



知识点多总结





- **switch**语句是C语言中用于多分支选择的一种控制结构。
- 语法格式：

```
switch(表达式A) { // 括号里的表达式可以是具体值、变量或表达式
    case 常量表达式1: // 可以是具体的常量值，也可以是常量表达式
        语句 1;
        ...
        break; // 可选
    case 常量表达式2:
        语句 2;
        ...
        break;
    // 可以有多个 case 分支
    default: // 可选
        语句 n;
        ...
}
```

- 程序会将switch后面括号内表达式A的值，与各个case后面的常量表达式的值进行比较。
- 如果找到匹配的case，就执行该case开始往后的代码，直到遇到break语句或switch语句的末尾。
- 如果没有匹配的case，且存在default分支，则执行default后的代码。



·注意：

- 1、switch后面表达式的值必须是**整数类型**（如：int、short、long等），或者是可以被隐式转换为整数的类型（如：char），不能是浮点数类型（如：float、double）。
- 2、case后面的值必须是**常量**，不能是变量。而且，在同一个switch里，case后面常量表达式的值必须是唯一的。
- 3、break语句是可选的，它是用来跳出当前的switch语句的。如果匹配的分支里没有break，程序就会继续执行下一个分支里的代码，这是“**穿透**”现象。程序会一直执行后续的分支，直到遇到break或switch语句的末尾，才会跳出switch。
- 4、default分支是**可选的**，表示所有case都不匹配时要执行的代码。default分支在switch语句中的位置可以是任意的，如果default分支内没有break语句，且其下方还有其他case分支，同样会出现“**穿透**”现象。



```
int x = 1;
switch(x) {
    case 1: // x = 1, 匹配到该分支
        语句 1;
        ...
    ... // 没有break, 继续执行下一个分支
    case 2:
        语句 2;
        ...
    ...
    break; // 遇到break, 跳出switch
    case 3:
        语句 3;
        ...
    ...
    break;
    default:
        语句 n;
        ...
}
```

```
int x = 1;
switch(x) {
    case 1: // x = 1, 匹配到该分支
        语句 1;
        ...
    ... // 没有break, 继续执行下一个分支
    case 2:
        语句 2;
        ...
    ... // 没有break, 继续执行下一个分支
    case 3:
        语句 3;
        ...
    ... // 没有break, 继续执行下一个分支
    default:
        语句 n;
        ...
    ... // 执行到switch语句末尾, 跳出switch
}
```



4、default分支是可选的，表示所有case都不匹配时要执行的代码。
default分支在switch语句中的位置可以是任意的，如果default分支内没有break语句，且其下方还有其他case分支，同样会出现“穿透”现象。

```
int x = 1;  
switch(x) {  
    default:  
        语句 n;  
        ...  
        break;  
    case 1:  
        语句 1;  
        ...  
        break;  
    case 2:  
        语句 2;  
        ...  
        break;  
    case 3:  
        语句 3;  
        ...  
}
```

```
int x = 1;  
switch(x) {  
    case 1:  
        语句 1;  
        ...  
        break;  
    case 2:  
        语句 2;  
        ...  
        break;  
    default:  
        语句 n;  
        ...  
        break;  
    case 3:  
        语句 3;  
        ...  
}
```



- 嵌套switch语句是指在一个switch语句的某个分支里又包含了一个switch语句。

- 程序会先根据外层switch表达式的值，找到匹配的case分支。

- 若该case分支中包含内层switch，则根据内层switch表达式的值，再在内层switch中找到匹配的case分支并执行相应的代码。

- 内层switch执行完后，会回到外层对应的case分支，继续执行后续的代码，直到遇到break或到达外层switch的末尾，才会跳出外层switch。

```
// 外层switch
switch(表达式A) {
    case 常量表达式1:
        // 内层switch
        switch(表达式B) {
            case 常量表达式3:
                语句1;
                break; // 跳出内层switch
            case 常量表达式4:
                ...
                break;
            ...
        }
        break; // 跳出外层switch
    case 常量表达式2:
        ...
        break;
    default:
        ...
}
```



·注意：

- 1、内层switch里的break语句只能跳出内层的switch，不会影响外层switch的执行。
- 2、default分支里也是可以嵌套switch语句的。



题目1、若有定义：

```
float x = 1.5;
```

```
int a = 1, b = 3, c = 2;
```

则正确的switch语句是 (D)

A.

```
switch(a + b) {  
    case 1:  
        printf("*\n");  
    case c:  
        printf("**\n");  
}
```

B.

```
switch((int)x); {  
    case 1:  
        printf("*\n");  
    case 2:  
        printf("**\n");  
}
```

D.

```
switch(a + b) {  
    case 1:  
        printf("*\n");  
    case 2 + 1:  
        printf("**\n");  
}
```

C.

```
switch(x) {  
    case 1.0:  
        printf("*\n");  
    case 2.0:  
        printf("**\n");  
}
```



题目2、若以下选项中的变量全部为整型变量，且已正确定义并赋值，则语法正确的 switch语句是 (D)

A.
switch(a + 9) {
 case c1:
 y = a - b;
 case c2:
 y = a + b;
}

B.
switch a * b {
 case 10:
 x = a + b;
 default:
 y = a - b;
}

C.
switch(a + b) {
 case1:
 case2:
 case3:
 y = a + b;
 break;
 case0:
 case4:
 y = a - b;
}

D.
switch(a * a + b * b) {
 default:
 break;
 case 3:
 y = a + b;
 break;
 case 2:
 y = a - b;
 break;



题目3、若有定义语句

```
int a, b;
```

```
double x;
```

则下列选项中没有错误的是 (C)

```
A.  
switch(x % 2) {  
    case 0:  
        a++;  
        break;  
    case 1:  
        b++;  
        break;  
    default:  
        a++;  
        b++;  
}
```

B.

```
switch((int)x / 2.0) {  
    case 0:  
        a++;  
        break;  
    case 1:  
        b++;  
        break;  
    default:  
        a++;  
        b++;  
}
```

C.

```
switch((int)x % 2) {  
    case 0:  
        a++;  
        break;  
    case 1:  
        b++;  
        break;  
    default:  
        a++;  
        b++;  
}
```

D.

```
switch((int)(x) % 2) {  
    case 0.0:  
        a++;  
        break;  
    case 1.0:  
        b++;  
        break;  
    default:  
        a++;  
        b++;  
}
```



题目4、以下选项中与

```
if (a == 1)  
    a = b;
```

else

```
    a++;
```

功能不同的switch语句是 (B)

A.

```
switch(a) {  
    case 1:
```

```
        a = b;
```

```
    break;
```

```
default:
```

```
    a++;
```

```
}
```

B.

```
switch(a == 1)  
{
```

```
    case 0:
```

```
        a = b;
```

```
    break;
```

```
case 1:
```

```
    a++;
```

```
}
```

C.

```
switch(a) {  
    default:
```

```
        a++;
```

```
    break;
```

```
case 1:
```

```
    a = b;
```

```
}
```

D.

```
switch(a == 1)  
{
```

```
    case 1:
```

```
        a = b;
```

```
    break;
```

```
case 0:
```

```
    a++;
```

```
}
```



题目5、有以下程序：

```
#include <stdio.h>

main() {
    int x = 1, y = 0, a = 0, b = 0;
    switch(x) {
        case 1:
            switch(y) {
                case 0:
                    a++;
                    break;
                case 1:
                    b++;
                    break;
            }
        case 2:
            a++;
            b++;
            break;
        case 3:
            a++;
            b++;
            break;
    }
    printf("a=%d,b=%d\n", a, b);
}
```

程序的运行结果是 (D)

- A. a=1,b=0
- B. a=2,b=2
- C. a=1,b=1
- D. a=2,b=1



题目6、有以下程序：

```
#include <stdio.h>

main( ) {
    char ch = 'A';
    switch(ch) {
        case 'A':
            printf("%c", 'A');
        case 'B' :
            printf("%c", 'B');
            break;
        default:
            printf("other");
    }
}
```

程序的运行结果为 AB。