

C语言刷题训练营-第六讲

比特就业课@提供课程讲解和技术支持

题目来自: [生客网](#)

BC52-衡量人体胖瘦程度

题目描述

在计算BMI (BodyMassIndex，身体质量指数) 的案例基础上，判断人体胖瘦程度。BMI中国标准如下表所示。

表 BMI 中国标准

BMI 范围	分类
$BMI < 18.5$	偏瘦 (Underweight)
$BMI \geq 18.5$ 且 $BMI \leq 23.9$	正常 (Normal)
$BMI > 23.9$ 且 $BMI \leq 27.9$	过重 (Overweight)
$BMI > 27.9$	肥胖 (Obese)

输入描述:

多组输入，每一行包括两个整数，用空格隔开，分别为体重（公斤）和身高（厘米）。

输出描述:

针对每行输入，输出为一行，人体胖瘦程度，即分类。

示例1

输入

```
80 170
60 170
90 160
50 185
```

输出

```
Overweight
Normal
Obese
Underweight
```

参考代码:

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

```
int main()
{
    float weight = 0.0;
    float hight = 0.0;
    while(scanf("%f %f", &weight, &hight) != EOF)
    {
        float bmi = weight/(hight*hight/100/100);
        if(bmi < 18.5)
        {
            printf("Underweight\n");
        }
        else if(bmi>=18.5 && bmi <= 23.9)
        {
            printf("Normal\n");
        }
        else if(bmi > 23.9 && bmi <= 27.9)
        {
            printf("Overweight\n");
        }
        else{
            printf("Obese\n");
        }
    }
    return 0;
}
```

答案解析:

1. 多组输入
2. 浮点数除法要保证除号的两端至少有一个浮点数

BC53-计算一元二次方程

题目描述

从键盘输入a, b, c的值，编程计算并输出一元二次方程 $ax^2 + bx + c = 0$ 的根，当 $a = 0$ 时，输出“Not quadratic equation”，当 $a \neq 0$ 时，根据 $\Delta = b^2 - 4ac$ 的三种情况计算并输出方程的根。

输入描述:

多组输入，一行，包含三个浮点数a, b, c，以一个空格分隔，表示一元二次方程 $ax^2 + bx + c = 0$ 的系数。

输出描述:

针对每组输入，输出一行，输出一元二次方程 $ax^2 + bx + c = 0$ 的根的情况。

如果 $a = 0$ ，输出“Not quadratic equation”；

如果 $a \neq 0$ ，分三种情况：

$\Delta = 0$ ，则两个实根相等，输出形式为：x1=x2=...

$\Delta > 0$ ，则两个实根不等，输出形式为：x1=...;x2=...，其中 $x1 \leq x2$ 。

$\Delta < 0$ ，则有两个虚根，则输出：x1=实部-虚部i;x2=实部+虚部i，即x1的虚部系数小于等于x2的虚部系数，实部为0时不可省略。实部= $-b / (2*a)$ ，虚部= $\sqrt{-\Delta} / (2*a)$ 所有实数部分要求精确到小数点后2位，数字、符号之间没有空格。

示例1

输入

2.0 7.0 1.0

输出

$x_1 = -3.35; x_2 = -0.15$

示例2

输入

0.0 3.0 3.0

输出

Not quadratic equation

示例3

输入

1 2 1

输出

$x_1 = x_2 = -1.00$

示例4

输入

2 2 5

输出

$x_1 = -0.50 - 1.50i; x_2 = -0.50 + 1.50i$

示例5

输入

1 0 1

输出

$x_1 = 0.00 - 1.00i; x_2 = 0.00 + 1.00i$

参考代码：

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

int main()
{
    float a = 0.0;
    float b = 0.0;
    float c = 0.0;
    while(scanf("%f %f %f", &a, &b, &c) != EOF)
    {
        if(a != 0)
        {
            float disc = b*b-4*a*c;
            if(disc>0.0)
            {
                printf("x1=%.2f;x2=%.2f\n",
                    (-b-sqrt(disc))/(2*a),
                    (-b+sqrt(disc))/(2*a)
                );
            }
            else if(disc < 0.0)
            {
                printf("x1=%.2f-%.2fi;x2=%.2f+%.2fi\n",
                    (-b)/(2*a), sqrt(-disc)/(2*a),
                    (-b)/(2*a), sqrt(-disc)/(2*a));
            }
            else
            {
                printf("x1=x2=%.2f\n", (-b)/(2*a));
            }
        }
        else
        {
            printf("Not quadratic equation\n");
        }
    }
    return 0;
}
```

答案解析:

1. 多组输入
2. 根据不同的限制条件给出相应的计算结果
3. 小点数后位数有限制

BC54-获得月份天数

题目描述

KiKi想获得某年某月有多少天，请帮他编程实现。输入年份和月份，计算这一年这个月有多少天。

输入描述:

多组输入，一行有两个整数，分别表示年份和月份，用空格分隔。

输出描述:

针对每组输入，输出为一行，一个整数，表示这一年这个月有多少天。

示例1

输入

2008 2

输出

29

参考代码:

```
#include <stdio.h>
int main()
{
    int y = 0;
    int m = 0;
    int days[12] = {31,28,31,30,31,30,31,31,30,31,30,31};
    while(scanf("%d%d", &y, &m) != EOF)
    {
        int day = days[m-1];
        if((y%4==0 && y%100!=0) || (y%400==0))
        {
            if(m == 2)
                day += 1;
        }
        printf("%d\n", day);
    }
    return 0;
}
```

代码解析:

1. 多组输入
2. 判断闰年，如果是闰年，2月多一天，其他都正常。

BC55-简单计算器

题目描述

KiKi实现一个简单计算器，实现两个数的“加减乘除”运算，用户从键盘输入算式“操作数1运算符操作数2”，计算并输出表达式的值，如果输入的运算符不包括在（+、-、*、/）范围内，输出“Invalid operation!”。当运算符为除法运算，即“/”时。如果操作数2等于0.0，则输出“Wrong!Division by zero!”

输入描述:

多组输入，一行，操作数1运算符操作数2（其中运算符包括四种：+、-、*、/）。

输出描述:

针对每组输入，输出为一行。

如果操作数和运算符号均合法，则输出一个表达式，操作数1运算符操作数2=运算结果，各数小数点后均保留4位，数和符号之间没有空格。

如果输入的运算符号不包括在（+、-、*、/）范围内，输出“Invalid operation! ”。当运算符为除法运算，即“/”时。

如果操作数2等于0.0，则输出“Wrong!Division by zero!”。

示例1

输入

```
1.0+3.0
1.0;4.0
44.0/0.0
```

输出

```
1.0000+3.0000=4.0000
Invalid operation!
Wrong!Division by zero!
```

参考代码:

```
#include <stdio.h>
//精度尽量保持高
//这个题使用float精度不满足要求
int main()
{
    double n1 = 0.0;
    double n2 = 0.0;
    char op = 0;
    while(scanf("%lf%c%lf", &n1, &op, &n2) != EOF)
    {
        switch(op)
        {
            case '+':
                printf("%.4lf+%.4lf=%.4lf", n1, n2, n1+n2);
                break;
            case '-':
                printf("%.4lf-%.4lf=%.4lf", n1, n2, n1-n2);
                break;
            case '*':
                printf("%.4lf*%.4lf=%.4lf", n1, n2, n1*n2);
                break;
            case '/':
                if(n2 == 0.0)
                    printf("Wrong!Division by zero!\n");
                else
                    printf("%.4lf/%.4lf=%.4lf", n1, n2, n1/n2);
                break;
        }
    }
}
```

```

        default:
            printf("Invalid operation!\n");
            break;
    }
}
return 0;
}
    
```

代码解析：

1. 多组输入
2. 按照格式接收，按照要求编写代码就行
3. switch语句

BC56-线段图案

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“*”组成的线段图案。

输入描述:

多组输入，一个整数（1~100），表示线段长度，即“*”的数量。

输出描述:

针对每行输入，输出占一行，用“*”组成的对应长度的线段。

示例1

输入

```

10
2
    
```

输出

```

*****
**
    
```

参考代码：

```

#include <stdio.h>

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

```
    }
    printf("\n");
}
return 0;
}
```

答案解析：

1. 多组输入
2. 每次打印一个*，按照输入的个数打印后换行就行

BC57-正方形图案

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“*”组成的正方形图案。

输入描述:

多组输入，一个整数（1~20），表示正方形的长度，也表示输出行数。

输出描述:

针对每行输入，输出用“*”组成的对应边长的正方形，每个“*”后面有一个空格。

示例1

输入

4

输出

```
* * * *
* * * *
* * * *
* * * *
```

示例2

输入

5

输出

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```


参考代码：

```
#include <stdio.h>

int main()
{
    int n = 0;
    while(scanf("%d", &n) != EOF)
    {
        int i = 0;
        //确定行数
        for(i=0; i<n; i++)
        {
            int j = 0;
            //每一行的打印，打印n组*+空格
            for(j=0; j<n; j++)
            {
                printf("* ");
            }
            printf("\n");
        }
    }
    return 0;
}
```

答案解析：

1. 多组输入
2. 每组是由n组*+空格组成

BC58-直角三角形图案

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“*”组成的直角三角形图案。

输入描述:

多组输入，一个整数（2~20），表示直角三角形直角边的长度，即“*”的数量，也表示输出行数。

输出描述:

针对每行输入，输出用“*”组成的对应长度的直角三角形，每个“*”后面有一个空格。

示例1

输入

4

输出

```
*  
* *  
* * *  
* * * *
```

示例2

输入

5

输出

```
*  
* *  
* * *  
* * * *  
* * * * *
```

参考代码：

```
#include<stdio.h>  
int main()  
{  
    int n = 0;  
    while(scanf("%d", &n) != EOF)  
    {  
        int i = 0;  
        //控制行数  
        for(i=1; i<=n; i++)  
        {  
            //每一行  
            int j = 0;  
            for(j=1; j<=i; j++)  
            {  
                printf("* ");  
            }  
            printf("\n");  
        }  
    }  
    return 0;  
}
```

答案解析：

1. 多组输入
2. 每一行的输出要有计算清楚

[BC59-翻转直角三角形图案](#)

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“*”组成的翻转直角三角形图案。

输入描述:

多组输入，一个整数（2~20），表示翻转直角三角形直角边的长度，即“*”的数量，也表示输出行数。

输出描述:

针对每行输入，输出用“*”组成的对应长度的翻转直角三角形，每个“*”后面有一个空格。

示例1

输入

5

输出

```
* * * * *
* * * *
* * *
* *
*
```

示例2

输入

6

输出

```
* * * * *
* * * *
* * *
* *
*
*
```

参考代码:

```
#include <stdio.h>

int main()
{
    int n = 0;
    while(scanf("%d", &n) != EOF)
    {
        int i = 0;
        for(i=n; i>=1; i--)
        {
            int j = 0;
```

比特就业课，400小时就业课，保姆式就业服务

```
for(j=1;j<=i;j++)
{
    //这里注意空格
    printf("* ");
}
printf("\n");
}
}
return 0;
}
```

答案解析：

1. 多组输入
2. 每两个*中间有空格，要注意
3. 控制好一行输出的内容

BC60-带空格的直角三角形

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“*”组成的带空格直角三角形图案。

输入描述:

多组输入，一个整数（2~20），表示直角三角形直角边的长度，即“*”的数量，也表示输出行数。

输出描述:

针对每行输入，输出用“*”组成的对应长度的直角三角形，每个“*”后面有一个空格。

示例1

输入

5

输出

```

    *
  * *
 * * *
* * * *
* * * * *
```

示例2

输入

4

输出

比特就业课，400小时就业课，保姆式就业服务

```

*
* *
* * *
* * * *
    
```

参考代码：

```

//代码1-常规写法
#include <stdio.h>
int main()
{
    int n = 0;
    while(scanf("%d", &n) != EOF)
    {
        int i = 0;
        //行数控制
        for(i=0; i<n; i++)
        {
            //空格
            int j = 0;
            for(j=0; j<n-1-i; j++)
            {
                printf(" ");
            }
            // *
            for(j=0; j<=i; j++)
            {
                printf("* ");
            }
            printf("\n");
        }

    }
    return 0;
}

//代码2
#include <stdio.h>
int main()
{
    int n = 0;
    while(scanf("%d", &n) != EOF)
    {
        int i = 0;
        int j = 0;
        //行数
        for(i=0; i<n; i++)
        {
            //一行
            for(j=0; j<n; j++)
            {
                //行和列的和
                //这里可以把行数和列数标出来就能看明白
                if(i+j<n-1)
                {
                    printf(" ");
                }
            }
        }
    }
}
    
```

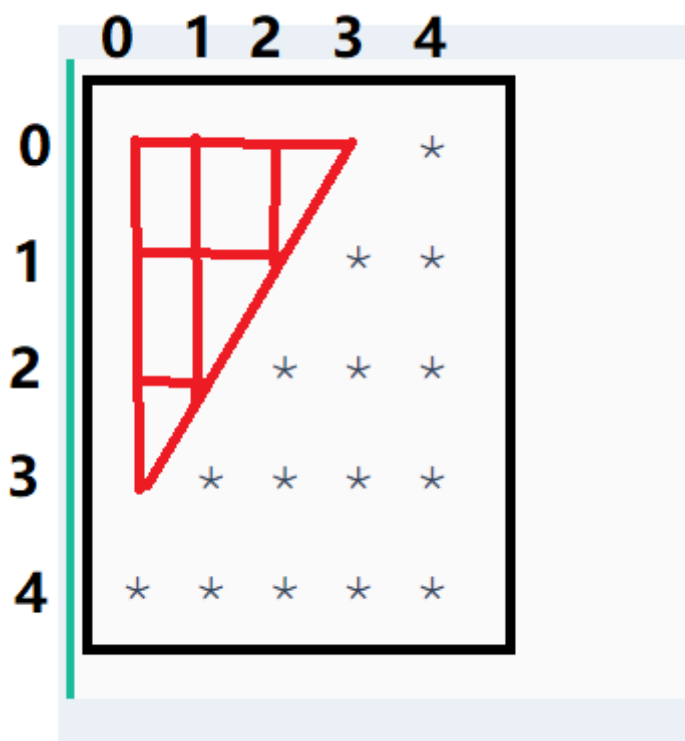
```

    }
    else
    {
        printf("* ");
    }
}
printf("\n");
}
return 0;
}

```

答案解析:

1. 多组输入
2. 方法2中 $i+j < n-1$ 理解的配图



在红色三角范围内的每个坐标上，横纵坐标之和小于 $n-1$ 的。

BC61-金字塔图案

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“*”组成的金字塔图案。

输入描述:

多组输入，一个整数（2~20），表示金字塔边的长度，即“*”的数量，也表示输出行数。

输出描述:

示例1

输入

4

输出

```

    *
  * *
* * *
* * * *
```

示例2

输入

5

输出

```

    *
  * *
* * *
* * * *
* * * * *
```

参考代码：

```

#include <stdio.h>
int main()
{
    int n = 0;
    while(scanf("%d", &n) != EOF)
    {
        int i = 0;
        for(i=0; i<n; i++)
        {
            //一行
            int j = 0;
            //空格
            for(j=0; j<n-1-i; j++)
            {
                printf(" ");
            }
            // *
            for(j=0; j<=i; j++)
            {
                printf("* ");
            }
            printf("\n");
        }
    }
}
```

```
}  
    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. 每一行的前边应该是空格，后边是*+空格，控制好个数就行

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](#)

比特就业课，400小时就业课，保姆式就业服务