

# C语言习题课



中国科学技术大学 教务处

University of Science and Technology of China

报告人：蒋欣玥

# 计算算术符优先级

1、计算  $(a++*1/3)$ ，假设  $a=2$ 。

注意  $++a$  和  $a++$  的区别。

$++a$  是先对  $a$  进行加1的操作，再让  $a$  参与后面的计算。而  $a++$  是先让  $a$  参与后面的计算，再进行+1的操作。 $a$  的初值是2，先参与后面的计算， $2*1$  等于2。 $2/3=0$ 。此时再对  $a$  进行+1的操作，此时  $a=3$ 。

答案：0

## 计算算术符优先级

2、计算 $(\text{float})(a+b)/2+(\text{int})x\%(\text{int})y$ ，假设 $a=2$ ， $b=3$ ， $x=3.5$ ， $y=4.6$

浮点数与整型做除法，结果为浮点型。

(注：整型与整型做除法，结果仍为整型，例如 $5/2=2$ )

答案： $2.5+3\%4=5.5$

## 计算算术符优先级

3、计算  $!(a+b)+c-1\&\&b+c/2$ ，假设  $a=2$ ， $b=3$ ， $c=4$

注意+的优先级大于&&

(注：负数是逻辑真，只要非0，都是真)

答案：  $3\&\&5=1$

补充：&与&&的区别。

&：二进制位与运算。例如  
 $2\&3$ 。

$10\&11=10$

&&：短路与。如果两个条件都成立则执行之后的逻辑。例如：  
 $\text{if}(a==0\&\&b==0)$ ，意思就是  
if a为0并且b为0的时候，进行  
下一步操作。

## 计算算术符优先级

4、下列程序的结果是

```
a=1,b=1;  
if(++a || ++b)  
printf("%d,%d",a,b);
```

||具有短路的功能，即如果第一个表达式为True，则不再计算第二个表达式

&&具有短路的功能，即如果第一个表达式为False，则不再计算第二个表达式

答案：2, 1

## 计算算术符优先级

5、计算 $a*=b\%c$ ，假设 $a=2$ ， $b=3$ ， $c=4$

% 优先于\*=

答案： $a*=3$ ， $a=2\times 3=6$

## 计算算术符优先级

6、表达式 $a+=a-=a=9$ 的值是

$a=9$

$a-=9$

$a=a-9=0$

$a+=0$

$a=a+0=0$

答案：0

## 计算算术符优先级

7、计算 $a=(a=++b, a+5, a/5)$ ，假设 $a=2, b=3$

逗号表达式从左到右，逐个计算，最右边的操作对象的值为整个逗号表达式的值

$a=++b;$

$a+5;$

$a=a/5$

答案：0



# 计算算术符优先级

## 8、下列程序的结果是

```
struct st
{
    int x, y;
} a[4]={10,20},{30,40},{50,60},{70,80}};
struct st *p=a;
printf("%d\n",p++->x);
printf("%d\n",++p->y);
```

注意->的优先级高于++

答案： 10, 41

## 9、读程序，写结果

```
main()
{
    int a[3][2] = { (0, 1), (2, 3), (4, 5) };
    int *p;
    p = a[0];
    printf("%d,%d,%d", p[0],p[1],p[2]);
}
```

注意逗号表达式的结果是取最右边的值，所以数组a的值为{1, 3, 5, 0, 0, 0}

答案：1,3,5

## 10、读程序，写结果

```
#define F(x) x-2
```

```
#define D(x) x • F(x)
```

```
printf("%d,%d",D(3),D(D(3)));
```

$D(3)=x \cdot x-2=9-2=7$

错误思路：  $D(3)=x \cdot (x-2)$

$D(D(3))=x \cdot x-2 \cdot x \cdot x-2-2=3 \cdot 3-2 \cdot 3 \cdot 3-2-2=9-18-4=-13$

把define当成单纯的替换的功能，替换时不要脑补括号

错误思路：  $D(D(3))= (x \cdot x-2) \cdot (x \cdot x-2-2)$

答案： 7,-13

## 11、读程序，写结果

```
char ch1 = 'a', ch2 = 'b';
```

```
    switch (ch1)
```

```
{
```

```
    case 'a':
```

```
        switch (ch2 )
```

```
{
```

```
        case 'a' :
```

```
            printf ( "aaa" );break;
```

```
        case 'b':
```

```
            printf ("bbb" );break;
```

```
        }
```

```
    case 'b':
```

```
        printf ( "ccc" ) ;break ;
```

```
}
```

case后未break，将继续执行后续语句

答案： bbbccc



12、 `int a[][3]={1,2,3,4,5}`, `y`, `*p=*a`。执行语句`y=(*++p)--`后, 数组`a`中各元素的值变为?

1 2 3

4 5 0

`++p`指向2, `2-1=1`

答案: 1, 1, 3, 4, 5, 0

## 13、读程序，写结果

```
#include <stdio.h>
void p(int *x,int y)
{
    --*x;
    y++;
    printf("%d,%d\n",*x,y);
}
int main ()
{
    int x=5,y=2;
    p(&x,y);
    printf("%d,%d\n",x,y);
    p(&y,x);
    printf("%d,%d\n",x,y);
    return 0;
}
```

(1)p(&x,y)                      x=5,y=2

    --\*x;                      x=5-1=4

    y++;                      y=y+1=3

(2)main                      x=4,y=2

x所在地址的内容已被修改为4。

y不变，因为形参改变不影响实参。

(3)p(&y,x)                      形参x=2,y=4

    --\*x;                      x=2-1=1

    y++;                      y=y+1=5

(4)main                      x=1,y=4

x不变，因为形参改变不影响实参。

y所在地址的内容已被修改为1。

答案： 4, 3

4, 2

1, 5

4, 1

14、下列程序中，while循环的执行次数是

main()

{

int i=0;

while(i<10)

{

if(i<1) continue;

if(i==5) break;

i++;

}

}

continue是结束本次循环，继续下一次循环

答案：无限循环

## 15、下列选项中错误的是

- A. 同一个数组中所有元素的类型相同
- B. 初始化时不可以跳过前面的数组元素给后面的元素赋初值定义语句
- C. `int a [10]={0};`给a数组中所有元素赋初值0
- D. 若有定义语句`int a[ 4]={1,2,3,4,5}`, 编译时将忽略多余的初值

数组溢出

答案：D



16、 int a[4]={0};下列表达式中与语法错误的是

- A. \*a
- B. \*(a+1)
- C. &a[3]
- D. ++a

数组名是地址，不能进行自增的操作

答案：D

## 17、下列选项中错误的转义字符是

- A. `'\n'`
- B. `'\x78'`
- C. `'\78'`
- D. `'\"'`

转义字符以\或者\x开头，以\开头表示后跟八进制形式的编码值，以\x开头表示后跟十六进制形式的编码值。对于转义字符来说，只能使用八进制或者十六进制。

八进制转义字符中不能出现比8大的数字。

答案：C

18、自定义函数void f(int x)，在main函数中有局部变量int y=1，且调用了语句f(y)，下列关于函数形参和实参的叙述中正确的是

- A. 形参x和实参y实际占同一存储单元
- B. 形参只是形式上的存在，并不占实际的存储单元
- C. 函数f被调用时，形参x和实参y分别占用不同的存储单元
- D. 如果将形参变量名也改为y，则形参x和实参y将占用同一存储单元

实参和形参各占用独立的存储单元

答案：C

## 形参与实参

### 形参（形式参数）

在函数定义中出现的参数可以看做是一个占位符，它没有数据，只能等到函数被调用时，接收传递进来的数据，所以称为形式参数，简称形参。

### 实参（实际参数）

函数被调用时给出的参数包含了实实在在的数据，会被函数内部的代码使用，所以称为实际参数，简称实参。

形参和实参的功能是传递数据，发生函数调用时，实参的值会传递给形参。

## 形参与实参

(1) 形参变量只有在函数被调用时才会分配内存，调用结束后，立刻释放内存，所以形参变量只有在函数内部有效，不能在函数外部使用。

(2) 实参可以是常量、变量、表达式、函数等，无论实参是何种类型的数据，在进行函数调用时，它们都必须有确定的值，以便把这些值传送给形参，所以应该提前用赋值、输入等方法使实参获得确定值。

(3) 实参和形参在数量上、类型上、顺序上必须严格一致，否则会发生“类型不匹配”的错误。当然，如果能够进行自动类型转换，或者进行了强制类型转换，那么实参类型也可以不同于形参类型。

(4) 函数调用中发生的数据传递是单向的，只能把实参的值传递给形参，而不能把形参的值反向地传递给实参；换句话说，一旦完成数据的传递，实参和形参就再也没有联系了，所以，在函数调用过程中，形参的值发生改变并不会影响实参。



# 感谢观看



中国科学技术大学 教务处  
University of Science and Technology of China

报告人：蒋欣玥