

分支和循环语句

★分支语句:

if嵌套语句:

```
//if 判断语句
#include <stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    if (n == 0) {
        printf("this number is zero");
    }
    else {
        if (n > 0) {
            printf("this number is Higher than zero");
        }
        else {
            printf("this number is Lower than zero");
        }
    }
    return 0;
}
```

输入输出:

0

this number is zero

-1

this number is Lower than zero

2

this number is Higher than zero

如果是单语句 不需要加花括号，如果是语句块，就要加上花括号，否则无法执行

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d",&n);
    if (n < 18)
        printf("未成年");
    else if (n < 40)
        printf("青壮年");
    else if (n < 89)
        printf("老年");
    else
        printf("老寿星");
    return 0;
}
```

```
}
```

输入输出:

89

老年

19

青壮年

13

未成年

悬空else问题

```
#include <stdio.h>
int main() {
    int a = 0;
    int b = 2;
    if (a == 1) {
        if (b == 2)
            printf("hehe\n");
        else
            printf("haha\n");
    }
    return 0;
}
```

```
#include <stdio.h>
int main() {
    int a = 0;
    int b = 2;
    if (a == 1) {
        if (b == 2)
            printf("hehe\n");
        else
            printf("haha\n");
    }
    return 0;
}
```

else跟他离的最近的if语句匹配，合理使用大括号来匹配if语句，代码的书写风格很重要！

switch语句

whitch后的expression必须是整形表达式

case后的值，必须是整形常量表达式

```
switch (expression)
{
    case value1: statement
    case value2: statement
    ...
    default: statement
}
```

```
//输入一个数，计算他的余数
#include <stdio.h>

int main() {
    int n = 0;
    scanf("%d",&n);
    switch (n % 3)
    {
        case 0:
            printf("余数为0");
            break;
        case 1:
            printf("余数为1");
            break;
        case 2:
            printf("余数为2");
            break;
    }
    return 0;
}
```

case是它的入口
break是它的出口

输入1-5输出工作日，输入6和7输出休息日。

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d",&n);
    switch (n) {
        case 1:
        case 2:
        case 3:
        case 4:
        case 5:
            printf("工作日\n");
            break;
        case 6:
        case 7:
            printf("休息日\n");
            break;
        default:
            printf("输入错误! \n");
    }
    return 0;
}
```

输入输出结果：

3

工作日

7

休息日

12

输入错误！

★ 循环语句

for循环语句

语法形式：

```
for(表达式1 ; 表达式2 ; 表达式3)
    语句; //要包含更多语句的话，可以加上大括号
```

表达式1：用于循环变量的初始化

表达式2：用于循环结束条件的判断

表达式3：用于循环变量的调整

for循环的变种

```
//for循环的三部分可以省略不写，语法是支持的
//判断部分被省略，判断就恒为真，就会造成死循环
#include <stdio.h>
```

```
int main() {
    for (;;)
    {
        printf("hehe");
    }
    return 0;
}
```

while循环语句

执行条件：0表示假，非0为真

```
//打印1~10的值
int main() {
    int n = 1;
    while (n <= 10) {
        printf("%d ", n);
        n++;
    }
    return 0;
}
```

do-while循环语句

do-while的循环体至少被循环一次

```
//输出1~10的值
#include <stdio.h>
int main() {
    int i = 1;
    do {
        printf("%d ", i++);
    }
    while (i <= 10);
    return 0;
}
```

break和continue语句

break的作用是用于用永久的终止循环，只要break被执行，直接就会跳出循环，继续往后执行。

continue的作用是跳过本次循环中continue后面的代码，在for循环和while循环中有所差异的

循环的嵌套

```
//打印100~200之间的素数
#include <stdio.h>
#include <math.h>

int main() {
    int i = 0;
    for (i = 101; i <= 200; i+=2) {        //优化代码，首先偶数不可能是素数，所以从101开始
        //且步长设置为2
        int flag = 1;
        int j = 0;
        for (j = 2; j <= sqrt(i); j++) {    //sqrt 开根号，如果根号下的数字被整除，后面的
            //也不用计算了
            if (i % j == 0) {
                flag = 0;
                break;
            }
        }
        if (flag == 1) {

```

```
        printf("%d ", i);  
    }  
}  
return 0;  
}
```

输出结果:

```
101 103 107 109 113 121 127 131 137 139 149 151 157 163 167 169 173 179 181 191  
193 197 199
```
