In C, if the condition is true, it is replaced by 1 and if the condition is false, it is replaced by 0. Conversely, any non-zero number (positive or negative) represents ‘true’ and zero is represented as ‘false’ value.

So assuming x, y, z are defined variables and

x = 3, y = 3;

z = x==y;

Then z stores the value 1 because the equality condition is true.

In C, data type of result of comparison operators as int. So ‘x’ should be defined as ‘int’ in C in above program. Whereas in C++, the result of comparison is of data type ‘bool’. Bool data type stores only 0 and 1. It is better to define ‘x’ as ‘bool’ in C++ than int.

Consider the following program:

```
#include <stdio.h>

int main()
{
        int a, b, c;
        a = b = c = 5;
        if (a == b == c)  printf(“I expected this!”);
        else    printf(“I didn’t expect this.”);
        return 0;
}
```

Output: I didn’t expect this.

Because first a == b is checked which is true and hence replaced by 1. Then 1 == c is checked which is false. So overall the result is false and hence the else part is executed.