

# 第六次上机题解

## --刘志江

本次上机考察了条件判断、循环、数组、常用字符串函数及数学函数等内容。大家出错较多的地方是C题的数组越界和F题的二分法解方程。部分同学在写程序时仍然不写缩进，不便于理清程序的逻辑，容易出现细节上的问题。此外，没有仔细阅读比赛简介的同学，请及时阅读。

## A 温标转换 (839/971)

### 题意

多组输入，每组输入一个浮点数加一个字符。根据字符判断输入的浮点数是摄氏度或华氏度，若低于绝对零度，输出"Out of Range"，否则输出该温度在华氏度或摄氏度中的数值并加字符F或C。

### 解题思路

首先判断输入的是摄氏度还是华氏度，然后根据温度转换公式计算转换后的温度值，判断是否低于绝对零度；若低于绝对零度，输出"Out of Range"，否则输出转换后的温度值以及温度类型符。

### 代码

```
#include<stdio.h>

double t;
char c;

int main()
{
    while(scanf("%lf%c",&t,&c)==2)
    {
        if(c=='F')
            t=5.0/9.0*(t-32);
        else
            t=9.0/5.0*t+32;
        if(c=='F'&&t<-273.15||c=='C'&&t<-459.67)
            puts("Out of Range");
        else
            printf("%.2lf%c\n",t,'F'+'C'-c);
    }
    return 0;
}
```

# B HugeGun学姐的数列(612/820)

## 题意

输入x，计算

$$\lim_{n \rightarrow +\infty} \left( \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!} \right)$$

输出计算结果，并保留x位小数。

## 解题思路1

$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + o(x^n)$  (迈克劳林公式)，在这里将x=1代入后题目中所求的极限可以化成 $e^1 - 1$ 。使用exp()函数计算e；使用格式控制符，保留x位输出即可。

## 代码1

```
#include <stdio.h>
#include <math.h>
int main()
{
    int x;
    scanf("%d",&x);
    printf("%.*lf",x,exp(1)-1);
    return 0;
}
```

## 解题思路2

x的取值范围是

$$x(0 \leq x \leq 9)$$

容易发现

$$\frac{1}{14!} \leq 1e - 10$$

所以我们只需要计算

$$\frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{14!}$$

即可得到题目要求的答案。将printf()函数的格式串保存在字符串中，输入x后，将格式串的第2个字符置为x的ascii码值即可。

## 代码2

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
```

```
double jie[20];
char s[20]="%.1f";
int main()
{
    int i;
    double ans=0;
    jie[0]=1;
    for(i=1;i<=14;i++)jie[i]=jie[i-1]*i;
    for(i=1;i<=14;i++)ans+=1./jie[i];
    int x;
    scanf("%d",&x);
    s[2]=x+'0';
    printf(s,ans);
    return 0;
}
```

## C 离散卷积 (68/207)

### 题意

输入两个数组f、g，输出他们的离散卷积

$$y_i = \sum_{m=0}^{M-1} f_{i-m} * g_m$$

### 解题思路

双重for循环计算第三个数组中每一项的值。由卷积定义可知，y数组共有N+M-1个元素，第一重循环遍历y数组中的每个元素；第二重循环为求和符号的范围0~M-1，如果i-j>=0且i-j<n（即f[i-j]存在，没有越界），则y[i] += f[i-j] \* g[j]。计算完成后输出计算结果即可。大家出错的主要原因是没有判断f[i-j]是否存在。

有了思路后，我们考虑是否需要使用longlong数据类型。-1000 < f[i],g[i] < 1000，即f[i] \* g[i] < 1000000，m < 100，可知y[i] < 100000000，所以说本题中int类型的变量足够使用，无需定义longlong类型的变量。

### 代码

```
#include<stdio.h>

int main()
{
    int n, m, i, j, t, f[105], g[105], y[205];

    scanf("%d", &t);
    while (t--){
        scanf("%d %d", &n, &m);
        for (i=0; i<n; i++) scanf("%d", &f[i]);
        for (i=0; i<m; i++) scanf("%d", &g[i]);
        for (i=0; i<n+m-1; i++){
```

```

        y[i] = 0;
        for (j=0; j<m; j++){
            if (i-j>=0 && i-j<n){
                y[i] += f[i-j] * g[j];
            }
        }
    }
    for (i=0; i<n+m-1; i++) printf("%d ", y[i]);
    printf("\n");
}
}

```

## D 石头剪刀布 (793/892)

### 题意

输入"paper" / "rock" / "scissors", 根据输入的字符串, 输出"scissors" / "paper" / "rock".

### 解题思路

本题的一个小技巧是根据每个字符串的首个字符, 判断输入字符串的含义, 输出相应的字符串。wa了的同学问题主要在于多组数据的输入。下面介绍几种常用的字符串读取函数的返回值。

scanf()函数成功读入时会返回读取字段的个数, 返回0表示没有数据被成功地读入; 当读入错误或者函数只读到文件结束符时, 返回一个EOF常量。while(~scanf("%s",str))表示当scanf()函数的返回值不为0时。

get()正常情况下返回其读取的字符串, 执行遇到异常时返回NULL。

相比于gets(), 推荐使用更安全的fgets()函数, fgets()函数返回值和gets()函数类似。

### 代码1

```

#include <stdio.h>

char str[10];

int main(){

    while (~scanf("%s",str))
    {
        if (str[0]=='r')
            printf("paper\n");
        else if (str[0]=='p')
            printf("scissors\n");
        else if (str[0]=='s')
            printf("rock\n");
    }
}

```

```
    return 0;
}
```

## 代码2

```
#include <stdio.h>
#include<string.h>

char str[10];

int main(){

    while (gets(str)!=NULL)
    {
        if (str[0]=='r')
            printf("paper\n");
        else if (str[0]=='p')
            printf("scissors\n");
        else if (str[0]=='s')
            printf("rock\n");
    }

    return 0;
}
```

## 代码3

```
#include <stdio.h>
#include<string.h>

char str[10];

int main(){

    while (fgets(str, 10, stdin)!=NULL)
    {
        if (str[0]=='r')
            printf("paper\n");
        else if (str[0]=='p')
            printf("scissors\n");
        else if (str[0]=='s')
            printf("rock\n");
    }

    return 0;
}
```

## 题意

以字符串形式读入两个十六进制数，并比较大小。

## 解题思路

题目中说明了没有前导0，故长的字符串一定大；当两个字符串长度一样时，使用strcmp()函数比较字符串大小即可。

## 代码

```
#include <stdio.h>
#include <string.h>

char a[5005],b[5005];
int T;

int main()
{
    scanf("%d",&T);
    while (T--)
    {
        scanf("%s",a);
        scanf("%s",b);

        if (strlen(a) < strlen(b)) puts("B");
        else if (strlen(a) > strlen(b)) puts("A");
        else if (strcmp(a,b) < 0) puts("B");
        else if (strcmp(a,b) > 0) puts("A");
        else puts("EQUAL");
    }

    return 0;
}
```

# F lx解方程 (168/317)

## 题意

给出函数值y，求解方程

$$\frac{\sin(\sqrt{x}) + e^{-(x^{\frac{1}{3}})}}{\ln(\pi x)} = y$$

## 解题思路

因为题目给出了x的明确范围，故考虑使用二分法求解。令

$$f(x) = \frac{\sin(\sqrt{x}) + e^{-(x^{\frac{1}{3}})}}{\ln(\pi x)}$$

容易算出 $f(0.33) - y > 0$ ,  $f(10) - y < 0$ , 因此可以使用二分法求解此方程。因为存在浮点误差, 当  $\text{fabs}(f(\text{mid}) - y) < 1e-7$  时, 即认为找到了方程的根。下面分别给出使用循环和递归解决此题的代码。

## 代码1

```
#include <stdio.h>
#include <math.h>
double f(double x)
{
    return (sin(sqrt(x)) + exp(-pow(x, 1.0/3))) / log(acos(-1.0)*x);
}

int main()
{
    double y;
    scanf("%lf", &y);
    double l = 0.33, r = 10, mid;
    while (l < r) {
        mid = (l+r) / 2;
        double t = f(mid);
        if (fabs(f(mid) - y) < 1e-7) break;
        if (f(mid) < y) r = mid; else l = mid;
    }
    printf("%.5f\n", mid);
    return 0;
}
```

## 代码2

```
#include <stdio.h>
#include <math.h>
#define PI acos(-1)
double y;
double check(double a, double b)
{
    double mid = (a+b)/2;
    double fx = (sin(sqrt(mid)) + exp(-pow(mid, 1.0/3))) / log(PI*mid);
    if (fabs(fx - y) < 1E-7) return mid;
    else if (fx > y + 1E-7) return check(mid, b);
    else return check(a, mid);
}

int main()
{
    scanf("%lf", &y);
    printf("%.5lf", check(0.33, 10));
    return 0;
}
```

