

分支 循环

1. if语句

1.1 简述

```
1 if(表达式)
2 {
3     语句1
4 }
```

比如

```
1 if(a>=10)
2 {
3     a++;
4 }
```

这里要补充几个内容。

一个是关于判断的部分。在c语言中，我们如果想要判断两个数之间的大小应该怎么做呢？

判断大于或小于很简单，就是我们经常使用的大于号与小于号（">"和"<"）。

在需要大于等于或小于等于时，我们使用两个符号来代替，即">="和"<="。

但是我们之前提到过，在C语言中等于号“=”表示赋值的意思，所以我们必须要用两个连续的等于号才能表示等于号原本的意思，即“==”。还有一个是刚刚代码块中的这一句

a++;

这一句表示自增，也就是说下面两句的作用是一样的：

```
1 a++;
2 a=a+1;
```

同理还有

```
1 a--;  
2 a=a-1;
```

此外也有一些比较省事的写法：

```
1 a+=2;  
2 a-=2;  
3 a*=2;  
4 a/=2;
```

还有关于优先级的判断，我们都知道在数学中，乘号的优先级是大于加号的，那么我们如果想对一个式子先算加法再算乘法就要用到括号了，比如：

$(1+2)*3$

在C语言中也可以这样，比如

```
1 int a=1;  
2 int b=2;  
3 int c=3;  
4 int d=(a+b)*c;
```

但是此外，假如我们在判断的时候要用到类似的情况时也可以用到括号，比如：

```
1 if((a+b)*c > 4)  
2 {  
3     printf("GOOD\n");  
4 }
```

在之后的编程中可能会遇到很复杂的判断场景，推荐多使用括号帮助自己来理清整个结构的关系。

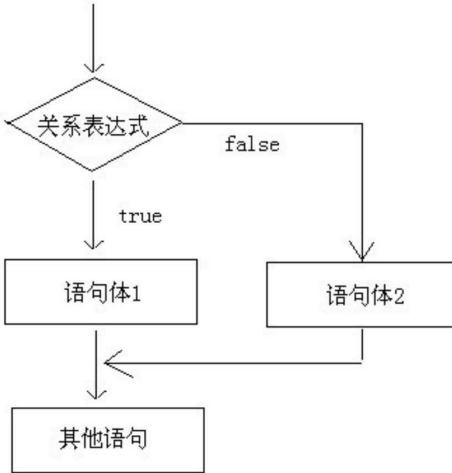
if语句还有一种表现形式是这样：

```
1 if(表达式)  
2     语句1  
3 else  
4     语句2
```

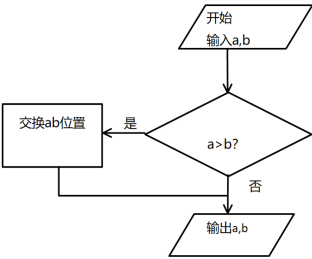
比如

```
1  if(a>=1)
2      a++;
3  else
4      a--;
```

当“if”，或者接下来讲到的其他语句后只有一个语句时可以省略大括号，但一般不建议这样做。其流程图如下：



有个问号表达式，也就是当某一个条件表达式是真时，问号表达式的结果为冒号前的结果，当条件表达式是假时，问号表达式的结果为冒号后的结果。这里我们可以分析一下，问号表达式其实是一个条件判断功能表达式，可以举个简单的例子，如为了将a和b对换，不能把两个变量直接互相赋值，也就是不能用下面的办法：



a=b; //把变量b的值赋给变量a，a的值等于b的值

b=a; //再把变量a的值赋给变量b，变量b值没有改变

为了实现互换，必须借助于第3个变量。可以这样考虑：将A和B两个杯子中的水互换，用两个杯子的水倒来倒去的办法是无法实现的。必须借助于第3个杯子C，先把A杯的水倒在C杯中，再把B杯的水倒在A杯中，最后再把C杯的水倒在B杯中，这就实现了两个杯子中的水互换。这是在程序中实现两变量换值的算法。

1.2 程序举例

```
1 #include <stdio.h>
2
3 int main() {
4     int a, b, t;
5     scanf("%d%d", &a, &b);
6     if (a > b) {
7         //如果a>b, 则:
8         //if(表达式, 满足(或不为0)则执行下面语句)
9         //将a和b的值互换, 使用额外变量
10        t = a;
11        a = b;
12        b = t;
13        //不使用额外变量的交换方法
14        //a ^= b; b ^= a; a ^= b;
15        //a += b; b = a - b; a -= b;
16    }
17    printf("%d,%d", a, b);
18    return 0;
19 }
```

输入示例1:

2 6

运行结果:

2,6

输入示例2:

514 114

运行结果:

114,514

下面我们来看一个例子说明if else语句的功能，二月份一共有28天，但当闰年时，二月份就有29天，闰年的计算公式为：能被4整除，不能被100整除的年是闰年，能被400整除的看也是闰年。所以我们可以编写一个用于计算二月份天数的程序：

```

1 #include <stdio.h>
2
3 int main() {
4     int year = 2016;
5     int daysOfFebruary;
6     if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
7         //如果该年分为闰年，则：
8         //闰年判定：当不是整百倍的情况就/4没有余数
9         //如果是整百倍的/400没有余数
10        daysOfFebruary = 29;
11    }
12    else {
13        //否则，则：
14        daysOfFebruary = 28;
15    }
16    printf("%d\n", daysOfFebruary);
17 }

```

运行结果：

29

1.3 else if

1. 简述

有时候，我们需要将条件划分为多个不同的情况，往往是需要多种情况的分支，所以一个if else语句是不能满足我们的需要的。

```

1 if(表达式1)
2 {
3     语句1
4 }
5 else if(表达式2)
6 {
7     语句2
8 }
9 ...
10 else if(表达式m)
11 {
12     语句m
13 }
14 else
15 {
16     语句m+1
17 }

```

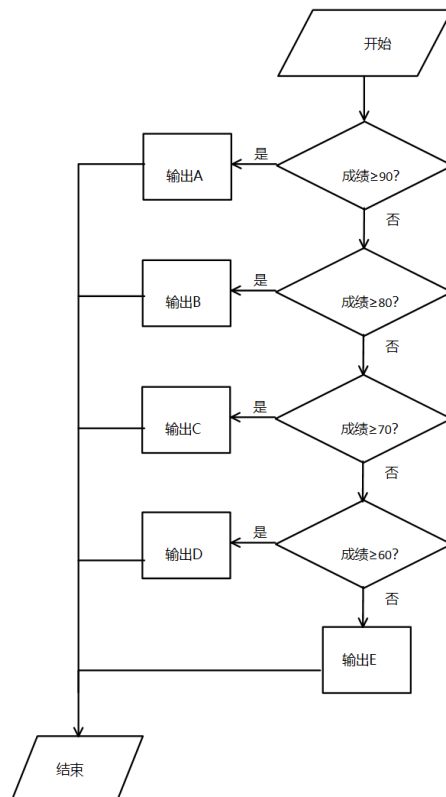
比如我可以这样写：

```
1  if(a>1)
2  {
3      a++
4  }
5  else if(a==1)
6  {
7      a=2;
8  }
9  else
10 {
11     a--;
12 }
```

要注意的是，每一个判断是依次进行的，只有上一个不满足才会进入下一个判断，并且后面的判断结束之后不会再判断之前的条件。

1.4 程序举例

一个学生期末考试的成绩单在90分为A，80分以上时为B，70分以上为C，60分以上为D，而小于60为E。如果要用if else语句来避免多次嵌套很多层if语句,if else一旦匹配，后面语句不会继续执行：



```

1 #include <stdio.h>
2
3 int main()
4 {
5     int score = 78;
6     if (score >= 90) //如果分数≥90
7     {
8         printf("A\n");
9     }
10    else if (score >= 80) //否则，如果分数≥80
11    {
12        printf("B\n");
13    }
14    else if (score >= 70)
15    {
16        printf("C\n");
17    }
18    else if (score >= 60)
19    {
20        printf("D\n");
21    }
22    else //否则
23    {
24        printf("E\n");
25    }
26 }

```

运行结果：

C

2. Switch语句

2.1 简述

if语句只有两个分支可供选择，而实际问题中常常需要用到多分支的选择。例如，学生成绩分类(85分以上为A等，70~84分为B等，60~69分为C等)，人口统计分类(按年龄分为老、中、青、少、儿童)，工资统计分类，银行存款分类等。当然这些都可以用if语句来处理，不过面对大量情况需要列举时，更适合使用switch语句，用switch写出来的程序可读性好。

switch和if的最大区别是switch是得单独列出每一种情况(枚举)，但if可以使用表达式来表示多种情况。

总结switch语句表达式如下：

```

1 switch(表达式)

```

```
2 {
3     case 常量1: 语句1
4     case 常量2: 语句2
5     ...
6     case 常量n: 语句n
7     default: 语句n+1
8 }
```

注意`switch`后面括号内的表达式类型为**整数类型**（包括字符型）。举个栗子，我可以这样写：

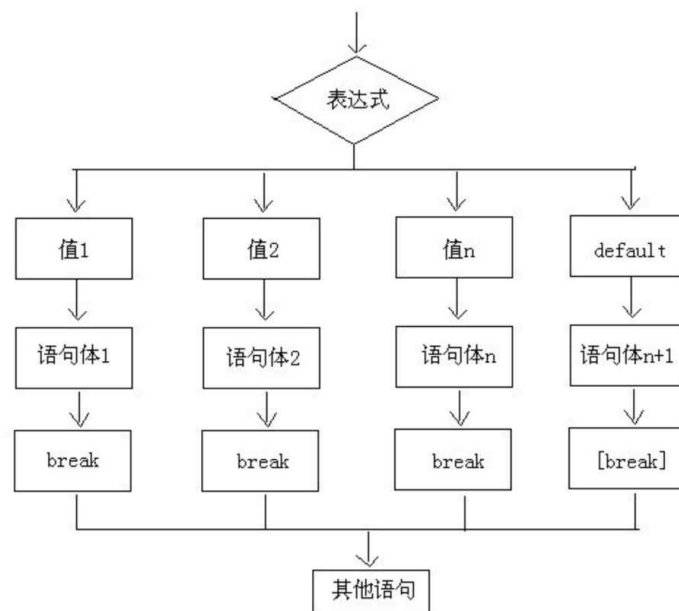
```
1 switch(a)
2 {
3     case 1: a++;
4     case 2: a--;
5     default: a=3;
6 }
```

但是注意，`switch`语句不会自动跳出循环，也就是说，按照上面那个示例来的话，如果最开始变量"`a`"等于1的

话，最后变量会依次执行所有语句。那么这时就需要有一个办法让程序执行完之后就退出，恰好C语言中就有

这样的语句，所以我们可以这样写：

```
1 switch(a)
2 {
3     case 1: a++;break;
4     case 2: a--;break;
5     default: a=3;break;
6 }
```

```

1 switch(表达式)
2 {
3     case 常量1: 语句1;break;
4     case 常量2: 语句2;break;
5     case 常量n: 语句n;break;
6     default 语句 n+1;break;
7 }

```

2.2 程序举例

例如判定字符s是不是元音字母aeiou，使用两种语句写出来的结果分别是：

```

1 if(s=='a' || s=='e' || s=='i' || s=='o' || s=='u' || s=='A' || s=='E' || s=='I' || s=='O' || s=='U')
2 {
3     //...
4 }
5 switch(s) //switch(检查表达式)
6 {
7     case a: //该表达式是否为a?
8     case e: //该表达式是否为e?
9     case i:
10    case o:
11    case u:
12    case A:
13    case E:
14    case I:
15    case O:
16    case U:
17        //...

```

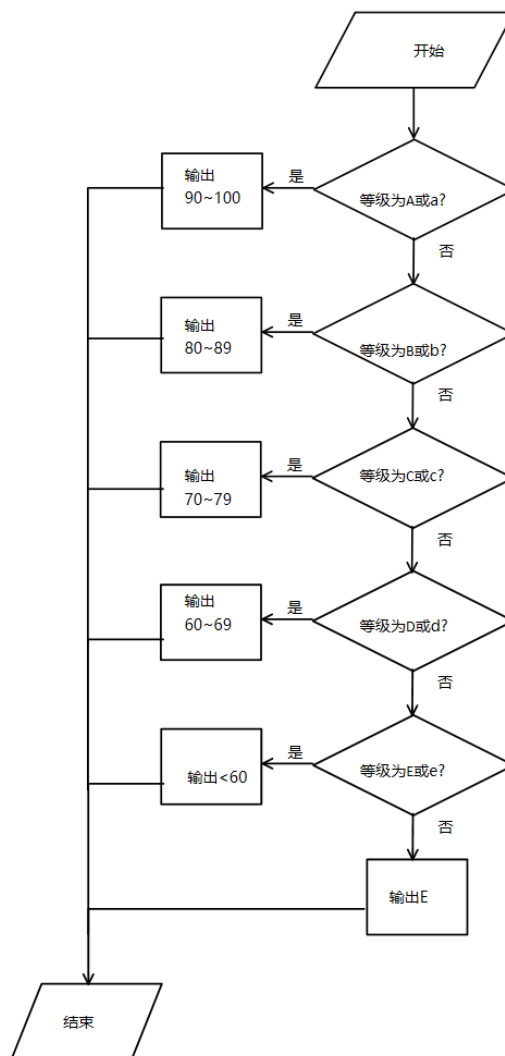
```

18     break;
19     //执行到这里跳出switch判断，若不用break则会执行道下面的default的语句。
20     default://匹配失败处理
21     //...
22 }

```

当变量名很长时，程序可读性会严重下降。

【例4.6】要求按照考试成绩的等级输出百分制分数段，A等为90分以上，B等为80～89分，C等为70～79分，D等为60～69分，E为60分以下。成绩的等级由键盘输入。这是一个多分支选择问题，根据百分制分数将学生成绩分为4个等级，。用switch语句，进行一次检查即可得到结果



```

1  #include <stdio.h>
2
3  int main() {
4      char grade;
5      scanf("%c", &grade);
6      printf("Your score:");
7      switch (grade) {
8          case 'A':

```

```

9         case 'a':
10             printf("90~100\n");
11         break;
12         case 'B':
13         case 'b':
14             printf("80~89\n");
15         break;
16         case 'C':
17         case 'c':
18             printf("70~79\n");
19         break;
20         case 'D':
21         case 'd':
22             printf("60~69\n");
23         break;
24         case 'E':
25         case 'e':
26             printf("<60");
27         break;
28         default:
29             printf("enter data error!\n");
30     }
31     return 0;
32 }

```

输入示例1:

A

运行结果:

Your score:90~100

输入示例2:

d

运行结果:

Your score:60~69

输入示例3:

G

运行结果:

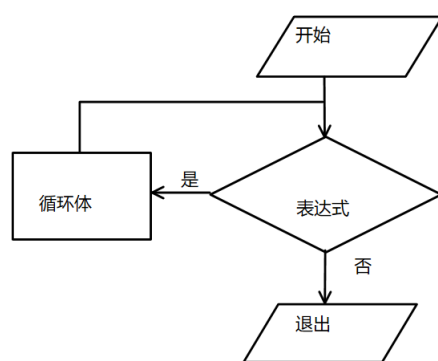
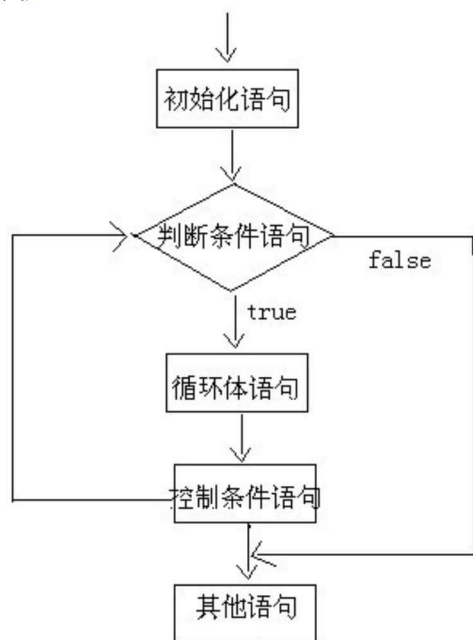
Your score: enter data error!

3. while语句

3.1 简述

while语句中的“语句”就是循环体。循环体只能是一个语句，可以是一个简单的语句，还可以是复合语句（用花括号括起来的若干语句）。执行循环体的次数是由循环条件控制的，这个循环条件就是上面一般形式中的“表达式”，它也称为循环条件表达式。当此表达式的值为“真”（以非0值表示）时，就执行循环体语句；为“假”时（以0表示）时，就不执行循环体语句。

框图为：



3.2 程序举例

例：求 $1+2+3+4+\cdots+100$

```
1 #include <stdio.h>
2
3 int main() {
4     int i = 1, sum = 0;
5     while (i <= 100) {
6         sum = sum + i;
```

```
7         i++;
8     }
9     printf("sum=%d\n", sum);
10    return 0;
11 }
```

运行结果：

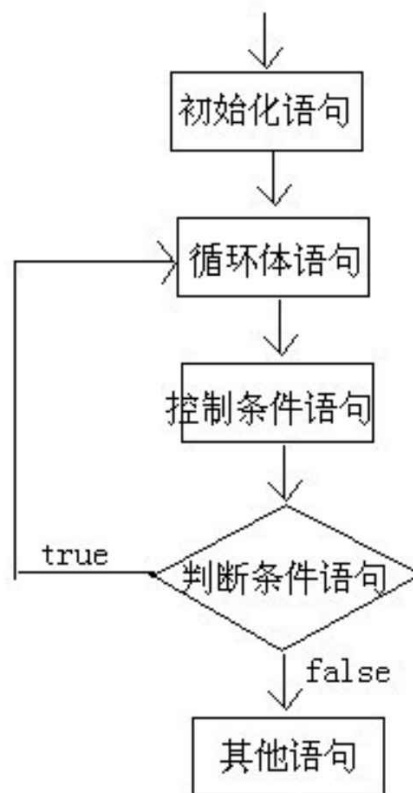
sum=5050

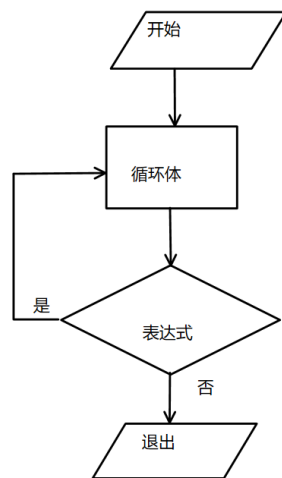
4. do...while循环

4.1 简述

do...while和while的区别在于：前者是先进进行判断再决定是否执行执行循环体并循环，后者是先执行循环体再判断是否循环。

框图为：

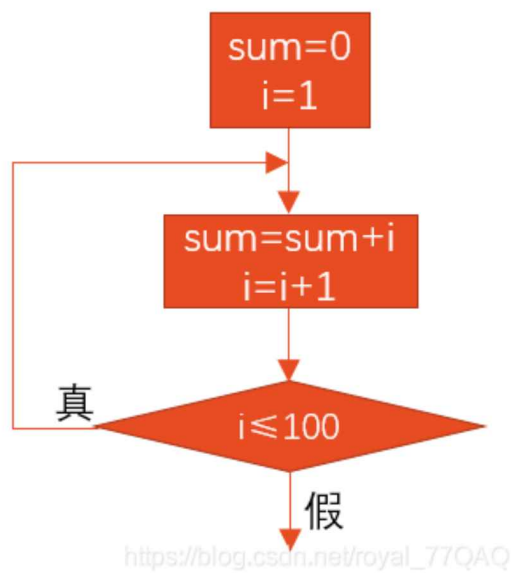
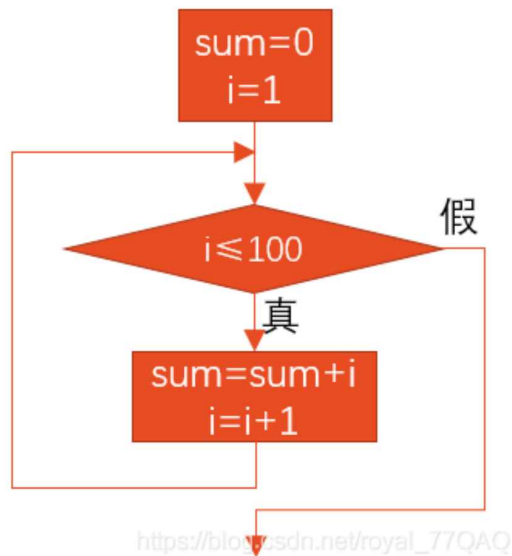




4.2 程序举例

```
1  #include <stdio.h>
2  int main()
3  {
4      int i=1,sum=0;
5      do
6      {
7          sum=sum+i;
8          i++;
9      }while(i<=100);
10     printf("sum=%d\n",sum);
11     return 0;
12 }
```

注意事项： 在一般情况下，用while语句和用do...while语句处理同一问题时，若二者的循环体部分是一样的，那么结果也一样。但是如果while后面的表达式一开始就为假(0值)时，两种循环的结果是不同的。



例：输入为n，输出为2的n次方:

```

1  #include <stdio.h>
2  int main()
3  {
4      int n = 0, i = 1, sum = 1;
5      scanf("%d", &n);
6      do
7      {
8          sum += sum;
9          i++;
10     }while (i <= n) ;
11     printf("2^d=%d\n", n, sum); //输出 do...while 的计算结果
12     sum = 1, i = 1;
13     while (i <= n)
14     {
15         sum += sum;
16         i++;
    
```

```
17     }
18     printf("2^%d=%d\n", n, sum); //输出 while 的计算结果
19     return 0; }
```

输入示例1:

10

运行结果:

2^10=1024

2^10=1024

输入示例2:

0

运行结果:

2^0=2

2^0=1

可见，两者最大区别就在一开始执行的阶段，由于do...while先执行了循环体在判断，导致其输出错误的结果2，当两者都执行了循环体时，其作用相同。

5. for循环

5.1 简述

```
1 for (表达式1;表达式2;表达式3)
2 {
3     (循环内容)
4 }
```

表达式1：设置循环变量的初值。

表达式2：循环条件表达式，当该条件符合时，执行循环。

表达式3：使循环变量增值

执行for语句的顺序是：表达式1→若满足表达式2→循环内容→表达式3→若满足表达式2→…→不满足表达式2→结束。

5.2 程序举例


```
1 #include <stdio.h>
2
3 int main() {
4     for (int i = 0; i < 5; i++) {
5         //第一个参数是进入循环前执行的，一般用于初始化
6         //第二个是循环条件，满足该条件才会继续循环
7         //第三个条件是单次循环执行完才会执行的，一般是执行循环计数条件
8         printf("%d", i);
9     }
10    return 0;
11 }
```

运行结果：

01234

执行过程：

首先给循环变量i赋值初值为0；

i=0时，满足i<5，执行for括号内的程序，执行i++，此时i=1；

i=1时，满足i<5，执行for括号内的程序，执行i++，此时i=2；

i=2时，满足i<5，执行for括号内的程序，执行i++，此时i=3；

i=3时，满足i<5，执行for括号内的程序，执行i++，此时i=4；

i=4时，满足i<5，执行for括号内的程序，执行i++，此时i=5；

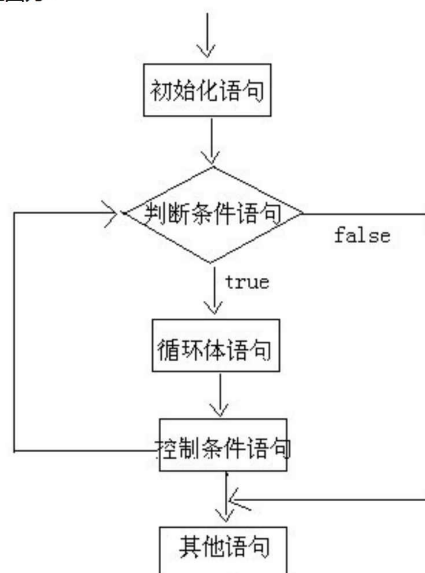
i=5时，不满足i<5，程序结束。

用for循环求1+2+...+100；

```
1 #include <stdio.h>
2 int main()
3 {
4     int i;
5     int sum = 0;
6     for(i = 0; i <= 100; i++)
7     {
8         sum = sum + i;
9     }
10    printf("%d\n", sum);
11 }
```

//输出为5050

框图为：



我们再介绍一下两个功能语句 “**break**” 和“**continue**”。

其中 “**break**” 我们之前已经介绍过了，用于跳出某个循环，而“**continue**”的**用法**与 “**break**” 相同，但**功能**上会有区别。

“**continue**”是用于结束当前的一次循环，继续下一次循环，

6. break

6.1 简述

前面我们学switch语句时可以知道，使用break语句能够使流程跳出switch结构，继续执行switch语句下面的语句。（不理解break作用的可以试着把switch例程中的break去掉，观察运行结果有何不同）

同理的，break语句也能在其它循环语句之中使用（但不能单独使用）。

6.2 程序举例

```
1  #include <stdio.h>
2
3  int main() {
4      for (int i = 0, k = 0; i < 10; i++) {
5          k++;
6          if (k == 5)
7              break;
8          printf("i=%d ", i);
9          printf("k=%d\n", k);
10     }
```

```
11     return 0;
12 }
```

运行结果：

i=0 k=1

i=1 k=2

i=2 k=3

i=3 k=4

可以看到，如果没有第八行的语句，运行结果会一直到i=9 k=10，也就是k==5时，执行break语句会跳出for循环。

7. continue

7.1 简述

有时候不想终止整个循环，而只希望结束本次循环，再接着执行下次循环时，可以使用continue语句。

7.2 程序举例

在break语句中的程序举例中将break改为continue即可显著的看到差别：

```
1  #include <stdio.h>
2
3  int main() {
4      for (int i = 0, k = 0; i < 10; i++) {
5          k++;
6          if (k == 5)
7              continue;
8          printf("i=%d ", i);
9          printf("k=%d\n", k);
10     }
11     return 0;
12 }
```

运行结果：

i=0 k=1

i=1 k=2

```
i=2 k=3  
i=3 k=4  
i=5 k=6  
i=6 k=7  
i=7 k=8  
i=8 k=9  
i=9 k=10
```

- 在第四次打印出*i*=3 *k*=4后，执行*i*++。此时*i*=4，*k*=4。
- 第五次循环，执行*k*++。此时*i*=4，*k*=5。
- 进入if判断语句。由于满足*k*==5的条件，执行跳过此次循环的continue语句，于是后面的printf指令没有被执行。
- 执行*i*++。此时*i*=5，*k*=5。
- 第六次开始*k*的值为5，执行*k*++，此时*i*=5，*k*=6。我们可以看到，这次与break的例子中不同的是，程序仍在for循环里进行。
- 进入if判断语句。不满足*k*==5，往下执行printf指令。

下面我们说一下如果你的程序出了逻辑上的问题，应该如何找到并改正这个问题（如果语法问题编译器会直接报错）。

调试程序分为两种思路，一个是打印，一个是断点。

打印就是在程序运行中间打印出当前变量的值，得知程序运行的当前状态，这种方法比较简单，就是在需要的地方加上“**printf**”函数即可。（比如.....）

8. 课后练习

Q1: 给出三个整数，要求把这三位整数从小到大排序。

（见洛谷习题：<https://www.luogu.com.cn/problem/P5715>）（10分）

Q2: 输入年份和月份，输出这一年的这一月有多少天。需要考虑闰年。

（见洛谷习题：<https://www.luogu.com.cn/problem/P5716>）（10分）

Q3: 使用for循环将1到100相加，打印出最终结果sum。（10分）

Q4: 使用for循环计算斐波那契数列，要求：输入任意整数n，输出第n位斐波那契数列。斐波那契数列有如下递推公式：即该数列前十项为：1 1 2 3 5 8 13 21 34 55

(见洛谷习题：<https://www.luogu.com.cn/problem/P1720>) (10分)

Q5: 输入n个整数，求这n个整数中最小值。

(见洛谷习题：<https://www.luogu.com.cn/problem/P5718>) (10分)

Q6: 使用for循环将1到100递归相加，当总和sum为1275时结束循环。 (10分)

Q7: 输入为n个不同的字符，输出之前读取到的字符，到第一个' a' 截止。例：输入vcdefegavds，输出：vcdefega (10分)

Q8: 使用continue语句和for循环语句，输出100~200内的奇数。 (10分)

Q9: 输入为n个不同的字符，输出之前读取到的字符，但不输出所有的' d' 和' b' 。例：输入vcdefbbebgavbds，输出：vcefegavs (10分)

Q10: 国王将金币作为工资，发放给忠诚的骑士。第一天，骑士收到一枚金币；之后两天（第二天和第三天），每天收到两枚金币；之后三天（第四、五、六天），每天收到三枚金币；之后四天（第七、八、九、十天），每天收到四枚金币……；这种工资发放模式会一直这样延续下去：当连续 n 天每天收到枚金币后，骑士会在之后的连续 天里，每天收到 枚金币。请计算在前天里，骑士一共获得了多少金币。(提示：本题可以依照题意暴力循环穷举或用数学方法解答，可用洛谷的在线评测来测试程序正确性)

例1：输入：6 输出：14

例2：输入：1000 输出29820

例3：输入15 输出：55

(见洛谷习题：<https://www.luogu.com.cn/problem/P2669>) (10分)