

# C语言刷题训练营-第八讲

比特就业课@提供课程讲解和技术支持

题目来自: [牛客网](#)

## BC72-平均身高

### 题目描述

从键盘输入5个人的身高（米），求他们的平均身高（米）。

### 输入描述:

一行，连续输入5个身高（范围0.00~2.00），用空格分隔。

### 输出描述:

一行，输出平均身高，保留两位小数。

### 示例1

#### 输入

1.68 1.75 1.82 1.60 1.92

#### 输出

1.75

### 参考代码:

```
#include <stdio.h>

int main()
{
    int i = 0;
    double score = 0.0;
    double sum = 0.0;
    for(i=0; i<5; i++)
    {
        scanf("%lf", &score);
        sum += score;
    }
    printf("%.2lf\n", sum/5);
    return 0;
}
```

## BC74-HTTP状态码

### 题目描述

KiKi访问网站，得到HTTP状态码，但他不知道什么含义，BoBo老师告诉他常见HTTP状态码：200（OK，请求已成功），202（Accepted，服务器已接受请求，但尚未处理。）400（Bad Request，请求参数有误），403（Forbidden，被禁止），404（Not Found，请求失败），500（Internal Server Error，服务器内部错误），502（Bad Gateway，错误网关）。

### 输入描述:

多组输入，一行，一个整数（100~600），表示HTTP状态码。

### 输出描述:

针对每组输入的HTTP状态，输出该状态码对应的含义，具体对应如下：

200-OK  
202-Accepted  
400-Bad Request  
403-Forbidden  
404-Not Found  
500-Internal Server Error  
502-Bad Gateway

### 示例1

#### 输入

200

#### 输出

OK

### 参考代码:

```
#include <stdio.h>

int main()
{
    int s = 0;
    while(scanf("%d", &s) != EOF)
    {
        switch(s)
        {
            case 200:
                printf("OK\n");
                break;
            case 202:
                printf("Accepted\n");
                break;
            case 400:
                printf("Bad Request\n");
                break;
```

```

        case 403:
            printf("Forbidden\n");
            break;
        case 404:
            printf("Not Found\n");
            break;
        case 500:
            printf("Internal Server Error\n");
            break;
        case 502:
            printf("Bad Gateway\n");
            break;
    }
}
return 0;
}
    
```

### 答案解析:

1. 多组输入
2. 状态码比较多，且为整数，所以可以考虑用switch语句。

## BC75-数字三角形

### 题目描述

KiKi学习了循环，BoBo老师给他出了一系列打印图案的练习，该任务是打印用数字组成的数字三角形图案。

### 输入描述:

多组输入，一个整数（3~20），表示数字三角形边的长度，即数字的数量，也表示输出行数。

### 输出描述:

针对每行输入，输出用数字组成的对应长度的数字三角形，每个数字后面有一个空格。

### 示例1

#### 输入

4

#### 输出

```

1
1 2
1 2 3
1 2 3 4
    
```

### 示例2

#### 输入

**输出**

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

**参考代码：**

```
#include <stdio.h>

int main()
{
    int n = 0;
    while(scanf("%d", &n) != EOF)
    {
        int i = 0;
        for(i=1; i<=n; i++)
        {
            int j = 0;
            for(j=1; j<=i; j++)
            {
                printf("%d ", j);
            }
            printf("\n");
        }
    }
    return 0;
}
```

**答案解析：**

1. 多组输入
2. 分析清楚每行打印的内容就行了

**BC76-公务员面试****题目描述**

公务员面试现场打分。有7位考官，从键盘输入若干组成绩，每组7个分数（百分制），去掉一个最高分和一个最低分，输出每组的平均成绩。

**输入描述：**

一行，输入7个整数（0~100），代表7个成绩，用空格分隔。

**输出描述：**

一行，输出去掉最高分和最低分的平均成绩，小数点后保留2位，每行输出后换行。

## 示例1

### 输入

```
99 45 78 67 72 88 60
```

### 输出

```
73.00
```

参考答案：

```
#include <stdio.h>

int main()
{
    int i = 0;
    int sum = 0;
    int min = 100; //假设最小值是100
    int max = 0; //假设最大值是0
    int score = 0;
    for(i=0; i<7; i++)
    {
        scanf("%d ", &score); //输入
        sum += score; //求和
        if(score > max)
            max = score; //求最大值
        if(score < min)
            min = score; //求最小值
    }
    printf("%.2f\n", (sum-min-max)/5.0);
    return 0;
}
```

答案解析：

1. 多组输入
2. 一边输入，一边求和，一边求最大值最小值
3. 再算平均值

## BC77-有序序列插入一个数

### 题目描述

有一个有序数字序列，从小到大排序，将一个新输入的数插入到序列中，保证插入新数后，序列仍然是升序。

### 输入描述:

第一行输入一个整数( $0 \leq N \leq 50$ )。第二行输入N个升序排列的整数，输入用空格分隔的N个整数。第三行输入想要进行插入的一个整数。

### 输出描述:

## 示例1

### 输入

```
5
1 6 9 22 30
8
```

### 输出

```
1 6 8 9 22 30
```

### 参考代码：

```
#include <stdio.h>

int main()
{
    int n = 0;
    int arr[50] = {0};
    int m = 0; //要插入的数据
    //输入数据
    scanf("%d", &n);
    int i = 0;
    for(i=0; i<n; i++)
    {
        scanf("%d", &arr[i]);
    }
    scanf("%d", &m);
    //处理
    for(i=n; i>0 ; i--)
    {
        if(arr[i-1]> m)
        {
            arr[i] = arr[i-1];
        }
        else
        {
            arr[i] = m;
            break;
        }
    }
    //如果插入数字小于所有的数字
    if(i == 0)
    {
        arr[i] = m;
    }
    //打印
    for(i=0; i<n+1; i++)
    {
        printf("%d ", arr[i]);
    }
}
```

```
return 0;    比特就业课，400小时就业课，保姆式就业服务
}
```

### 答案解析:

1. 多组输入
2. 从往前比较并一定数字找到合适的位置，就可以插入数字了，但是如果插入的数字比所有的元素都小，那就插入到第一个位置上。

## BC78-筛选法求素数

### 题目描述

用筛选法求n以内的素数。筛选法求解过程为：将2~n之间的正整数放在数组内存储，将数组中2之后的所有能被2整除的数清0，再将3之后的所有能被3整除的数清0，以此类推，直到n为止。数组中不为0的数即为素数。

### 输入描述:

多组输入，每行输入一个正整数（不大于100）。

### 输出描述:

针对每行输入的整数n，输出两行，第一行，输出n之内（包括n）的素数，用空格分隔，

第二行，输出数组中2之后0 的个数。每行输出后换行。

### 示例1

#### 输入

20

#### 输出

2 3 5 7 11 13 17 19  
11

### 参考答案:

```
#include <stdio.h>
//方法1: 筛选法
int main()
{
    int n = 0;
    while (~scanf("%d", &n))
    {
        int arr[n+1];
        int i = 0;
        //初始化数组的内容为0-n
        for (i = 0; i<=n; i++)
        {
```

```

arr[i] = 1;
}
for (i = 2; i <= n; i++)
{
    int j = 0;
    for (j = i + 1; j <= n; j++)
    {
        if (arr[j] % i == 0)
            arr[j] = 0;
    }
}
int count = 0;
for (i = 2; i <= n; i++)
{
    if (arr[i] != 0)
        printf("%d ", arr[i]);
    else
        count++;
}
printf("\n%d\n", count);
}
return 0;
}

```

//方法2: 试除法

```

int is_prime(int n)
{
    int i = 0;
    for(i=2; i<n; i++)
    {
        if(n%i == 0)
            return 0;
    }
    return 1;
}

```

```

int main()
{
    int n = 0;
    //题目相当于打印n以内的素数
    while(scanf("%d", &n) != EOF)
    {
        int i = 0;
        int count = 0;
        for(i=2; i<=n; i++)
        {
            if(is_prime(i)==1)
            {
                printf("%d ", i);
            }
            else
            {
                count++;
            }
        }
        printf("\n%d\n", count);
    }
    return 0;
}

```



## BC79-图像相似度

### 题目描述

给出两幅相同大小的黑白图像（用0-1矩阵）表示，求它们的相似度。若两幅图像在相同位置上的像素点颜色相同，则称它们在该位置具有相同的像素点。两幅图像的相似度定义为相同像素点数占总像素点数的百分比。

### 输入描述:

第一行包含两个整数m和n，表示图像的行数和列数，用单个空格隔开。 $1 \leq m \leq 100$ ， $1 \leq n \leq 100$ 。之后m行，每行n个整数0或1，表示第一幅黑白图像上各像素点的颜色，相邻两个数用单个空格隔开。之后m行，每行n个整数0或1，表示第二幅黑白图像上各像素点的颜色，相邻两个数用单个空格隔开。

### 输出描述:

一个实数，表示相似度(以百分比的形式给出)，精确到小数点后两位。

### 示例1

#### 输入

```
3 3
1 0 1
0 0 1
1 1 0
1 1 0
0 0 1
0 0 1
```

#### 输出

```
44.44
```

### 参考代码:

```
#include <stdio.h>
int main()
{
    int m = 0;
    int n = 0;
    int a[100][100] = {0}, b[100][100] = {0};
    scanf("%d%d", &m, &n);
    int i = 0;
    int j = 0;
    int count = 0.0;
    //输入数据
    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++)
```

```

    {
        scanf("%d", &a[i][j]);
    }
}
for(i=0; i<m; i++)
{
    for(j=0; j<n; j++)
    {
        scanf("%d", &b[i][j]);
    }
}
//求相同元素的个数
for(i=0; i<m; i++)
{
    for(j=0; j<n; j++)
    {
        if(a[i][j] == b[i][j])
        {
            count++;
        }
    }
}
//求相速度注意乘以100
printf("%.2f\n", 100.0*count/(m*n));
return 0;
}

```

### 答案解析:

1. 输入二维数组
2. 求两个二维数组的相同元素个数
3. 求百分比要注意\*100.0，小时点后保留2位

## BC80-验证登录

### 题目描述

有个软件系统登录的用户名和密码为（用户名：admin，密码：admin），用户输入用户名和密码，判断是否登录成功。

### 输入描述:

多组测试数据，每行有两个用空格分开的字符串，第一个为用户名，第二个为密码。

### 输出描述:

针对每组输入测试数据，输出为一行，一个字符串（“Login Success!”或“Login Fail!”）。

### 示例1

#### 输入

admin admin

输出

Login Success!

## 示例2

输入

admin abc

输出

Login Fail!

## 参考代码：

```
#include <stdio.h>

int main()
{
    char name[20] = {0};
    char password[20] = {0};
    while(scanf("%s %s", name, password) != EOF)
    {
        if(strcmp(name, "admin")==0 && strcmp(password, "admin")==0)
        {
            printf("Login Success!\n");
        }
        else
        {
            printf("Login Fail!\n");
        }
    }
    return 0;
}
```

## 代码解析：

1. 多组输入
2. 两个字符串比较大小不能使用==，得使用strcmp函数，比较的是两个字符串中对应字符的ASCII码值。

## BC85-包含数字9的数

### 题目描述

今年是2019年，KiKi想知道1~2019中有多少个包含数字9的数。包含数字的数是指有某一位是“9”的数，例如“2019”、“199”等。

### 输入描述：

无

**输出描述:**

一行，一个整数，表示1~2019中共有多少个数包含数字9。

**参考代码:**

```
#include <stdio.h>

int main()
{
    int i = 0;
    int count = 0;
    for(i=1; i<=2019; i++)
    {
        //判断i是否包含9
        int t = i;
        while(t)
        {
            if(t%10 == 9)
            {
                count++;
                break;
            }
            t = t/10;
        }
    }
    printf("%d\n", count);

    return 0;
}
```

**答案解析:**

1. 要判断一个数字是否包含9就看一下这个数字的某一位是否为9即可，只要某一位为9就停下里来。

## BC86-奇偶统计

**题目描述**

任意输入一个正整数N，统计1~N之间奇数的个数和偶数的个数，并输出。

**输入描述:**

一行，一个正整数N。（ $1 \leq N \leq 100,000$ ）

**输出描述:**

一行，1~N之间奇数的个数和偶数的个数，用空格分开。

**示例1****输入**

## 输出

3 2

## 参考答案：

```
//代码1
#include <stdio.h>
int main()
{
    int n = 0;
    scanf("%d", &n);
    int i = 0;
    int odd = 0; //奇数的个数
    int even = 0;
    for(i=1; i<=n; i++)
    {
        if(i%2==1)
            odd++;
    }
    printf("%d %d\n", odd, n-odd);
    return 0;
}

//代码2
#include <stdio.h>
int main()
{
    int n = 0;
    scanf("%d", &n);
    int i = 0;
    int odd = 0; //奇数的个数
    if(n%2==0)
        odd = n/2; //如果n是偶数，则奇数和偶数一样多
    else
        odd = n/2+1; //如果n是奇数，则奇数比偶数多1
    printf("%d %d\n", odd, n-odd);
    return 0;
}
```

## 答案解析：

1. 代码1就是统计技术的个数，算出偶数的个数。
2. 代码2是根据数字的个数推算。

---

比特IT读书会@提供课程讲解和技术支持

联系鹏哥：15596668862（同微信）

[完整的C语言教程链接](#)