

一、嵌套if

if...else if...

```
if(表达式1)
{
    表达式1成立执行的语句;
}
else if(表达式2)
{
    表达式1不成立，表达式2成立的时候;
}
```

```
if(表达式1)
{
    if(表达式2)
    {
        表达式1和表达式2都成立的时候执行的代码块;
    }
    else
    {
        表达式1成立，表达式2不成立执行的代码块;
    }
}
else
{
    表达式1不成立的时候执行的代码块;
}
```

输入一个年份，判断是否是世纪闰年，能被400整除的是世纪闰年，能被4整除，且不被100整除的是普通闰年。

```
#include <stdio.h>
int main(int argc, const char *argv[])
{
    int year; //定义年份
    scanf("%d",&year); //从终端获取
    if ((year%4==0&&year%100!=0)||year%400==0) //判断是否是闰年
    {
        if (year%400==0) //在是闰年的基础上，判断是不是世纪闰年（能被400整除）
        {
            printf("世纪闰年");
        }
        else //不是世纪闰年
        {
            printf("普通闰年");
        }
    }
    else //不是闰年
    {
        printf("不是闰年");
    }
    return 0;
}
```

```
}
```

二、switch…case语句

```
int a;
switch (a)
{
    case 常量1:
        语句;
        break;
    case 常量2:
        语句;
        break;
```

```
case 常量n:
    语句;
break; //退出这一层case
default: //如果不关心除了上述case结果外的其他结果，可以不写
    语句;
```

```
}
```

****switch…case注意: ****

1) switch后面的括号里面，不可以定义变量，可以放变量以及变量的表达式。

****只能是整形和字符型****

2) case对应的是switch括号里面的变量可能出现的情况。

3) 不同的case后面不能有相同的值，会报错。

4) case的顺序不会影响程序运行的结果，

5) default的位置也不会影响程序运行的结果，switch…case的运算顺序是先找case，找不到对应的值才会去运行default。

6) 每个case后面可以不写break，不写就会发生case击穿。

```
``c
#include <stdio.h>
int main(int argc, const char *argv[])
{
    int score;
    scanf("%d",&score);
    int a = score/10;
    switch (a)
    {
        case 10: //利用case击穿
        case 9:
            printf("A");
            break;
        case 8:
            printf("B");
```

```

        break;
    case 7:
        printf("C");
        break;
    case 6:
        printf("D");
    case 5:
    case 4:
    case 3:
    case 2:
    case 1:
    case 0:
        printf("不及格\n");
    }
    return 0;
}

```

```

#include <stdio.h>
int main(int argc, const char *argv[])
{
    float a,b;
    char c;
    scanf("%c %f %f",&c,&a,&b);
    switch(c)
    {
        case '+':
            printf("%f\n",a+b);
            break;
        case '-':
            printf("%f\n",a-b);
            break;
        case '*':
            printf("%f\n",a*b);
            break;
        case '/':
            printf("%f\n",a/b);
        default:;
    }
    return 0;
}

```

```

#include <stdio.h>
int main(int argc, const char *argv[])
{
    int year,mon,day; //定义年月日
    scanf("%d %d %d",&year,&mon,&day); //终端输入
    switch (mon)
    {
        case(12):
            day+=30;
        case(11):
            day+=31;
        case(10):
            day+=30;
    }
}

```

```

    case(9):
        day+=31;
    case(8):
        day+=31;
    case(7):
        day+=30;
    case(6):
        day+=31;
    case(5):
        day+=30;
    case(4):
        day+=31;
    case(3):
        day+=28;
    case(2):
        day+=31;
    case(1):
        printf("%d\n",day);
    }
    if (((year%4==0&&year%100!=0) || year%400==0)&&mon>=3)
    {
        day = day+1; //如果是闰年，且月份大于等于3，天数再+1，2月29天
    }
    printf("%d",day);
    return 0;
}

```

三、while循环

```

int a = 0;
while(条件1)
{
    条件成立执行的代码块;
    改变循环变量的语句;
}

```

```

#include <stdio.h>
int main(int argc, const char *argv[])
{
    //要求输出5行helloworld，循环执行的次数只和循环变量初始值以及循环条件有关。
    int a;
    a = 3; //循环变量初始值
    while(a<8) //循环条件
    {
        printf("hello world\n"); //输出helloworld
        a++; //修改循环变量
    }
    return 0;
}

```

猴子吃桃问题

一个猴子第一天摘了若干个桃，当即就吃了一半数量的桃，没吃过瘾，又多吃了一个第二天，将剩下的桃又吃了一半数量，没吃过瘾，又多吃了一个，以后的每一天都吃一半数量多一个到第十天再想吃桃的时候，发现只剩一个桃了。问：猴子第一天摘了多少个桃。用while循环实现。把每天吃之前的桃子数量都打印出来 应该怎么做？

10: 1

9: (1+1) *2

```
#include <stdio.h>
int main(int argc, const char *argv[])
{
    int peach = 1;
    int i = 1;
    while(i<=9)
    {
        peach = (peach+1)*2; //计算前一天的桃子数
        i++;
    }
    printf("%d\n",peach);
    return 0;
}
```

输出 [3,100] 范围内所有的质数

```
#include <stdio.h>
int main(int argc, const char *argv[])
{
    int i =3,j=2;
    int count=1;
    while(i<=6) //从3开始遍历到100
    {
        //count = 1; //每次把count重置为1
        //j=2; //把j重置为2
        printf("%d %d\n",count,j);
        while (j<i) //遍历从2-i的每一个数
        {
            if (i%j==0) //如果能整除说明不是质数
            {
                count = 0; //不是质数count置为0
            }
            j++; //改变内层循环变量
        }
        if (count==1) //判断标志位
        {
            printf("%d是质数\n",i);
        }
        i++; //改变外层循环变量
    }
    return 0;
}
```

四、do…while循环

```
do
{
    条件成立执行的代码块；
}while(条件);
do...while循环，不管循环条件是否成立，都会执行一次。
```

用do...while来实现死循环

```
#include <stdio.h>
int main(int argc, const char *argv[])
{
    do
    {
        printf("1");
    }while(1); //条件永远为真
    return 0;
}
```

五、for循环

```
for(循环变量初始化;循环条件;修改循环变量)
{
    当循环条件满足的时候，执行的代码块；
}
```

```
#include <stdio.h>
int main(int argc, const char *argv[])
{
    for (int i =0;i<3;i--)
    {
        printf("hello world\n");
    }
    for (;1; )
    {
        printf("1");
    }
    return 0;
}
```