

# 上次作业参照答案



## //5.3

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

```
main()
```

```
{
```

```
    int a=3,b=4,c=5;
```

```
    printf("输出成果是: \n" );
```

```
    printf("a+b>c&&b==c的逻辑体现式的值是%d\n", (a+b>c&&b==c));
```

```
    printf("a||b+c&&b-c的逻辑体现式的值是%d\n", (a||b+c&&b-c));
```

```
    printf("!(a>b)&&!c||1的逻辑体现式的值是%d\n", (!(a>b)&&!c||1));
```

```
    printf("!(x=a)&&(y=b)&&0的逻辑体现式的值是%d\n", !(x=a)&&(y=b)&&0);
```

```
    printf("!(a+b)+c-1&&b+c/2的逻辑体现式的值是%d\n", !(a+b)+c-1&&b+c/2));
```

```
}
```



## //5.4

```
#include<stdio.h>
main()
{
    int a,b,c,t,max;
    printf("请输入3个整数: \n");
    scanf("%d,%d,%d",&a,&b,&c);
    t=(a>b)?a:b;
    max=(t>c)?t:c;
    printf("3个整数的最大数是
%d\n",max);
}
```

## //5.4另解

```
#include<stdio.h>
void main()
{
    int a,b,c;
    scanf("%d,%d,%d",&a,&b,&c);
    if(a>=b&&a>=c)
        printf("%d",a);
    if(b>=a&&b>=c)
        printf("%d",b);
    if(c>=a&&c>=b)
        printf("%d",c);
}
```

## #include<stdio.h> //5.4另解

```
void main()
{
    int a,b,c,t;
    scanf("%d,%d,%d",&a,&b,&c);
    if(a>=b)
        t=a;
    else t=b;
    if(t>=c)
        printf("%d",t);
    else printf("%d",c);
}
```

## //5.5

```
#include<stdio.h>
main()
{
    int x,y;
    printf("请输入x: \n");
    scanf("%d",&x);
    if(x<1)
        y=x;
    else if (x<10)
        y=2*x-1;
    else
        y=3*x-11;
    printf("y=%d\n",y);
}
```

## //5.9

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

```
main()
```

```
{
```

```
    int t,a,b,c,d;
```

```
    printf("请输入4个整数: \n");
```

```
    scanf("%d,%d,%d,%d",&a,&b,&c,&d);
```

```
    printf("a=%d,b=%d,c=%d,d=%d\n",a,b,c,d);
```

```
    if(a>b)
```

```
    {t=a;a=b;b=t;}
```

```
    if(a>c)
```

```
    {t=a;a=c;c=t;}
```

```
    if(a>d) //a依次和b、c、d比较
```

```
    {t=a;a=d;d=t;}
```

```
    if(b>c)
```

```
    {t=b;b=c;c=t;}
```

```
    if(b>d) //b和c、d比较
```

```
    {t=b;b=d;d=t;}
```

```
    if(c>d)
```

```
    {t=c;c=d;d=t;} //c和d比较
```

```
    printf("排序成果是: \n");
```

```
    printf("%d,%d,%d,%d\n",a,b,c,d);
```

```
}
```



## //5.7

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

```
{
    int i,a,b,c,d,e;
    printf("请输入一种正整数\n");
    scanf("%d",&i);
    if(i>=10000&&i<=99999)
    {
```

```
        printf("这是一种五位数; \n");
        a=i/10000;
        b=(i-a*10000)/1000;
        c=(i-a*10000-b*1000)/100;
        d=(i-a*10000-b*1000-c*100)/10;
        e=i-a*10000-b*1000-c*100-d*10;
        printf("其数位分别是: %d,%d,%d,%d,%d\n",a,b,c,d,e);
        printf("其逆序数位分别是: %d,%d,%d,%d,%d\n",e,d,c,b,a);
    }
```

...//其他几种同上处理

还能够按如下措施求解:

```
e=i%10;
d=i/10%10;
c=i/100%10; //c=i/10/10%10;
b=i/1000%10;
//b=i/10/10/10%10;
a=i/10000
```

# 上讲回忆



- 关系运算符、关系体现式；
- 逻辑运算符、逻辑体现式；
- 条件运算符的使用；
- 条件判断语句if等的使用；
- 多分支选择语句的使用；
- 选择构造程序设计；



# 本讲要点



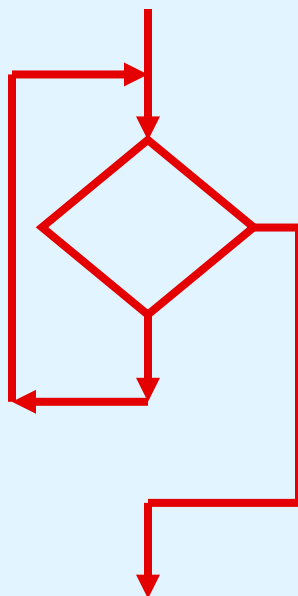
- 掌握3种常见的C循环语句;
- 了解、掌握循环构造程序设计的3要素;
- 掌握循环构造语句执行的顺序;
- 掌握简朴的循环构造程序设计;



# 循环构造程序设计

吉祥

```
...  
main()  
{  
...  
...  
}
```



# 例1：从键盘输入三角形的三条边长，求三角形的面积。

*/\*第3讲源程序\*/*

```
#include<stdio.h>
#include<math.h>
main()
{
int a,b,c; /*三个变量，用来表达
三条边长*/
float s,area; /*area变量用来表达
面积*/
scanf("%d,%d,%d",&a,&b,&c);
/*输入三条边长的值*/
s=(a+b+c)/2.0;
area=sqrt(s*(s-a)*(s-b)*(s-c));
/*海伦公式求面积*/
printf("The resule is
area=%.2f\n",area);
}
```

*/\*第4讲源程序\*/*

```
#include<stdio.h>
#include<math.h>
main()
{
int a,b,c; /*三个变量，用来表达三条边长
*/
float s,area; /*area变量用来表达面积*/
scanf("%d,%d,%d",&a,&b,&c); /*输入
三条边长的值*/
if(a+b>c&&a+c>b&&b+c>a)
{ s=(a+b+c)/2.0;
area=sqrt(s*(s-a)*(s-b)*(s-c)); /*海伦
公式求面积*/
printf("The resule is
area=%.2f\n",area);
}
else
printf("你输入的数据无效\n");
}
```



# 分析



- 3讲中的源程序只考虑了最理想的情况，没有对是否能够构成三角形作出判断，4讲中作了补充，那么4讲中的例程是否还有不足？
- 4讲中对于输入的三边，假如不能构成三角形，则程序直接输出“输入无效”后而结束。一般，我们希望程序能够这么：假如顾客输入的数据无效，给出提醒，能够反复输入，直到输入有效的数据。



# 改善后的源程序

```
#include<stdio.h>
#include<math.h>
main()
{
    int a,b,c;        /*三个变量，用来表达三条边长*/
    float s,area;     /*area变量用来表达面积*/
    scanf("%d,%d,%d",&a,&b,&c); /*输入三条边长的值*/
    while(a+b<=c||a+c<=b||b+c<=a) //当不能构成三角形的时候
    {
        printf("你输入的数据无效,请重新输入\n");
        scanf("%d,%d,%d",&a,&b,&c);
    }
    s=(a+b+c)/2.0;
    area=sqrt(s*(s-a)*(s-b)*(s-c)); /*海伦公式求面积*/
    printf("The resule is area=%.2f\n",area);
}
```

循环条件

循环体

# 循环构造三要素



- 循环变量赋初值

- 循环的起始，从哪里开始。

- 循环条件

- 满足什么条件时才执行循环；也即告知循环终止的条件。

- 循环体

- 要被反复执行的对象



# While循环语句

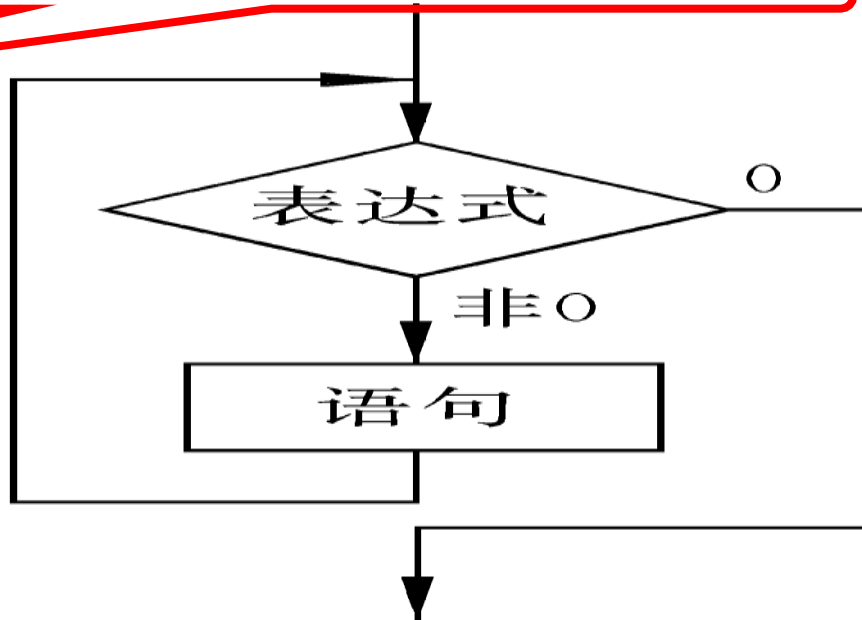
## ■ 三种基本循环语句

体现式能够是关系体现式、逻辑体现式等

### ① while(体现式)

```
{  
    循环体语句  
}
```

- 执行过程：当体现式为非0值时，执行while语句中的内嵌语句。其特点是：先判断体现式，后执行语句。



e.g. `i=0;`

初值

`while(i<5)`

循环条件

{

`printf("*");`

循环体

`i++;`

}

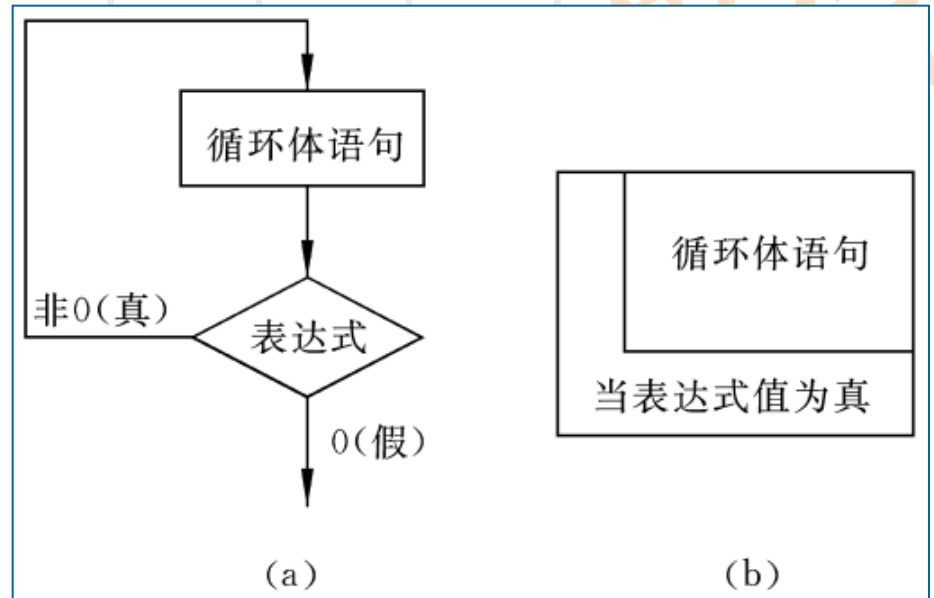
# do...while循环语句

②

```
do  
{  
    循环体语句  
}while(体现式);
```

体现式能够是  
关系体现式、  
逻辑体现式等

执行过程：先执行一次指定的循环体语句，然后鉴别体现式，当体现式的值为非零（“真”）时，返回重新执行循环体语句，如此反复，直到体现式的值等于0为止，此时循环结束。



```
e.g. i=0;  
do {  
    printf("*");  
    i++;  
} while(i<5);
```

# 例：while和do...while的比较

```
#include<stdio.h>
main( )
{
int sum=0,i;
scanf(“%d”,&i);
while(i<=10)
{
sum+=i;
i++;
}
printf(“sum=%d\n”,sum);
}
```

```
#include<stdio.h>
main( )
{
int sum=0,i;
scanf(“%d”,&i);
do
{
sum+=i;
i++;
} while(i<=10);
printf(“sum=%d\n”,sum);
}
```

# 分析

- 当第一次输入的*i*的值不小于10时，两个程序的成果相同；
- 当第一次输入的*i*的值不小于10时，成果不同；
- 结论：当while背面体现式的值第一次为真时，两种循环的成果相同，不然，循环的成果不同

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/378037117066006130>