选择结构

作用：程序中的每条语句是按照各个语句的先后顺序依次执行的，很多问题顺序结构无法解决，要通过给定条件的判断来决定要执行的操作。

```c
#include<stdio.h>
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    printf("%d",a);
    printf("%d",b);
    return 0;
}
```

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

```
    if(a>0)
    {
    printf("%d",a);
    }
    if(b>0)
    {
    printf("%d",b);
    }
    return 0;
}
```

选择语句的类型

单分支if选择结构

```
#include<stdio.h>
int main()
{
    if(条件1)
    {
        printf("%d",1);
    }
    printf("%d",0);
    return 0;
}
```

双分支if选择结构 if...else

```
if(条件1)
    printf( "%d",1);
    else
    printf("%d",0);
```

条件运算符

表达式1？表达式2：表达式3

```
条件1？printf("%d",1):printf("%d",0)
```

if语句的嵌套

在if语句中又包含一个或多个if语句称为if语句的嵌套。通过if语句的嵌套可以实现多分支选择结构，if语句的嵌套有下面两种形式。

阶梯if...else

```
if(表达式1)   语句1

else if(表达式2) 语句2

else if ( 表达式3 ) 语句3

else 语句4
```

执行流程是从上到下逐个对条件表达式进行判断，遇到某个条件表达式为真时，执行与之对应的语句，跳出整个选择判断结构。如果没有任何一个条件满足，则执行最后一个else后的语句。

循环结构

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

如果现在让你输入输出多组数据，如何处理？

循环语句的类别

while语句

```
while（表达式）
{
    语句体
}
    语句2

 while(scanf("%d%d",&a,&b)==2)
 {
     printf("%d",a+b);
 }
```

do while语句

```
do{
    语句
}
while（表达式）;
```

while语句和do while语句的差别

for语句

for（表达式1；表达式2；表达式3）

for（循环变量赋初值；循环条件；循环变量值更新）

```
int i,sum=0;
for(i=1;i<=10;i++)
{
    sum=sum+i;
}
```

---

break语句

作用：从循环体内跳出循环体，提前结束循环

```
while（表达式1）
{
语句1
if（表达式2） break;
语句2
}
语句3
```

注意break语句只能跳出一重循环

continue语句

结束本次循环，即跳过continue语句之后的其他语句，接着进行下一次是否执行循环的判定。

```
while（表达式1）
{
  语句1
  if（表达式2） continue;
  语句2
}
```

break语句和continue语句区别

acm题目特点

acm题目的输入数据和输出数据一般有多组（不定），并且格式多种多样，所以，要学会处理题目的输入输出。

推荐练习题目：杭电oj第11页题目（在百度上输入杭电）

基本输入输出

输入第一类：输入不说明有多少个input 模块，以EOF作为结束标志

EOF:End Of File 的缩写

参见杭电oj—1089（题号）

```
#include<stdio.h>
int main()
{
    int a.b;
    while(scanf("%d%d",&a,&b)!=EOF)
    printf("%d\n"，a+b);
}
```

说明：Scanf函数返回值就是读出的变量个数，如scanf("%d%d",&a,&b)

如果只有一个整数输入，返回值为1，如果有两个整数输入，返回值是2，如果一个都没有，则返回值是-1；

EOF是一个预定义的常量，等于-1.

输入第二类

输入一开始就会说有N个input Block

参见杭电oj—1090

```
#include<stdio.h>
int main()
{
    int n,i,a,b;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d%d",&a,&b);
        printf("%d",a+b);
    }
}
```

输入第三类

输入不说明有多少个input Block，但以某个特殊输入为结束标志。

参见杭电oj—1091

```
while(scanf("%d",&n)&&n!=0)
{
.......
}
```

输入第四类

输入是一整行字符串的

参见杭电oj—1048

```
char buf[];
gets(buf);
```

gets函数说明，从标准输入设备上读字符串函数，可以无限读取，不会判断上限，以回车键结束读取。

输出第一类

一个input Block 对应一个Output Block，Output Block之间没有空行

```
{
.....
printf("%d\n"，a+b);
}
```

输出第二类

一个input Block 对应一个Output Block,每个Output Block后都有空行。

```
{
......
printf("%d\n\n",a+b);
}
```

输出第三类

一个input Block 对应一个Output Block,Output Block之间都有空行。

```
    for(k=0;k<count;k++)
    {
        while(...)
        {
            printf("%d\n",a+b);
        }
        if(k!=count-1) printf("\n");
    }
```