

Программирование и алгоритмизация

Санкт-Петербургский государственный политехнический университет

11 сентября 2014

Бинарные

```
float x = -0.1, y = -0.05, z = 0.5;  
cout << (x<y<z) << endl;
```

Операции присваивания

Все эти операции изменяют значение операнда:

`++, --, +=, =, *=, /=, %=, <<=, >>=, &=, |=, ^=`

Операции присвоения

Чему будут равны значения переменных i, j в конце фрагмента?

```
int i=0, j=0;  
j = i++; //постинкремент  
i = ++j; //преинкремент
```

Результат префиксного инкремента lvalue, результат постфиксного инкремента — НЕ lvalue.

```
(x++)++;  
(++x)++; //без скобок ошибка  
++(++x);  
z=x+++y;  
z=x++++y; //???
```

Пример

```
y=3*(x=5 + (z=7)) + 1;
```

Совмещенные операторы присваивания

В C++ допускается совмещение арифметических операций одновременно с присваиванием. К примеру конструкции: `+=`, `*=`, `/=`.

```
a = a * b + c;  
a *= b += c;
```

В языках С и С++ существует две разновидности условных операторов или операторов разветвления: if-statement, и switch-statement

```
if (условие)
{
    операторы
}
else
{
    операторы
}
```

Для логических операций в языке С используются следующие символы:

- `||` — логическое ИЛИ (OR);
- `&&` — логическое И (AND);
- `!` — логическое НЕ (NOT);

```
if (min < n && n < max) // Если n лежит внутри
    диапазона
DoSomething();
```

Оператор switch

Алгоритм функционирования оператора можно описать в виде схемы:

```
switch ( выражение )
{
    case константа1: операторы ;
    case константа2: операторы ;
    .
    .
    .
    default: операторы ;
}
```

Оператор switch

После выполнения одной из ветвей case управление передается в следующую ветвь, поэтому для выхода из switch необходимо использовать один из следующих операторов:

- break;
- goto <метка>;
- continue;
- return;

Тернарные операции

Тернарная операция имеет следующий вид:

(условие) ? expr1 : expr2;

Тернарные операции

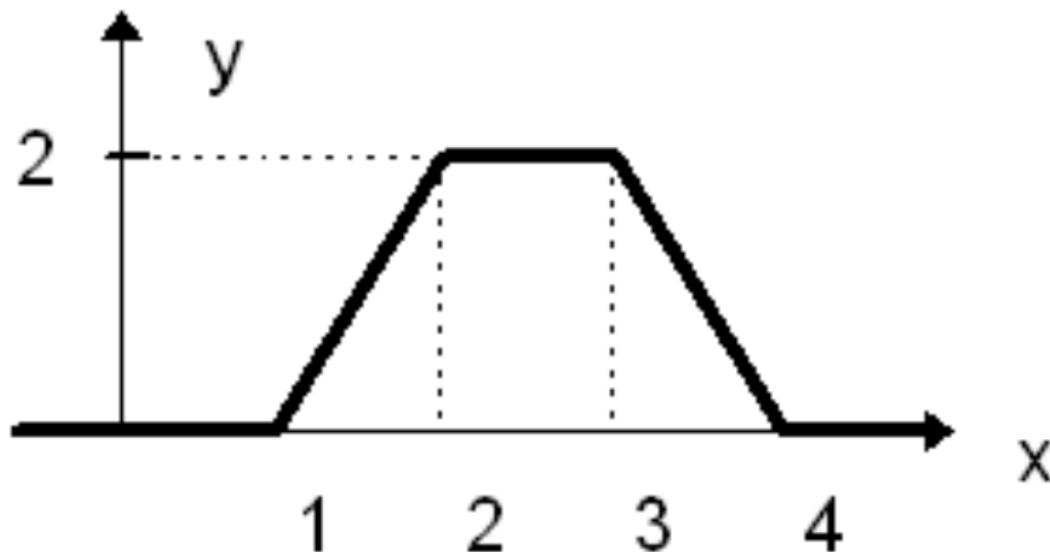
```
i = x < 0 ? -1 : 1;  
// Равносильно if (x<0) i=-1; else i=1;  
((a<=b) ? b : a) = max;  
//Равносильно if(a<=b) b = max; else a = max;
```

Тернарные операции

Специфика тернарной операции заключается в том, что она возвращает значение, т.е. она может быть rvalue (right-hand-side value), что позволяет создавать такие конструкции:

```
i = x < 0 ? -1 : x > 0 ? 1 : 0;
```

Пусть функция задана в виде графика.



Условные операторы

Чтобы задать ее с помощью цепочки вложенных условных операторов, определим переменные double x, y.

```
if (x>1)
    if (x>2)
        if (x>3)
            if (x>4)
                y=0;
            else
                y=-2*x+8;
            else
                y=2;
        else
            y=2*x-2;
    else
        y=0;
```

Условные операторы

Если результат логического выражения используется в арифметическом выражении, то вместо `false` компилятор подставит 0. А вместо `true` — единицу.

```
x = (z < 10) + 1;  
if(-1) {} //???  
int x, y;  
x = y = 0;  
if(x=0) {} //???  
if(x=y) {} //???  
if(x в диапазоне от 0 до 100) {}  
int x = -1;  
int y = !x; //???  
x=-1  
y=!(++x);
```

Сложные примеры

```
int x = -1, y = 1;  
if (x++ && y--) //???  
//x = ? y = ?  
x = -1, y = 1;  
if (++x && --y) //???  
//x = ? y = ?  
int x = 1, y = 2, z;  
z = (x<y) ? x++ : y++;  
//x = ? y = ?
```

Защита от деления на 0 с использованием тернарного оператора:

```
a = b ? 1/b : 0;
```

Оператор ,

```
int a = 1, b = 2, c = 3;  
x = a++, b++, c++;  
x = (a++, b++, c++);
```

Операторы циклов

Три оператора цикла языков С и С++ могут быть описаны следующей схемой:

```
while (условие) {тело цикла}
do {тело цикла} while (условие);
for ( ; ; ) {тело цикла}
```

Операторы циклов

Сколько раз выполнится каждое тело цикла?

```
while(0) {тело цикла}  
while (1) {тело цикла}  
do {тело цикла} while(0);
```

Операторы циклов

```
int i=1, j=1024;  
while (i < j) i=2*i;
```

```
int i=1, j=1024;  
while (i < j) j=2*i;
```

Цикл for

Последовательность выполнения операторов при реализации цикла:

- ① Операторы первой части заголовка (выполняются один раз до входа в цикл).
- ② Вторая часть (проверка условия и выход из цикла в случае его нарушения).
- ③ Операторы тела цикла.
- ④ Операторы третьей части заголовка.
- ⑤ Переход к пункту 2.

Цикл for

```
for (int i=0; i<1024; i++)
cout << i << endl; //Печать i-го значения
```

Задача

Исправить 1 символ чтобы программа напечатала 20 *. Именно исправить, а не удалить или добавить, пробел тоже символ.
Найти 3 решения.

```
int main () { int i , N=20; for ( i = 0 ; i < N ; i -- )  
{ printf ("*"); } }
```
