

2018-2019 秋季学期程序设计 I 期中测试

中国人民大学 信息学院

注意事项

1. 本次测试的时间为 180 分钟；编程结果采用机器自动评测。
2. 共有 5 题，第 1、2 题，25 分；第 3 题，20 分；第 4、5 题，15 分。
3. 提交到在线评测系统中的程序均采用标准输入和标准输出（键盘输入和屏幕输出）。
4. 程序设计语言选用 C 或 C++。
5. 所有题目的时间限制均为 1s。

1. 拆分数字

时间限制：1.0s; 内存限制：256.0MB

拆分数字

给出一个不多于4位的十进制非负整数N，求它是几位数，并按个十百千顺序打印出各位数字。

输入格式

一行，只包含一个十进制正整数N。

输出格式

一行，分为两部分，首先按个十百千输出各位数字；然后输出位数。数字之间以逗号分隔。

评测样例

• 样例 >>

123

• << 样例

3,2,1,3

参考代码

```
1 #include <stdio.h>
2 int main()
3 {
4     int In, Count = 0;
5     scanf("%d",&In);
6     do
7     {
8         printf("%d,", In % 10);
9         Count++;
10        In /= 10;
11    }
12    while(In > 0);
13    printf("%d",Count);
14    return 0;
15 }
```

2. 数学游戏

时间限制：1.0s; 内存限制：256.0MB

数学游戏

一天，蓬蓬和凯凯聚在一起讨论玩什么游戏。他们发现旁边的轩轩正在做一道数学题，计算多项式 $ax^4+bx^3+cx^2+dx+e$ 的值。于是，他们想到一个游戏。

游戏规则如下：

游戏有 n 个回合，每个回合开始，作为裁判的轩轩给出 a,b,c,d,e,x 的值，游戏双方蓬蓬和凯凯各自去计算多项式的值，并给出自己的答案，答对者得1分；当某一方连续回答对 k 道题后，有加分奖励，即连续回答对第 $k+1$ 题时，除了获得答对题目得到的1分，还会奖励1分，连续答对 $k+m$ 道题时，除了答对题目的1分，还会奖励 m 分。请你计算游戏结束时，蓬蓬和凯凯的得分。

输入格式

输入共 $n+1$ 行。

第1行是两个正整数 n,k 。 n 表示游戏的回合数， k 的含义如游戏规则所示。

第2行到第 $n+1$ 行，每行 8 个整数，分别表示 a,b,c,d,e,x 和蓬蓬的答案 ans_1 和凯凯的答案 ans_2

输出格式

输出只有一行，包含两个整数，分别表示蓬蓬和凯凯的得分，之间用一个空格分割。

评测样例

• 样例 >>

```
2 1
1 1 1 1 1 2 31 30
2 3 4 5 10 0 10 10
```

• << 样例

```
3 1
```

数据规模和约定

见试卷...

参考代码

```
1 #include <stdio.h>
2 long fx(int a, int b, int c, int d, int e, int x)
3 {
4     long ans = a * x * x * x * x + b * x * x * x + c * x * x + d * x + e;
5     return ans;
6 }
7 int main()
8 {
```

```

9     int n, k;
10    int a, b, c, d, e, x;
11    long ans1, ans2;
12    scanf("%d%d", &n, &k);
13    int Win1 = 0, Win2 = 0;
14    int Pt1 = 0, Crd1 = 0, Pt2 = 0, Crd2 = 0;
15    long Ans;
16    for(int i = 0; i < n; i++)
17    {
18        scanf("%d%d%d%d%d%d%ld%ld", &a, &b, &c, &d, &e, &x, &ans1, &ans2);
19        Ans = fx(a, b, c, d, e, x);
20        if(ans1 == Ans)
21        {
22            Win1++, Pt1++;
23            if(Win1 > k)
24                Crd1 += Win1 - k;
25        }
26        else Win1 = 0;
27        if(ans2 == Ans)
28        {
29            Win2++, Pt2++;
30            if(Win2 > k)
31                Crd2 += Win2 - k;
32        }
33        else Win2 = 0;
34    }
35    printf("%d %d", Pt1 + Crd1, Pt2 + Crd2);
36    return 0;
37 }

```

3. 字符串之差

时间限制：1.0s; 内存限制：256.0MB

字符串之差

编程输出两个字符串的差。

两个字符串S_1、S_2的差定义如下：S_1和S_1的按字典序比较大小。若S_1和S_2相等，则输出为0；若它们不相等，则输出其第一个不同字符的ASCII码的差值，且如果S_1>S_2，则差值为正；如果S_1<S_2，则差值为负。

输入格式

共2行，第1行是字符串S_1，第2行是字符串S_2。注意：输入数据保证每一个字符串不是另一个的前缀，且长度在100以内。

输出格式

1行仅1个整数，表示字符串之差。

评测样例

• 样例 >>

```
java
basic
```

• << 样例

```
8
```

数据规模和约定

50%的数据：输入的字符串S_1和S_2中不含分隔符；

100%的数据：每一个字符串不是另一个的前缀，且长度在100以内。

参考代码

```
1 #include <stdio.h>
2 #include <string.h>
3 int mycmp(char a[], char b[], int Lena, int Lenb)
4 {
5     int Pos = 0;
6     int Len = (Lena > Lenb) ? Lenb : Lena;
7     for(; Pos < Len; Pos++)
8     {
9         if(a[Pos] != b[Pos])
10             return Pos;
11     }
```

```
12     return -1;
13 }
14 int main()
15 {
16     char str1[101], str2[101];
17     gets(str1);
18     gets(str2);
19     int Pos = mycmp(str1, str2, strlen(str1), strlen(str2));
20     if(Pos == -1)
21         printf("0");
22     else
23         printf("%d", str1[Pos] - str2[Pos]);
24     return 0;
25 }
```

4. 购物车共同性调查

顾客购物车共同性调查

电商平台在购物狂欢节期间，从某类顾客中随机抽取 n 个顾客($n>1$)，调查他们购物车中预选保存的商品类别，以掌握顾客购物偏好的共同性，得到最受欢迎商品品类，为拓展市场销量服务。
假定商品品类是以整数进行编号，此问题为在 n 个集合中寻找交集的问题。

输入格式

共 $n+1$ 行。

第一行为一个整数 n ，表示选取的顾客数。

此后有 n 行，分别表示每个顾客所选商品种类构成，其中第一个数表示该顾客购物车的种类数。

调查抽取的顾客数 n 不超过20，每个顾客所选的商品种类不超过100。

输出格式

若不存在共同偏好，无交集，输出NO；若有，把共同商品种类编号从小到大输出，空格分隔。

评测样例

• 样例1 >>

```
3
5 1 2 3 4 5
5 2 4 6 8 10
8 1 2 3 4 5 6 8 10
```

• << 样例1

```
2 4
```

• 样例2 >>

```
2
4 1 2 3 4
3 5 6 7
```

• << 样例2

```
NO
```

参考代码

```
1 #include <stdio.h>
2 long long Num[2048] = {0};
3 void Sort(long long Num[], int n)
4 {
5     for(int i = 0; i < n - 1; i++)
6     {
7         int ptr = i;
```

```

8         for(int j = i + 1; j < n; j++)
9         {
10             if(Num[j] < Num[ptr])
11                 ptr = j;
12         }
13         if(ptr != i)
14         {
15             long long Tmp = Num[i];
16             Num[i] = Num[ptr];
17             Num[ptr] = Tmp;
18         }
19     }
20 }
21 int main()
22 {
23     int n;
24     scanf("%d", &n);
25     int Item, Total = 0;
26     for(int i = 0; i < n; i++)
27     {
28         scanf("%d", &Item);
29         for(int j = 0; j < Item; j++)
30         {
31             scanf("%lld", &Num[Total]);
32             Total++;
33         }
34     }
35     Sort(Num, Total);
36     int Flag = 0;
37     for(int i = 0; i <= Total - n; i++)
38     {
39         if(Num[i] == Num[i + n - 1])
40         {
41             printf("%lld ", Num[i]);
42             Flag = 1;
43         }
44     }

```



```
45     if(!Flag)
46         printf("NO");
47     return 0;
48 }
```

5. 大整数加减

时间限制：1.0s; 内存限制：256.0MB

大整数加减

比利经常会碰到超大整数的加减法运算，而普通的计算器上无法进行。因此他想你帮他写一个程序来计算结果。

输入格式

输入共三行。
第一行为一个符号，表示要进行的计算（“+”表示要进行加法运算，“-”表示要进行减法运算）。
第二行第三行每行为一个大整数（长度小于2000），表示要进行加减操作的两个数。注意：整数既可以为正数也可以为负数。

输出格式

加减运算后的结果。

评测样例

- 样例 1 >>

```
+
998877665544332211
11223344556677
```
- << 样例 1

```
998888888888888888
```
- 样例 2 >>

```
-
-8776655443322
-91223344556677
```
- << 样例 2

```
82446689113355
```
- 样例 3 >>

```
+
998877665544332211
112233445566778899
```
- << 样例 3

```
111111111111111110
```

数据规模和约定

运算要支持加法和减法两种运算。整数使用十进制，位数小于2000位。

参考代码

```
1 #include <stdio.h>
2 #include <string.h>
3 void ParseStr2Int(int Dest[], char Raw[], int Len)
4 {
5     for(int i = 0; i < Len; i++)
6         Dest[i] = Raw[Len - 1 - i] - '0';
7 }
8 int RawCalc(int Dest[], int A[], int B[], int LenA, int LenB, char Op)
9 {
10     if(Op == '+')
11     {
12         int Len = (LenA > LenB) ? LenA : LenB, r = 0;
13         for(int i = 0; i <= Len; i++)
14         {
15             Dest[i] = A[i] + B[i] + r;
16             r = Dest[i] / 10;
17             Dest[i] %= 10;
18         }
19         return 0;
20     }
21     else if(Op == '-')
22     {
23         if(LenA > LenB || (LenA == LenB && A[LenA - 1] > B[LenB - 1]))
24         {
25             for(int i = 0; i <= LenA; i++)
26             {
27                 while(A[i] < B[i])
28                     A[i + 1]--, A[i] += 10;
29                 Dest[i] = A[i] - B[i];
30             }
31             return 0;
32         }
33         else
34         {
35             RawCalc(Dest, B, A, LenB, LenA, '-');
36             return 1;
37         }
38     }
39 }
```

```

37     }
38 }
39 return -1;
40 }
41 int SwitchCalc(int Dest[], int A[], int B[], int LenA, int LenB, int MarkA, int
    MarkB, char Op)
42 {
43     //-(+/-)-    //+(+/-)+    //+(+/-)-    //-(+/-)+
44     if(Op == '+')
45     {
46         if(MarkA && MarkB)
47             return !(RawCalc(Dest, A, B, LenA, LenB, '+'));
48         else if(!(MarkA || MarkB))
49             return RawCalc(Dest, A, B, LenA, LenB, '+');
50         else if(!MarkA && MarkB)
51             return RawCalc(Dest, A, B, LenA, LenB, '-');
52         else if(MarkA && !MarkB)
53             return RawCalc(Dest, B, A, LenB, LenA, '-');
54     }
55     else if(Op == '-')
56     {
57         if(MarkA && MarkB)
58             return RawCalc(Dest, B, A, LenB, LenA, '-');
59         else if(!(MarkA || MarkB))
60             return RawCalc(Dest, A, B, LenA, LenB, '-');
61         else if(!MarkA && MarkB)
62             return RawCalc(Dest, A, B, LenA, LenB, '+');
63         else if(MarkA && !MarkB)
64             return !(RawCalc(Dest, A, B, LenA, LenB, '+'));
65     }
66     return -1;
67 }
68 int main(int argc, char *argv[])
69 {
70     char Op;
71     scanf("%c", &Op);
72     char RawA[2048] = {0}, RawB[2048] = {0};

```

```
73     scanf("%s%s", RawA, RawB);
74     int MarkA = 0, MarkB = 0;
75     if(RawA[0] == '-')
76         MarkA = 1;
77     if(RawB[0] == '-')
78         MarkB = 1;
79     int LenA = strlen(RawA + MarkA);
80     int LenB = strlen(RawB + MarkB);
81     int IntA[2048] = {0}, IntB[2048] = {0};
82     ParseStr2Int(IntA, RawA + MarkA, LenA);
83     ParseStr2Int(IntB, RawB + MarkB, LenB);
84     int Res[2048] = {0};
85     if(SwitchCalc(Res, IntA, IntB, LenA, LenB, MarkA, MarkB, Op))
86         printf("-");
87     int Pos;
88     for(Pos = 2047; Pos > 0; Pos--)
89         if(Res[Pos])
90             break;
91     for(;Pos >= 0; Pos--)
92         printf("%d", Res[Pos]);
93     return 0;
94 }
```