



程序设计基础（C）——程序控制

郑州大学软件学院



Lecturer: 宋轩

Office : 行政楼-306

Email : songxuan@zzu.edu.cn

典型例题

数学计算

图形绘制

实际问题

算法一：利用for循环语句实现,在循环体外为sum赋初值0。

```
#include <stdio.h>
main()
{
    int i, sum = 0;
    for (i = 1; i <= 101; i = i+2)
    {
        sum = sum + i;
    }

    printf("sum=%d\n", sum);
}
```

例 1

算法二：利用while循环语句实现，在循环体外为i和sum赋初值。

```
#include <stdio.h>
main()
{
    int i = 1, sum = 0;
    while (i <= 101)
    {
        sum = sum + i;
        i = i + 2;
    }
    printf("sum=%d\n", sum);
}
```

编程计算

$1*2*3+3*4*5+.....+99*100*101$ 的值。

算法分析：用累加算法，通项公式为：

$term=i*(i+1)*(i+2); i=1,3,...,99。$

或： $term=(i-1)*i*(i+1); i=2,4,...,100。$
步长为2。

```
#include <stdio.h>
main()
{
    int i ;
    long term, sum = 0;
    for (i = 1; i <= 99; i = i + 2)
    {
        term = i * (i + 1) * (i + 2);
        sum = sum + term;
    }
    printf("sum=%ld",sum);
}
```

例 3

$$\frac{\pi}{2} = \frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \times \dots$$

前100项之积计算 π 。

累乘项为: $\text{term} = n * n / ((n-1) * (n+1))$; $n=2, 4, \dots, 100$, 步长为2。

```
#include <stdio.h>
main()
{
    float term, result = 1; /*累乘项初值应为1*/
    int n;
    for (n = 2; n <= 100; n = n + 2)
    {
        term = (float)( n * n) / (( n - 1) * ( n + 1));
        result = result * term;
    }
    printf("result = %f\n", 2*result);
}
```

例 3

采用累乘方法，累乘项为：

$\text{term} = 2 * n * 2 * n / ((2 * n - 1) * (2 * n + 1))$; $n = 1, 2, \dots, 50$ ，步长为1

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

```
main()
```

```
{
```

```
    float result = 1, term;
```

```
    int n;
```

```
    for (n = 1; n <= 50; n++)
```

```
    {
```

```
        term = (float)(2*n*2*n)/((2*n-1)*(2*n+1));
```

```
        result = result * term;
```

```
    }
```

```
    printf("result = %f\n", 2*result);
```

```
}
```

例 4

计算 $1-1/2+1/3-1/4+.....+1/99-1/100+.....$ ，直到最后一项的绝对值小于 10^{-4} 为止。

算法分析：采用累加算法，累加项通式为： $term=sign/n$ ；分子 $sign=-sign$ ；初值为1，分母 $n=n+1$ ；初值为1。

```
#include <stdio.h>
#include <math.h>
main()
{
    int n = 1;
    float term = 1.0, sign = 1, sum = 0;
    while (fabs(term) >= 1e-4)           /*判断末项大小*/
    {
        term = sign / n;                /*求出累加项*/
        sum = sum + term;               /*累加*/
        sign = -sign;                  /*改变项的符号*/
        n++;                           /*分母加1*/
    }
    printf("sum = %f\n", sum);
}
```


编程计算 $1! + 2! + 3! + 4! + \dots + 10!$ 的值。

算法一：用累加算法，累加项为：

term=term*i; i=1,2,...10。term初值为1，
使用单重循环完成。

```
#include <stdio.h>
main()
{
    long term = 1,sum = 0;
    int i;
    for (i = 1; i <= 10; i++)
    {
        term = term * i;
        sum = sum + term;
    }

    printf("1!+2!+...+10! = %ld \n", sum);
}
```

算法二：用内层循环求阶乘，外层循环控制累加的项数。

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

```
main()
```

```
{
```

```
    long term ,sum = 0;
```

```
    int i, j;
```

```
    for (i = 1; i <= 10; i++)
```

```
    {
```

```
        term = 1;
```

```
        for (j = 1; j <= i; j++)
```

```
        {
```

```
            term = term * j;
```

```
        }
```

```
        sum = sum + term;
```

```
    }
```

```
    printf("1!+2!+...+10! = %ld \n", sum);
```

```
}
```

编程计算 $a+aa+aaa+....+aa..a$ (n 个 a)的值, n 和 a 的值由键盘输入
算法思想: 用累加算法, 累加项为:

$term=term*10+a; i=1,2,...n$ 。term初值为0。

```
#include <stdio.h>
main()
{
    long term = 0,sum = 0;
    int a , i, n;
    printf("Input a,n: ");
    scanf("%d,%d", &a, &n);           /*输入a,n的值*/

    for (i = 1; i <= n; i++)
    {
        term = term * 10 + a;          /*求出累加项*/
        sum = sum + term;              /*进行累加*/
    }
    printf("sum=%ld\n",sum);
}
```

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

的近似值，当最后一项的绝对值小于 10^{-5} 时认为达到精度要求，要求统计总共累加了多少项。

算法思想：采用累加算法： **$e=e+term$** ；寻找累加项构成规律：利用前项计算后项比寻找统一的累加项表示形式要简单一些，由

$$\frac{1}{2!} = \frac{1}{1!} \div 2, \quad \frac{1}{3!} = \frac{1}{2!} \div 3, \quad \dots$$

可以发现前后项之间的关系是： **$term_n = term_{n-1} \div n$** ，写成C语句便是： **$term = term/n$** ； **$term$ 初值为1.0**， **n 初值也为1**， **n 按 $n=n+1$ 变化**。统计累加项数只要设置一个计数器变量即可，这里，计数器变量取名为 **$count$** ，初值为0，在循环体中每累加一项就加一次1。

例 7

```
#include <stdio.h>
#include <math.h>
main()
{
    int n = 1, count = 1;
    float sum = 1.0, term = 1.0;
    while (fabs(term) >= 1e-5)
    {
        term = term / n;
        sum = sum + term;
        n++;
        count++;
    }
    printf("sum = %f, count = %d\n", sum, count);
}
```

利用泰勒级数计算 $\sin(x)$

$$\sin(x) \approx x - x^3/3! + x^5/5! - x^7/7! + x^9/9! - \dots$$

要求最后一项的绝对值小于 10^{-5} ，并统计出此时累加了多少项。

算法思想： x 由键盘输入，采用累加算法， $\text{sum} = \text{sum} + \text{term}$ ， sum 初值为 x ，利用前项求后项的方法计算累加项：

$$\text{term} = -\text{term} * x * x / ((n+1) * (n+2));$$

term 初值为 x ， n 初值为1， $n = n + 2$ 。

```
#include <math.h>
main()
{
    int n = 1, count = 0;
    float x;
    double sum, term; /*因为位数多, 所以定义为双精度 */

    scanf("%f", &x);
    sum = x;
    term = x; /*赋初值*/

    do
    {
        /*计算相应项, 并改应符号*/
        term = -term * x * x / ((n + 1) * (n + 2));
        sum = sum + term; /*累加 */
        n = n + 2;
        count++;
    } while (fabs(term) >= 1e-5);

    printf("sin(x) = %f, count = %d\n", sum, count);
}
```

读入一个年份和月份，打印出该月有多少天
(考虑闰年)，用switch语句编程。

[提示：闰年的2月有29天，平年的2月有28天。]


```
#include <stdio.h>
main()
{
```

4.9

```
    int year, month;
    scanf("%d", &year, &month); /*输入相应的年和月*/
    switch (month)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            printf("31 days\n");
            break;
        case 2:
            if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
            {
                printf("29 days\n"); /*闰年的2月有29天*/
            }
            else
            {
                printf("28 days\n"); /*平年的2月有28天*/
            }
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            printf("30 days\n");
            break;
        default:
            printf("Input error!\n");
    }
}
```

程序运行结果如下：

第一次运行：

1988,5↵

31 days

第二次运行：

1988,2↵

29 days

第三次运行：

1989,2↵

28 days

打印所有的“水仙花数”，所谓“水仙花数”是指一个三位数，其各位数字立方和等于该数本身。例如153是“水仙花数”，因为 $153 = 1 + 125 + 27$ 。

[提示：首先确定水仙花数 n 可能存在的范围，因为 n 是一个三位数，所以范围确定为 n 从100变化到999，分离出 n 的个位 i 、十位 j 、百位 k 后，只要判断 n 是否等于 $i*i*i + j*j*j + k*k*k$ 即可知 n 是否是水仙花数。分离各位数字的方法可参考例4.2。]

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

4.13

```
main()
```

```
{
```

```
    int i, j, k, n;
```

```
    printf("result is:");
```

```
    for (n = 100; n < 1000; n++)
```

```
    {
```

```
        i = n / 100;
```

```
        /*分出百位*/
```

```
        j = (n - i * 100) / 10;
```

```
        /*分出十位*/
```

```
        k = n % 10;
```

```
        /*分出个位*/
```

```
        if (i*100 + j*10 + k == i*i*i + j*j*j + k*k*k)
```

```
        {
```

```
            printf("%d\t", n);
```

```
            /*输出结果*/
```

```
        }
```

```
    }
```

```
    printf("\n");
```

```
}
```

程序运行结果为:

result is:153 370 371 407

在屏幕上显示一张如下所示的时间表：

*******Time*******

1 morning

2 afternoon

3 night

Please enter your choice:

操作人员根据提示进行选择，程序根据输入的时间序号显示相应的问候信息，选择1时显示"Good morning"，选择2时显示"Good afternoon"，选择3时显示"Good night"，对于其它选择显示"Selection error!"，用switch语句编程实现。

```
#include <stdio.h>
main()
{
```

```
    char c;
```

```
    printf("*****Time*****\n");
```

```
    printf("1 morning \n");
```

```
    printf("2 afternoon \n");
```

```
    printf("3 night \n");
```

```
    printf("please enter your choice"); /*建立相应的菜单 */
```

```
    c = getchar(); /*输入选项*/
```

```
    switch (c) /*通过switch选择 */
```

```
    {
```

```
        case '1':
```

```
            printf("Good morning \n");
```

```
            break;
```

```
        case '2':
```

```
            printf("Good afternoon \n");
```

```
            break;
```

```
        case '3':
```

```
            printf("Good night\n");
```

```
            break;
```

```
        default:
```

```
            printf("Selection error!\n");
```

```
    }
```

```
}
```

4.8

```
*****Time*****
```

```
1 morning
```

```
2 afternoon
```

```
3 night
```

```
Please enter your choice:1↵
```

```
Good moning
```

1.编程判断输入整数的正负性和奇偶性。

```
#include <stdio.h>
main()
```

4.5

```
{
    int m;
    scanf("%d", &m);    /*输入一个整数*/
    if (m >= 0)         /*是否为正数*/
    {
        if (m%2 == 0)   /*是正数，且能被2整除，是正偶数*/
        {
            printf("%d is a positive even\n", m);
        }
        else           /*不能被2整除，是正奇数*/
        {
            printf("%d is a positive odd\n", m);
        }
    }
    else
    {
        if (m % 2 == 0)
        {
            printf("%d is a negative even\n", m);/*是负偶数*/
        }
        else
        {
            printf("%d is a negative odd\n", m);/*是负奇数*/
        }
    }
}
```

第一次运行:

6↵

6 is a positive even

第二次运行:

-7↵

-7 is a negative odd

编程计算分段函数：

$$y = \begin{cases} e^{-x} & x > 0 \\ 1 & x = 0 \\ -e^x & x < 0 \end{cases} \quad , \text{ 输入 } x, \text{ 打印出 } y \text{ 值。}$$


```
#include <stdio.h>
#include <math.h>
main()
{
    int x;
    double y;

    scanf("%d", &x); /* 输入一个整数*/
    if (x > 0)
    {
        y = exp(-x); /*如果大于0, 计算y=exp(-x)的值 */
    }
    else if (x == 0)
    {
        y = 1;      /*x=0, 则y=1*/
    }
    else
    {
        y = -exp(x); /*x<0, 则y=-exp(x)*/
    }
    printf("y=%f\n", y);
}
```

第一次运行:
4↵
y=0.018316
第二次运行:
0↵
y=1.000000
第三次运行:
-4↵
y=-0.018316

图形打印



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

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

```
main()
```

```
{
```

```
    int i, k;
```

```
    for (i = 1; i <= 4; i++)          /*i控制行数*/
```

```
    {
```

```
        for (k = 1 ;k <= 6; k++)      /*每行输出6个*字符*/
```

```
        {
```

```
            printf("*");
```

```
        }
```

```
        printf("\n");    /*将光标移到下一行起始位置处*/
```

```
    }
```

```
}
```

例 15

101010111

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

```
main()
```

```
{
```

```
    int i, j, k;
```

```
    for (i = 1; i <= 4; i++)          /*i控制行数*/
```

```
    {
```

```
        for (j = 1; j <= 4 - i; j++) /* 随行数的增加，输出递减数目的空格*/
```

```
        {
```

```
            printf(" ");
```

```
        }
```

```
        for (k = 1 ;k <= 6; k++)      /*每行输出6个*字符*/
```

```
        {
```

```
            printf("*");
```

```
        }
```

```
        printf("\n");                /*将光标移到下一行起始位置处*/
```

```
    }
```

```
}
```

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

例 16

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

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

```
main()
```

```
{
```

```
    int  i, j, k;
```

```
    char space = ' ';
```

```
    for (i=1;i<=4;i++)
```

```
    {
```

```
        for (j=1; j<=i-1; j++)
```

```
        {
```

```
            printf("%c",space);
```

```
        }
```

```
        for (k=1; k<=6; k++)
```

```
        {
```

```
            printf("*");
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
}
```

4.26

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

```
main()
```

```
{
```

```
    int i, j, k;
```

```
    for (i = 1; i <= 4; i++) /*控制行数*/
```

```
    {
```

```
        for (k = 1; k <= (2 * i - 1); k++) /*控制每行输出的*号个数*/
```

```
        {
```

```
            printf("*");
```

```
        }
```

```
        printf("\n"); /*输出一行后换行*/
```

```
    }
```

```
}
```

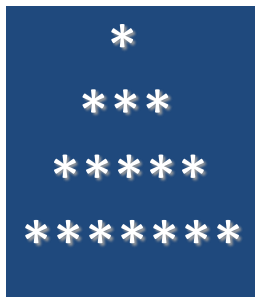
```
*
***
*****
*****
```

例 18

1010101110 10101010 10 10 101010101

111010101110 10101010

4.26

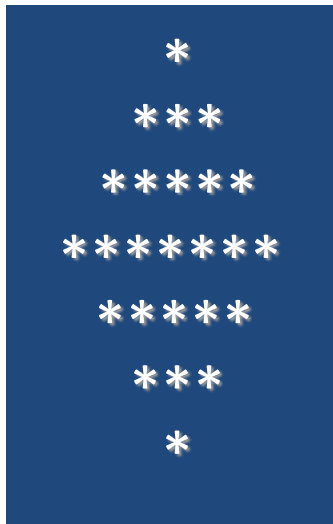


```
#include <stdio.h>
main()
{
    int i , j, k ;

    for (i = 1; i <= 4; i++)           /* 控制行数*/
    {
        for (j = 1; j <= 4 - i; j++)
        {
            printf(" ");
        }
        for (k = 1; k <= (2 * i - 1); k++)
        {
            printf("*");
        }
        printf("\n");
    }
}
```

例 19

4.26



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

```
main()
```

```
{
```

```
    int i , j, k ;
```

```
    for (i = 1; i <= 4; i++)
```

```
    {
```

```
        for (j = 1; j <= 4 - i; j++)
```

```
        {
```

```
            printf(" ");
```

```
        }
```

```
        for (k = 1; k <= (2 * i - 1); k++)
```

```
        {
```

```
            printf("*");
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
    for (i = 1; i <= 3; i++)
```

```
    {
```

```
        for (j = 1; j <= i; j++)
```

```
        {
```

```
            printf(" ");
```

```
        }
```

```
        for (k = 1; k <= 5 - 2 * (i - 1); k++)
```

```
        {
```

```
            printf("*");
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
}
```


例 20

按如下格式打印100以内整数的平方根表。

	0	1	2	3	4	5	6	7	8	9
0	0.000	1.000	1.414	1.732	2.000	2.236	2.449	2.646	2.828	3.000
1	3.162	3.317	3.464	3.606	3.742	3.873	4.000	4.123	4.243	4.359
2	4.472	4.583	4.690	4.796	4.899	5.000	5.099	5.196	5.292	5.385
3	5.477	5.568	5.657	5.745	5.831	5.916	6.000	6.083	6.164	6.245
4	6.325	6.403	6.481	6.557	6.633	6.708	6.782	6.856	6.928	7.000
5	7.071	7.141	7.211	7.280	7.348	7.416	7.483	7.550	7.616	7.681
6	7.746	7.810	7.874	7.937	8.000	8.062	8.124	8.185	8.246	8.307
7	8.367	8.426	8.485	8.544	8.602	8.660	8.718	8.775	8.832	8.888
8	8.944	9.000	9.055	9.110	9.165	9.220	9.274	9.327	9.381	9.434
9	9.487	9.539	9.592	9.644	9.695	9.747	9.798	9.849	9.899	9.950

例 20

4.23

```
#include <stdio.h>
#include <math.h>
main()
{
    int m, n, i;

    for (m = 0; m < 10; m++)
    {
        printf("%7d", m);                /*打印表头*/
    }
    printf("\n");
    for (n = 0; n < 10; n++)              /*乘数n从1变化到9*/
    {
        printf("%d", n);                 /*输出每行的头数字*/
        for (m=0; m<10; m++)              /*被乘数m从1变化到9*/
        {
            printf(" %4.3f ", sqrt(n * 10 + m));
        }
        printf("\n");                    /*输出第m行n列中的值*/
        /*输出换行符,准备打印下一行*/
    }
}
```

爱因斯坦数学题。爱因斯坦曾出过这样一道数学题：有一条长阶梯，若每步跨2阶，则最后剩下1阶，若每步跨3阶，则最后剩下2阶，若每步跨5阶，则最后剩下4阶，若每步跨6阶，则最后剩下5阶，只有每步跨7阶，最后才正好1阶不剩。请问，这条阶梯共有多少阶？

算法思想：设阶梯数为 x ，则根据题意：阶梯数满足下面的关系式：

$$x \% 2 == 1 \ \&\& \ x \% 3 == 2 \ \&\& \ x \% 5 == 4 \ \&\& \ x \% 6 == 5 \ \&\& \ x \% 7 == 0$$

采用穷举法对 x 从1开始实验，可计算出这条阶梯共有多少阶。

```
int main()
{
    int x = 1, find = 0;

    while (!find)
    {
        if (x % 2 == 1 && x % 3 == 2 && x % 5 == 4 && x % 6 == 5
            && x % 7 == 0)
        {
            find = 1;
        }
        x++;
    }
    printf(" x = %d\n", x);
}
```

程序运行结果为：
x = 120

三色球问题。若一个口袋中放有12个球，其中有3个红色的，3个白色的，6个黑色的，从中任取8个球，问共有多少种不同的颜色搭配？

算法思想：设任取的红球个数为 i ，白球个数为 j ，黑球个数为 k ，根据题意应有： $i+j+k=8$ ， $0 \leq i \leq 3$ ， $0 \leq j \leq 3$ ， $0 \leq k \leq 6$ 。若红球和白球个数的取值范围是 $0 \sim 3$ ，在红球和白球个数确定的条件下，黑球个数的取值范围应为 $k=8-i-j$ ，只要满足 $k \leq 6$ ， i 、 j 、 k 的组合即为所求。

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

```
main()
```

```
{
```

```
    int i, j, k;
```

```
    for (i = 0; i <= 3; i++)
```

```
    {
```

```
        for (j = 0; j <= 3; j++)
```

```
        {
```

```
            for (k = 0; k <= 6; k++)
```

```
            {
```

```
                if (i + j + k == 8)
```

```
                {
```

```
                    printf("i=%d, j=%d, k=%d\n", i, j, k);
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

程序运行结果:

i=0,j=2,k=6

i=0,j=3,k=5

i=1,j=1,k=6

i=1,j=2,k=5

i=1,j=3,k=4

i=2,j=0,k=6

i=2,j=1,k=5

i=2,j=2,k=4

i=2,j=3,k=3

i=3,j=0,k=5

i=3,j=1,k=4

i=3,j=2,k=3

i=3,j=3,k=2

鸡兔同笼，共有98个头，386只脚，编程求鸡、兔个多少。

算法思想：设鸡数为 x ，兔数为 y ，据题意有： $x+y=98$ ， $2x+4y=386$ 。采用穷举法， x 从1变化到97， y 取 $98-x$ ，如果 x 、 y 同时满足条件 $2x+4y=386$ ，则打印 x 、 y 的值。

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

```
main()
```

```
{
```

```
    int x, y;
```

```
    for (x = 1; x <= 97; x++)
```

```
    {
```

```
        y = 98 - x;
```

```
        if (2 * x + 4 * y == 386)
```

```
        {
```

```
            printf("x = %d, y = %d", x, y);
```

```
        }
```

```
    }
```

```
}
```

程序运行结果为:

x = 3, y = 95

我国古代的《张丘建算经》中有这样一道著名的百鸡问题：“鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一。百钱买百鸡，问鸡翁、母、雏各几何？”意为：公鸡每只5元，母鸡每只3元，小鸡3只1元。用100元买100只鸡，问公鸡、母鸡和小鸡各能买多少只？

算法思想：设公鸡、母鸡、小鸡数量分别为 x ， y ， z ，依题意列出方程组： $x+y+z=100$ ， $5x+3y+z/3=100$ ，采用穷举法求解，因100元买公鸡最多可买20只，买母鸡最多可买33只，所以， x 从0变化到20， y 从0变化到33，则 $z=100-x-y$ ，只要判断第二个条件是否满足即可。

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

```
main()
```

```
{
```

```
    int x, y, z;
```

```
    for (x = 0; x <= 20; x++)
```

```
    {
```

```
        for (y = 0; y <= 33; y++)
```

```
        {
```

```
            z = 100 - x - y;
```

```
            if (5*x + 3*y + z/3.0 == 100)
```

```
            {
```

```
                printf("x=%d, y=%d, z=%d\n", x, y, z);
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

程序运行结果为:

x=0,y=25,z=75

x=4,y=18,z=78

x=8,y=11,z=81

x=12,y=4,z=84

用一元五角钱人民币兑换5分、2分和1分的硬币（每一种都要有）共100枚，问共有几种兑换方案？每种方案各换多少枚？

算法思想：设5分、2分和1分的硬币各换 x 、 y 、 z 枚，据题意有：

$x+y+z=100$ ， $5x+2y+z=150$ ，由于每一种硬币都要有，故5分硬币最多可换28枚，2分硬币最多可换73枚，1分硬币可换 $100-x-y$ 枚， x 、 y 、 z 只需满足第二个方程即可打印，对每一组满足条件的 x 、 y 、 z 值用计数器计数即可得到兑换方案的数目。

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

```
main()
```

```
{
```

```
    int x, y, z, count = 0;
```

```
    for (x = 1; x <= 28; x++)
```

```
    {
```

```
        for (y = 1; y <= 73; y++)
```

```
        {
```

```
            z = 100 - x - y;
```

```
            if (5*x + 2*y + z == 150)
```

```
            {
```

```
                count++;
```

```
                printf("%d, %d, %d\n", x, y, z);
```

```
            }
```

```
        }
```

```
    }
```

```
    printf("count = %d\n", count);
```

```
}
```

程序运行结果为:

1,46,53

2,42,56

3,38,59

4,34,62

5,30,65

6,26,68

7,22,71

8,18,74

9,14,77

10,10,80

11,6,83

12,2,86

count = 12

Questions & Answers