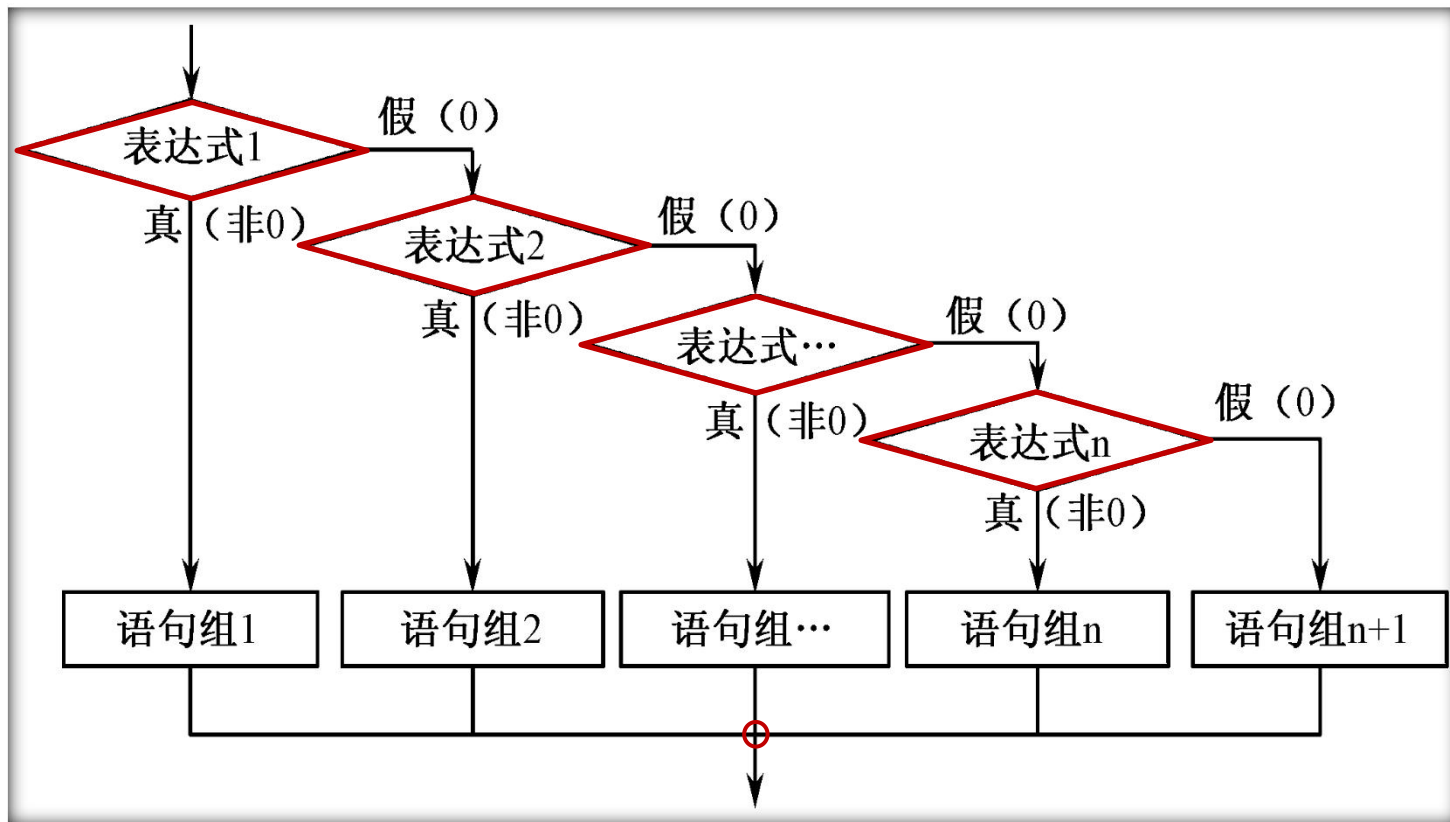


# SWITCH语句





多分支结构

# SWITCH语句用于多分支结构程序中

```
switch(表达式)
{
case 常量表达式1: {语句1; } break;
case 常量表达式2; {语句2; } break;
...;
case 常量表达式n; {语句n; } break;
default; {(语句n+1; }
}
```



# SWITCH语句用于多分支结构程序中

switch(表达式)

{

case 常量表达式1: {语句1; } break;

case 常量表达式2; {语句2; } break;

...;

case 常量表达式n; {语句n; } break;

default; {(语句n+1;}

}

注意标点符号



# SWITCH语句用于多分支结构程序中

switch(表达式)

{

case 常量表达式1: {语句1; } break;

case 常量表达式2; {语句2; } break;

...;

case 常量表达式n; {语句n; } break;

default; {(语句n+1;}

}

语句先进行表达式的运算，  
当表达式的值与某一case后  
面的常量表达式相等，就执  
行它后面的语句



# SWITCH语句用于多分支结构程序中

switch(表达式)

{

case 常量表达式1: {语句1; } break;

case 常量表达式2; {语句2; } break;

...;

case 常量表达式n; {语句n; } break;

default; {(语句n+1;}

}

每个CASE后面的表达式的值都是不同的



# SWITCH语句用于多分支结构程序中

```
switch(表达式)
{
  case 常量表达式1: {语句1; } break;
  case 常量表达式2; {语句2; } break;
  ...;
  case 常量表达式n; {语句n; } break;
  default; {(语句n+1;)}
}
```

在switch-case语句中，多个case可以共用一条执行语句，如：

```
case 'A':
case 'B':
case 'c':
    printf(">60\n"); break;
```



# SWITCH语句用于多分支结构程序中

```
switch(表达式)
{
case 常量表达式1: {语句1; } break;
case 常量表达式2; {语句2; } break;
    ...;
case 常量表达式n; {语句n; } break;
default; {(语句n+1;)}
}
```

当case语句后有break语句时，执行完这一case语句后，跳出switch语句，当case后面无break语句，程序将执行下一条case语句。





# SWITCH语句用于多分支结构程序中

switch(表达式)

{

case 常量表达式1: {语句1; } break;

case 常量表达式2; {语句2; } break;

...;

case 常量表达式n; {语句n; } break;

default; {(语句n+1;}

}

如果case中常量表达式值和表达式的值都不匹配，就执行default后面的语句。如果无default语句就退出switch语句。  
Default放在最后，也可无此语句。



# 例：

```
#include <stdio.h>
int main()
{
    int a;
    printf("Input integer number:");
    scanf("%d",&a);
    switch(a)
    {
        case 1: printf("Monday\n"); break;
        case 2: printf("Tuesday\n"); break;
        case 3: printf("Wednesday\n"); break;
        case 4: printf("Thursday\n"); break;
        case 5: printf("Friday\n"); break;
        case 6: printf("Saturday\n"); break;
        case 7: printf("Sunday\n"); break;
        default: printf("error\n"); break;
    }
    return 0;
}
```