C语言：逻辑操作符

上期答案：

```c
int main()//Dear submission
{
    int i = 0;
    int count = 0;
    for (i = 100; i <= 200; i++)
    {
        //判断i是否是素数
        //如果是素数，就打印，不是素数就跳过
        //拿2~i-1之间的数字去挨个试除i，如果其中有一个数字整除了i，i就不是素数
        //如果所有的数字都不能整除i，i就是素数
        int j = 0;
        int flag = 1;//假设i是素数
        for (j = 2; j <= i - 1; j++)//2~8
        {
            if (i % j == 0)
            {
                flag = 0;//证明不是素数
                break;
            }
        }
        if (flag == 1)//是素数
        {
            printf("%d ", i);
            count++;
        }
    }

    printf("\ncount = %d\n", count);

    return 0;
}
```

逻辑取反操作符:!

例如有一个变量flag，如果flag为真，那么！flag就为假。如果flag为假，那么！flag就为真

还有a==b为a等于b，那么a！=b就是a不等于b。

与运算符:&&

&&是一个**双目操作符**，使用方式为a&&b。只有a，b都为真的时候整个表达式才为真，只要有一个为假，整个表达式都为假。

或运算符：||

||也是**双目操作符**，使用方式为a||b，||两边的表达式只要有一个为真，整个表达式都为真，||两边都为假的时候，整个才为假。

短路

在C语言的逻辑运算符中有一个特点，它总是先对左侧的表达式求值再对右边的表达式求值。

那么如果左边的表达式已经满足了逻辑运算符的条件，那就不再对右边的表达式求值。称为短路。

```
if(a >= 3 && a <= 5)
```

在此代码中如果a等于二，那么a>=就不成立，就不会再判断执行a<=5了。

```
if(a==1||a==2||a==3)
```

那么对于||。如果a等于1，a==1成立，也不会再执行判断后面的表达式了。

```
int main()
{
	int i = 0,a=0,b=2,c =3,d=4;
	i = a++ && ++b && d++;
	//i = a++||++b||d++;
	printf("a = %d\n b = %d\n c = %d\nd = %d\n", a, b, c, d);
	return 0;
}
```

根据上面的介绍，这段代码的运行结果是：

a = 1 b = 2 c = 3 d = 4

像这种仅仅根据左操作数的结果就能知道整个表达式的结果，不再对右操作数进行计算的运算称为**短路求值**。

下期为扫雷游戏凸^-^凸