. 顺序的分数 (frac.cpp)

**【问题描述】**

输入一个自然数  $N$  ( $1 \leq N \leq 160$ )，请写一个程序来按递增序输出分母小于等于  $N$  的已约分的真分数。

**【输入】** (frac.in)

单独的一行，一个自然数  $N$ 。

**【输出】** (frac.out)

每个分数单独占一行。

**【输入样例】**

5

**【输出样例】**

0/1

1/5

1/4

1/3

2/5

1/2

3/5  
2/3  
3/4  
4/5  
1/1

4. 排座椅 (seat.cpp)

【问题描述】

上课的时候，总有一些同学和前后左右的人交头接耳，这是令小学班主任十分头疼的一件事情。不过，班主任小雪发现了一些有趣的现象：当同学们的座次确定下来之后，只有有限的  $D$  对同学上课时 would 交头接耳。同学们在教室中坐成了  $M$  行  $N$  列，坐在第  $i$  行第  $j$  列的同学的位置是  $(i, j)$ ，为了方便同学们进出，在教室中设置了  $K$  条横向的通道、 $L$  条纵向的通道。于是，聪明的小雪想到了一个办法，或许可以减少上课时学生交头接耳的问题：她打算重新摆放桌椅，改变同学们桌椅间通道的位置，因为如果一条通道隔开了两个会交头接耳的同学，那么他们就不会交头接耳了。

请你帮忙给小雪编写一个程序，给出最好的通道划分方案。在该方案下，上课时交头接耳的学生的对数最少。

【输入】

输入文件 seat.in 的第一行，有 5 个空格隔开的整数，分别是  $M$ 、 $N$ 、 $K$ 、 $L$ 、 $D$  ( $2 \leq N$ ,  $M \leq 1000$ ,  $0 \leq K < M$ ,  $0 \leq L < N$ ,  $D \leq 2000$ )。

接下来的  $D$  行，每行有 4 个用空格隔开的整数。第  $i$  行的 4 个整数  $X_i$ 、 $Y_i$ 、 $P_i$ 、 $Q_i$ ，表示坐在位置  $(X_i, Y_i)$  与  $(P_i, Q_i)$  的两个同学会交头接耳（输入保证他们前后相邻或者左右相邻）。

输入数据保证最优方案的唯一性。

【输出】

输出文件 seat.out 共两行。

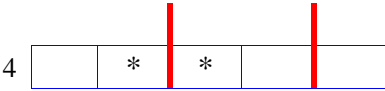
第一行包含  $K$  个整数  $a_1 \ a_2 \ \dots \ a_k$ ，表示第  $a_1$  行和  $a_1+1$  行之间、第  $a_2$  行和  $a_2+1$  行之间、 $\dots$ 、第  $a_k$  行和第  $a_k+1$  行之间要开辟通道，其中  $a_i < a_i+1$ ，每两个整数之间用空格隔开（行尾没有空格）。

第二行包含  $L$  个整数， $b_1 \ b_2 \ \dots \ b_L$ ，表示第  $b_1$  列和第  $b_1+1$  列之间、第  $b_2$  列和第  $b_2+1$  列之间、 $\dots$ 、第  $b_L$  列和第  $b_L+1$  列之间要开辟通道，其中  $b_i < b_i+1$ ，每两个整数之间用空格隔开（行尾没有空格）。

【输入输出样例】

seat.in	Seat.out
4 5 1 2 3	2
4 2 4 3	2 4
2 3 3 3	
2 5 2 4	

【输入输出样例解释】



3			※		
2			※	+	+
1					
	1	2	3	4	5

上图中用符号\*、※、+标出了 3 对会交头接耳的学生的位置，图中 3 条粗线的位置表示通道，图示的通道划分方案是唯一的最佳方案。

## 5. 货币系统 (money.cpp)

### 【问题描述】

一般来说，一个货币系统是由 1、5、10、20 或 25、50 和 100 的单位面值组成的。举例来说，若一个货币系统为  $\{1, 2, 5, 10, \dots\}$ ，要产生 18 这一面值，一些可能的方法是：18\*1、9\*2、8\*2+2\*1、3\*5+2+1，等等。

请你写一个程序，对于一个给定的货币系统，计算一下有多少种方法来构造某一面值。输入数据保证总的方法数不超过 long。

### 【输入】 (money.in)

第 1 行是两个整数  $V$  和  $N$ ，其中  $V (1 \leq V \leq 25)$  表示货币系统中货币的种类数目， $N (1 \leq N \leq 10000)$  表示要构造的面额。

接下来第 2 到第  $V+1$  行中，每行一个整数，给出了各个可用的货币。

### 【输出】 (money.out)

可能的方案数。

### 【样例输入】

```
3 10
1
2
5
```

### 【样例输出】

```
10
```