

if/else if/else结构：选择结构中的典型方法，可以实现任意条件的多选一，执行顺序如下：

如果if条件为真，则执行if大括号中的内容。

如果if条件为假，进入第一个else if，如果这个else if的条件为真，则执行这个else if大括号中的内容。

如果这个else if的条件为假，则寻找下一个else if。重复此过程直到没有else if为止。

如果所有的else if都不满足，则执行else的内容。

switch/case结构：选择结构的第二个成员，switch括号中可以放一个整型变量（可以是整型、字符型、枚举，总之必须是某种整型），case后面对switch括号中的值进行筛选，等于哪个值就走哪个case分支。如果所有的case分支均不满足，则走default分支。

分支结束时必须使用break破坏掉整个结构，跳出switch大括号。否则会一直按顺序执行，直到大括号结束。

复杂逻辑的编写：先确立一定成立的逻辑，再确立有条件成立的逻辑，将所有有条件成立的逻辑和其前提条件用&&相连，最后将这些条件用||相连。

循环结构就是让计算机反复不停的执行同一件事。

while的结构如下：

```
while(表达式)
{
    语句;
}
```

表达式成立则语句一直反复执行，表达式不成立，则循环结束。

while循环的结构可以视为同样内容的if反复执行。

for的结构如下：

```
for(语句1;表达式;语句3)
{
    语句2;
}
```

语句1先执行，然后表达式成立则语句2语句3一直循环执行，表达式不成立，则循环结束。

for循环的结构非常适合执行遍历，因为可以把执行遍历的变量放在for这一行内，使得代码非常的有序。

数字遍历：`for(i = a; i <= b; i++)` 通过i遍历a~b之间的内容（包括a和b）

`for(i = a + 1; i < b; i++)` 通过i遍历a~b之间的内容（不包括a和b）

break：直接跳出当前循环

continue：中止本次循环，开始下次循环（for语句不会跳过语句3）

迭代法：通过变化规律和统计规律，可以很快的解决一类数学式求结果的问题。

例如：

```
for(i = 0; i < m; i++)  
{  
    tmp = tmp * 10 + n; //变化规律  
    sum += tmp;        //统计规律  
}
```

数位遍历：可以遍历一个数字在m进制下的每一位

遍历头： `for(i = n; i; i /= m)`

`i % m` 可以遍历数字n在m进制下的每一位。

循环嵌套：外层负责让n变化，内层负责处理相应功能。