Decision Making in C

Faculty: Dr. Asif Uddin Khan

Review of Operators

Relational Operators

To Specify	Symbol Used
less than	<
greater than	>
less than or equal to greater than or equal to	<= >=

Equality and Logical Operators

To Specify	Symbol Used
Equal to	==
Not equal to	!=
Logical AND	&&
Logical OR	||
Negation	!

Points to Note

- If an expression, involving the relational operator, is true, it is given a value of 1. If an expression is false, it is given a value of 0. Similarly, if a numeric expression is used as a test expression, any non-zero value (including negative) will be considered as true, while a zero value will be considered as false.

- Space can be given between operand and operator (relational or logical) but space is not allowed between any compound operator like <=, >=, ==, !=. It is also compiler error to reverse them.

- a == b and a = b are not similar, as == is a test for equality, a = b is an assignment operator. Therefore, the equality operator has to be used carefully.

- The relational operators have lower precedence than all arithmetic operators.

A Few Examples

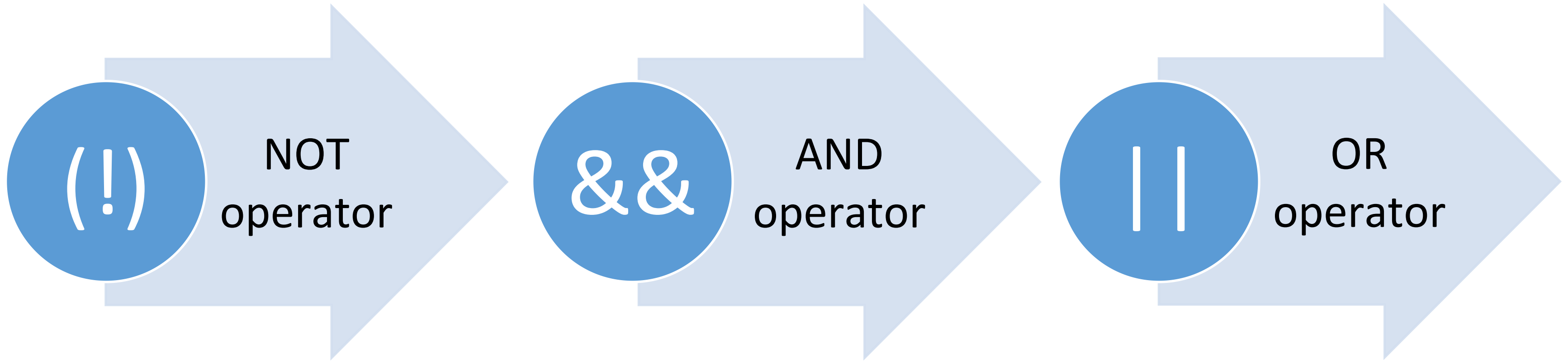
The following declarations and initializations are given:

```
int x=1, y=2, z=3;
```

Then,

- The expression x>=y evaluates to 0 (false).
- The expression x+y evaluates to 3 (true).
- The expression x=y evaluates to 2 (true).

Logical operators may be mixed within relational expressions but one must abide by their precedence rules which is as follows:



Operator Semantics

Operators	Associativity
() ++ (postfix) -- (postfix)	left to right
+ *(unary)* - *(unary)*	right to left
++ *(prefix)* -- *(prefix)* * / %	left to right
+ -	left to right
< <= > >=	left to right
== !=	left to right
&&	left to right
||	left to right
?:	right to left
= + = - = * = / =	right to left
, (comma operator)	left to right

Decision Making in C

- Deciding the order of execution of statements based on certain conditions.
- C language handles decision-making by supporting the following statements

  - ✓ if statement
  - ✓ switch statement
  - ✓ conditional operator statement (? : operator)
  - ✓ goto statement

if statements in C

- Used to perform the operations based on some specific condition
- The operations specified in the “if block” are executed if and only if the given condition is true

Syntax:

```
if(expression)
{
//if block code to be executed
}
```

Example:

```
int n=20;
if(n>18)
  {
Printf(“Eligible for Voting\n”);
  }
```

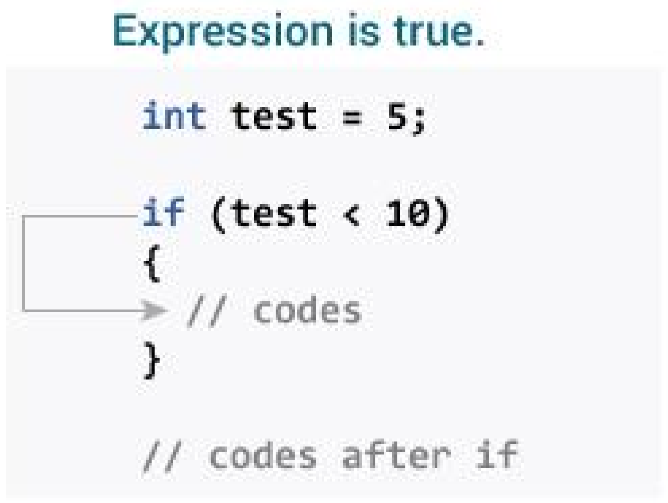
Variants of if statement

There are the following variants of if statement in C language.

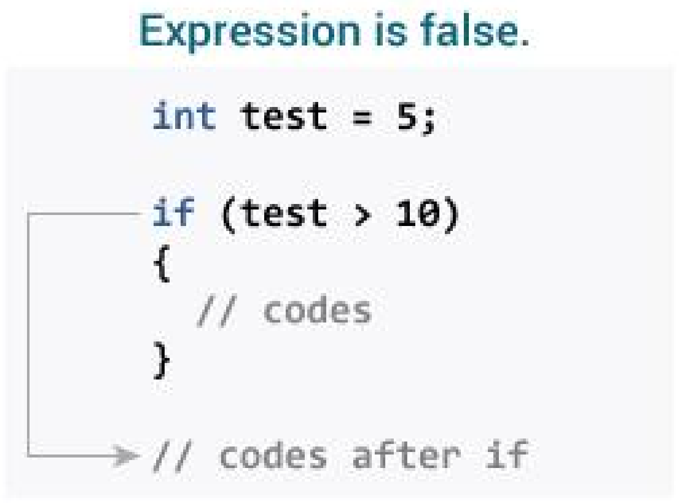
- if statement
- if…else statement
- if...else Ladder
- Nested if

if statement

- The if statement evaluates the test expression inside the parenthesis ().
- If the test expression is evaluated to true, statements inside the body of if are executed.
- If the test expression is evaluated to false, statements inside the body of if are not executed.



```
int test = 5;

if (test < 10)
{
  // codes
}

// codes after if
```

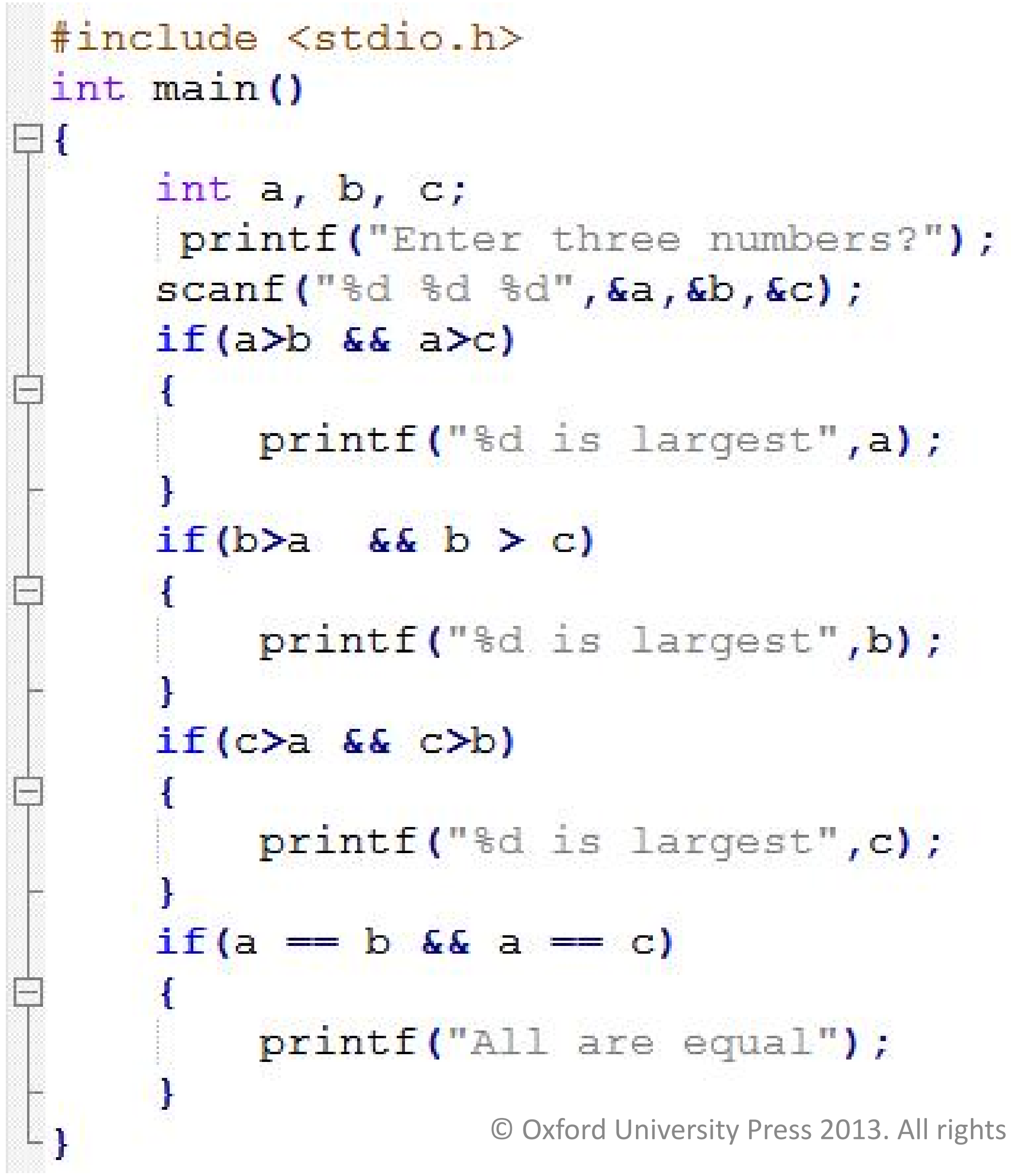


```
int test = 5;

if (test > 10)
{
  // codes
}

// codes after if
```

Example

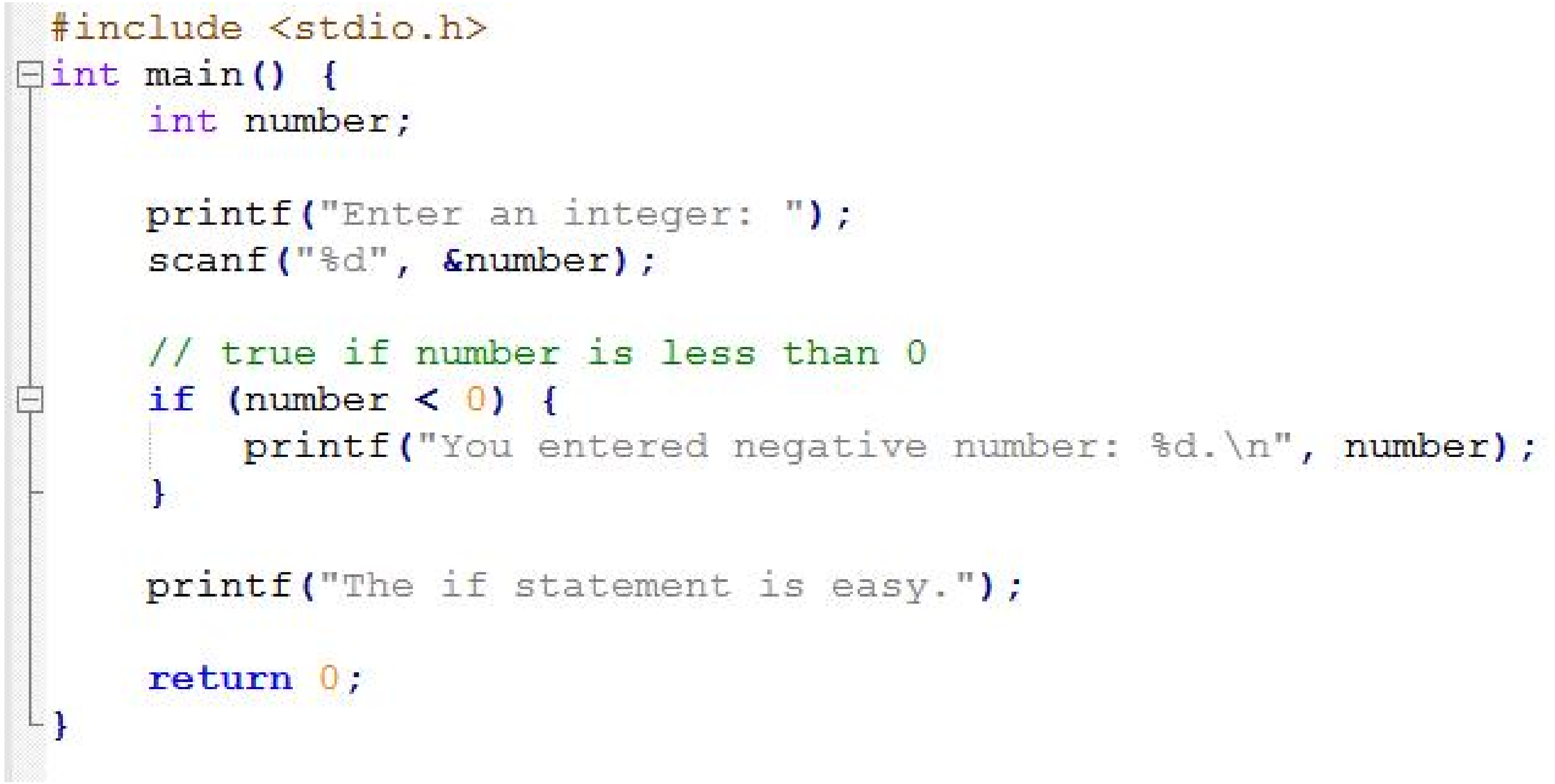
```
#include<stdio.h>
int main(){
int num;
printf("Enter a number:");
scanf("%d",&num);
if(num%2==0){
printf("%d is even number\n",num);
}
printf(“Welcome to KIIT\n”);
return 0;
}
```

Program to find the largest number of the three



```c
#include <stdio.h>
int main()
{
    int a, b, c;
     printf("Enter three numbers?");
    scanf("%d %d %d",&a,&b,&c);
    if(a>b && a>c)
    {
        printf("%d is largest",a);
    }
    if(b>a  && b > c)
    {
        printf("%d is largest",b);
    }
    if(c>a && c>b)
    {
        printf("%d is largest",c);
    }
    if(a == b && a == c)
    {
        printf("All are equal");
    }
}
```

Program to display a number if it is negative

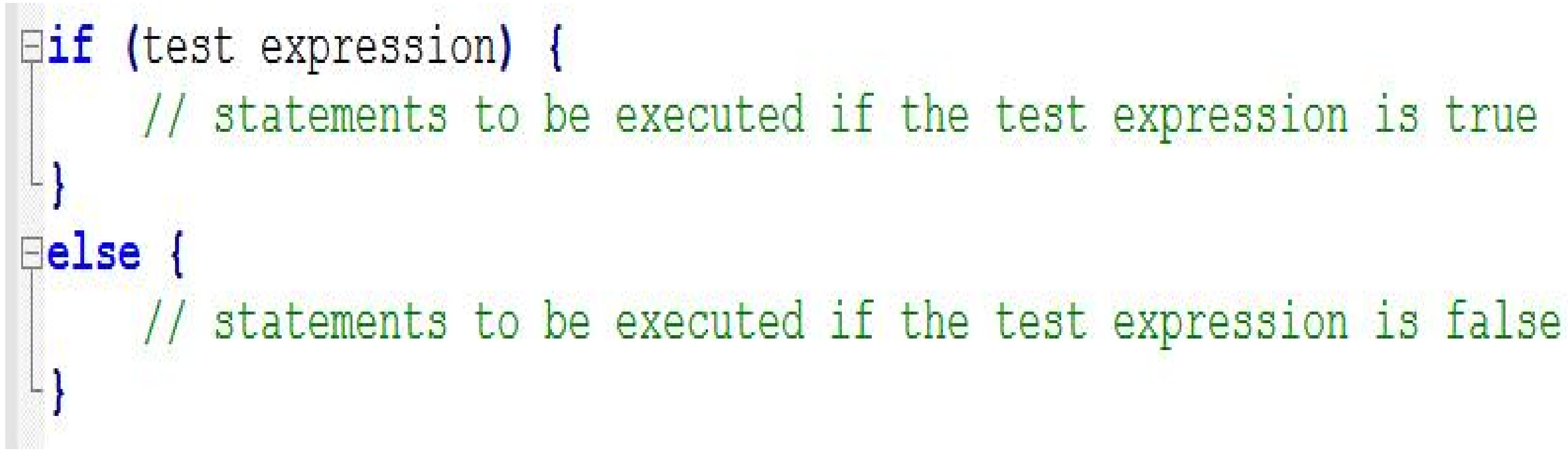


```c
#include <stdio.h>
int main() {
    int number;

    printf("Enter an integer: ");
    scanf("%d", &number);

    // true if number is less than 0
    if (number < 0) {
        printf("You entered negative number: %d.\n", number);
    }

    printf("The if statement is easy.");

    return 0;
}
```

C if...else Statement

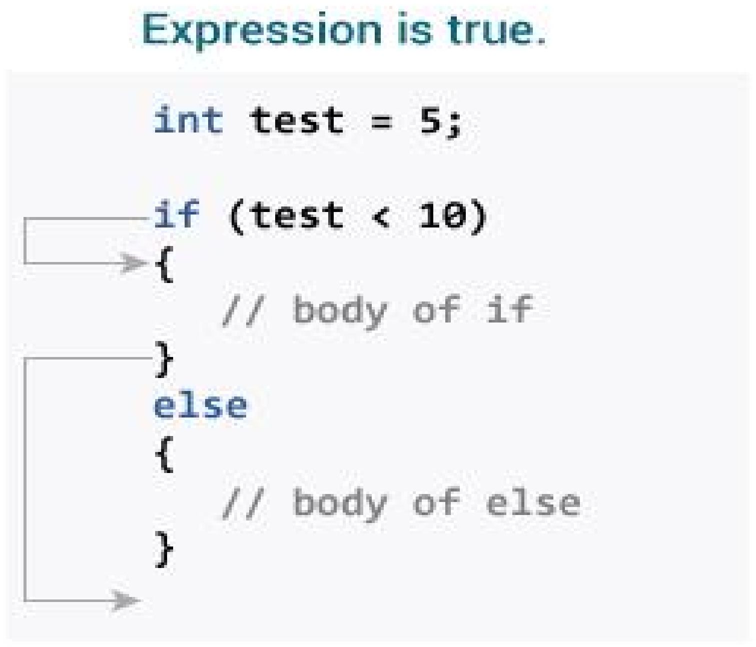
- The if statement may have an optional else block.

Syntax

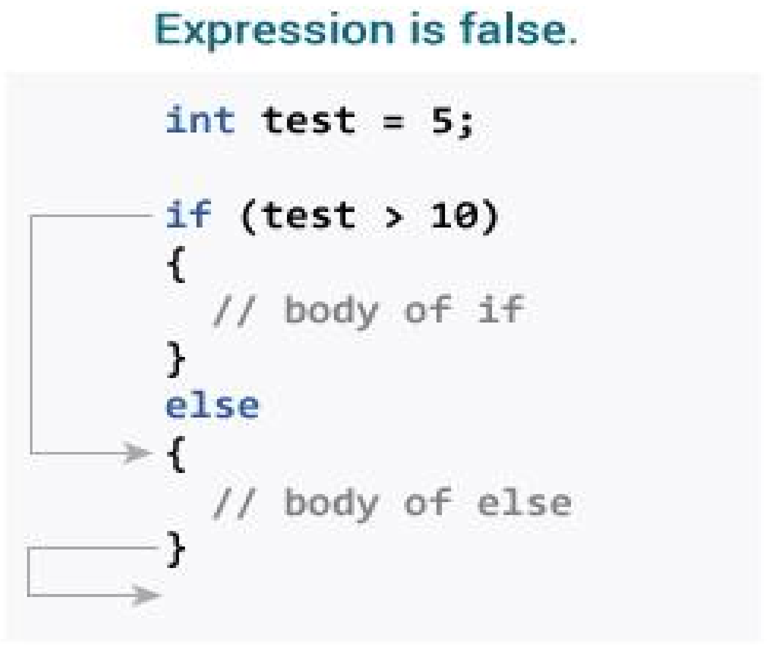


```
if (test expression) {
    // statements to be executed if the test expression is true
}
else {
    // statements to be executed if the test expression is false
}
```

C if...else Statement



```
int test = 5;

if (test < 10)
{
   // body of if
}
else
{
   // body of else
}
```

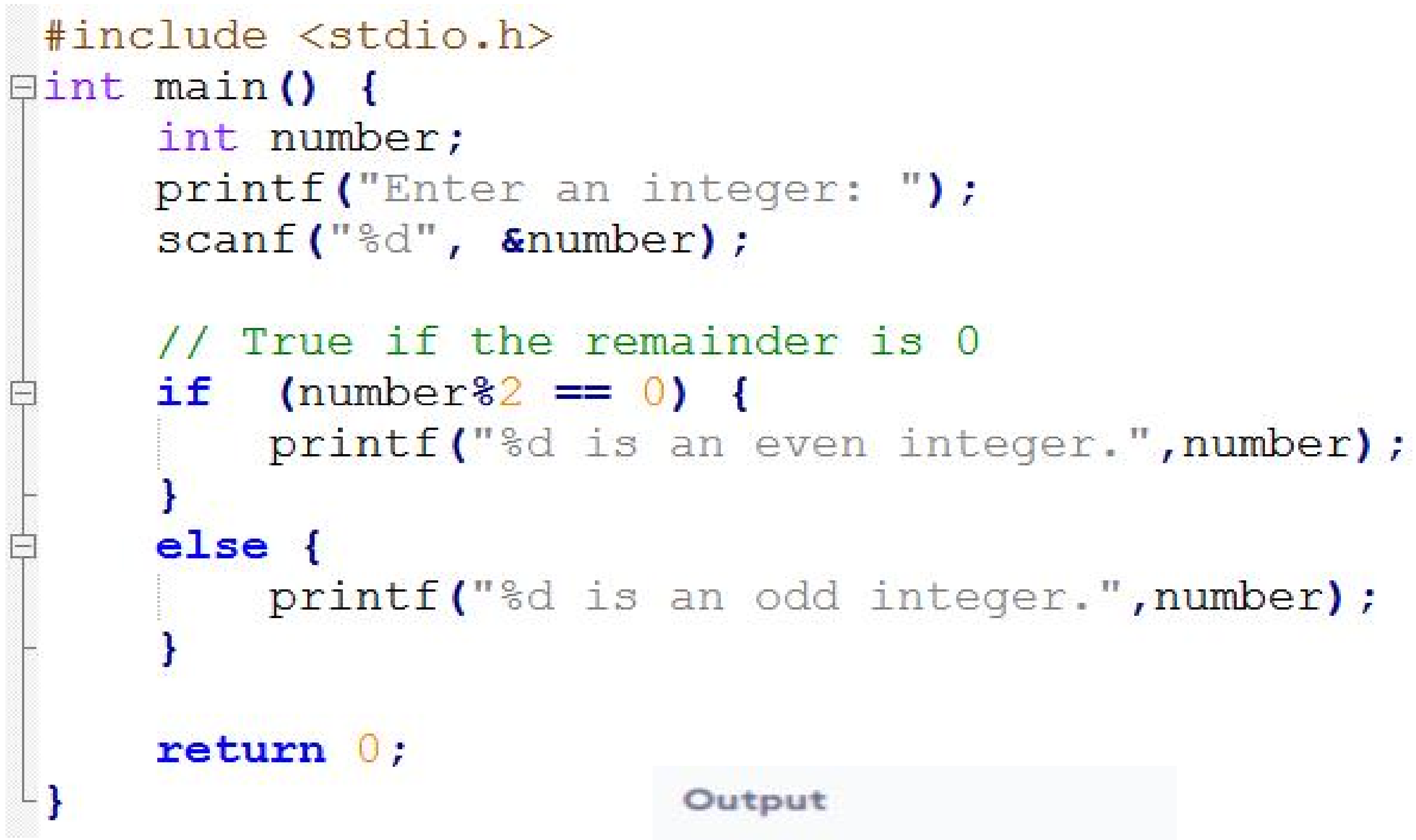


```
int test = 5;

if (test > 10)
{
   // body of if
}
else
{
   // body of else
}
```

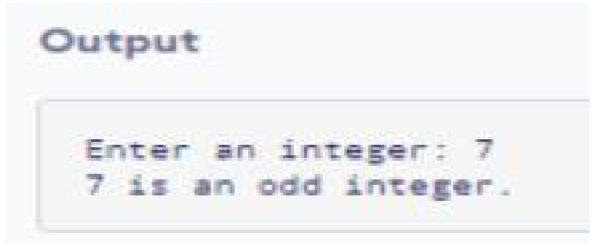
✓If the test expression is evaluated to true, statements inside the body of if are executed. Statements inside the body of else are skipped from execution.

✓If the test expression is evaluated to false, statements inside the body of else are executed. Statements inside the body of if are skipped from execution.

Example of if...else statement



```
#include <stdio.h>
int main() {
    int number;
    printf("Enter an integer: ");
    scanf("%d", &number);

    // True if the remainder is 0
    if  (number%2 == 0) {
        printf("%d is an even integer.",number);
    }
    else {
        printf("%d is an odd integer.",number);
    }

    return 0;
}
```



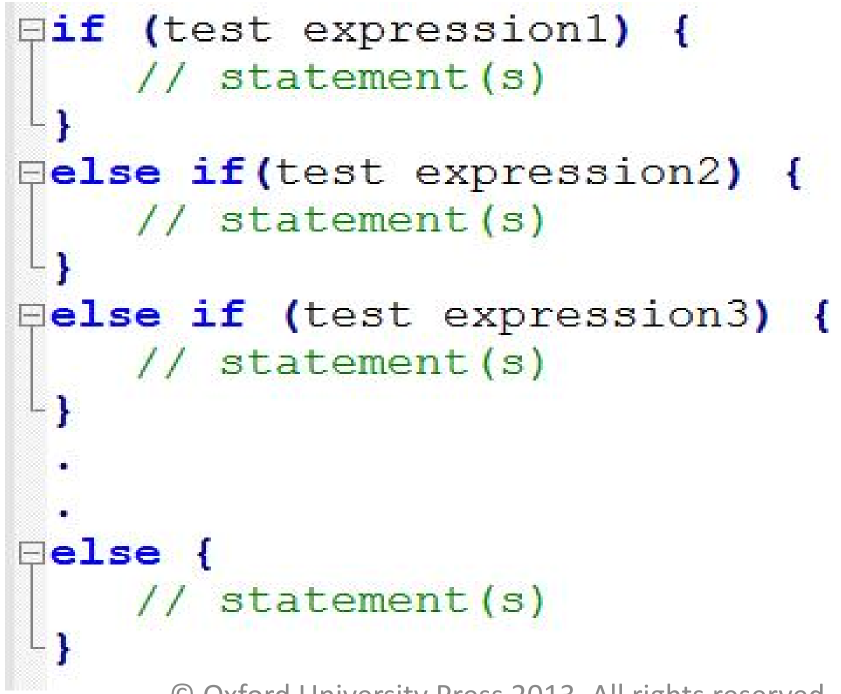
Output

```
Enter an integer: 7
7 is an odd integer.
```

if...else Ladder

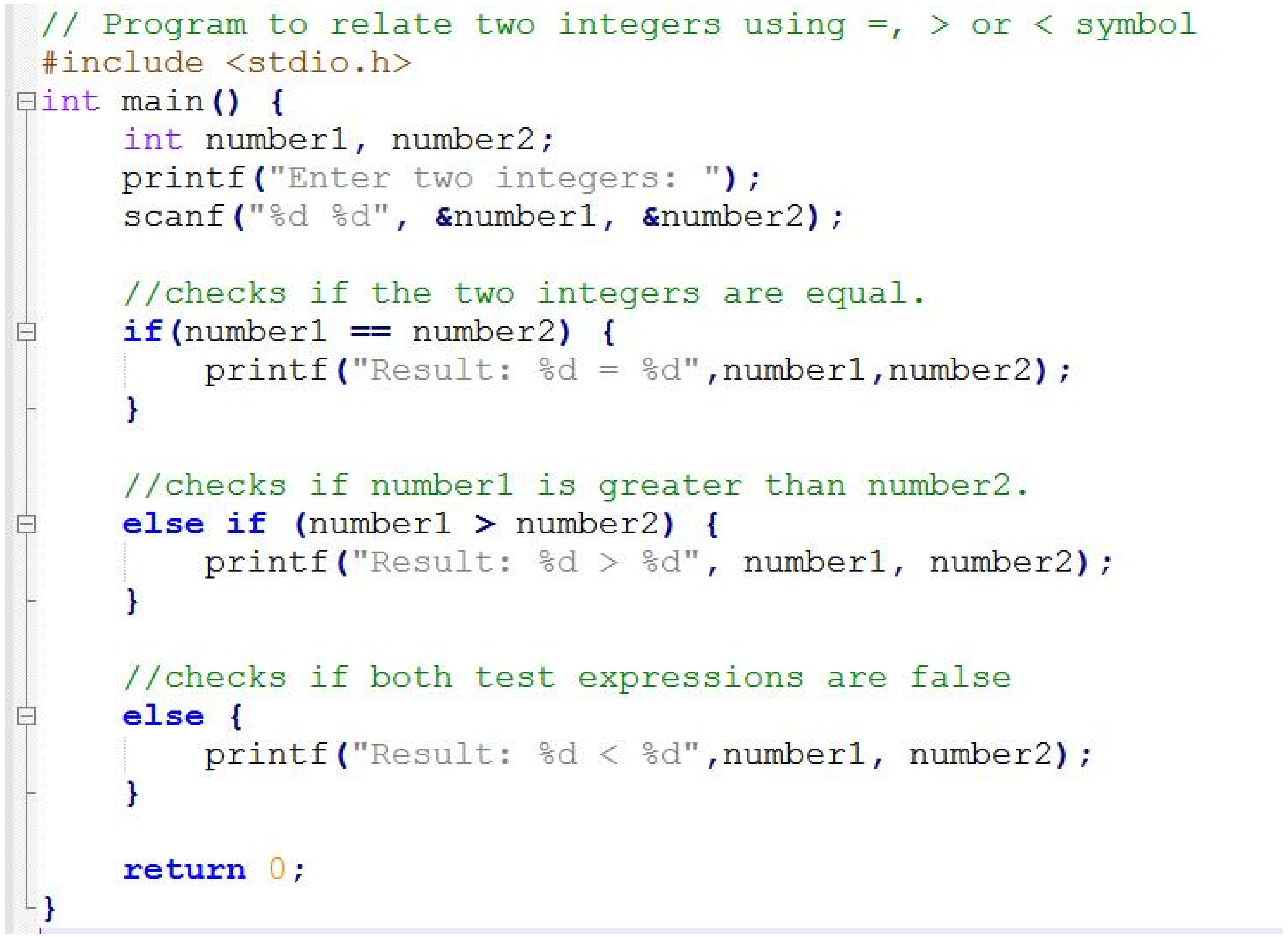
- Choice has to be made from more than 2 possibilities
- The if...else ladder allows you to check between multiple test expressions and execute different statements.

Syntax



```
if (test expression1) {
    // statement(s)
}
else if(test expression2) {
    // statement(s)
}
else if (test expression3) {
    // statement(s)
}
.
.
else {
    // statement(s)
}
```

Example of C if...else Ladder

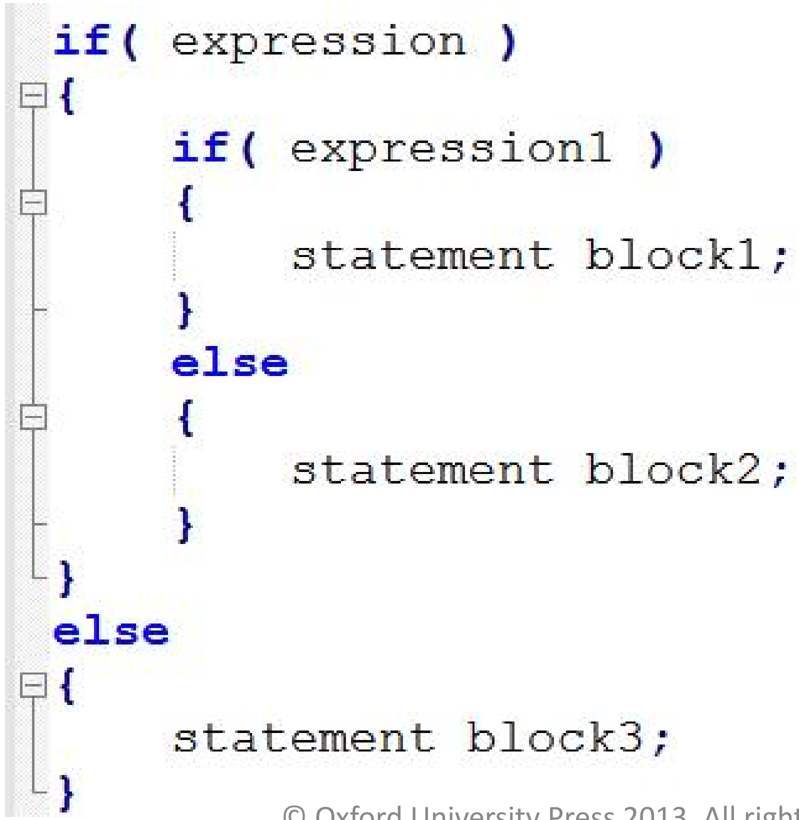


```
// Program to relate two integers using =, > or < symbol
#include <stdio.h>
int main() {
    int number1, number2;
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    //checks if the two integers are equal.
    if(number1 == number2) {
        printf("Result: %d = %d",number1,number2);
    }

    //checks if number1 is greater than number2.
    else if (number1 > number2) {
        printf("Result: %d > %d", number1, number2);
    }

    //checks if both test expressions are false
    else {
        printf("Result: %d < %d",number1, number2);
    }

    return 0;
}
```

Nested if...else

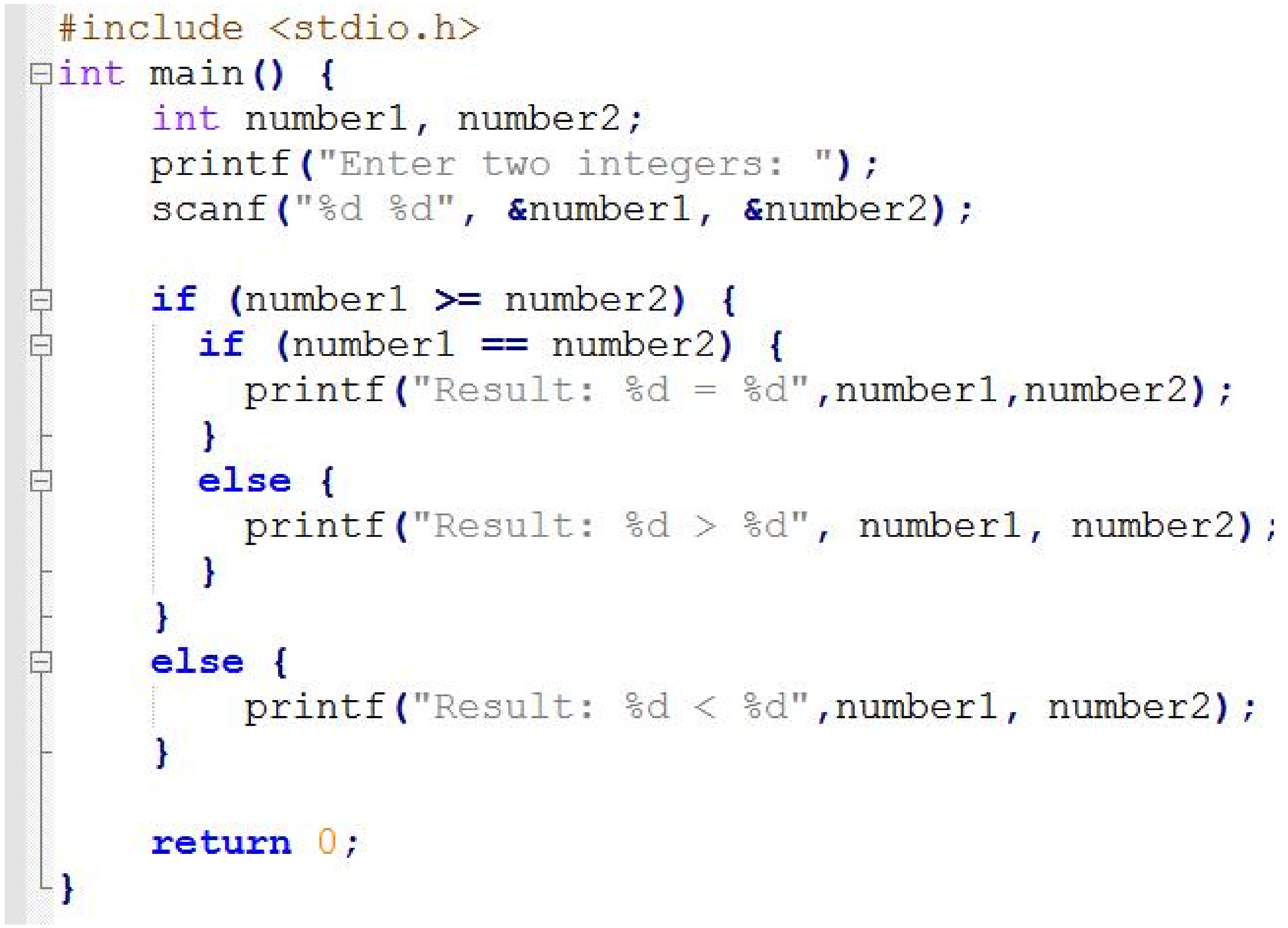
- It is possible to include an if...else statement inside the body of another if...else statement.

Syntax



```
if( expression )
{
    if( expression1 )
    {
        statement block1;
    }
    else
    {
        statement block2;
    }
}
else
{
    statement block3;
}
```

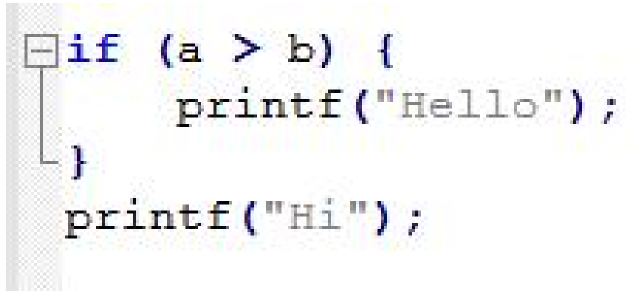
Example of Nested if...else



```c
#include <stdio.h>
int main() {
    int number1, number2;
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    if (number1 >= number2) {
      if (number1 == number2) {
        printf("Result: %d = %d",number1,number2);
      }
      else {
        printf("Result: %d > %d", number1, number2);
      }
    }
    else {
        printf("Result: %d < %d",number1, number2);
    }

    return 0;
}
```

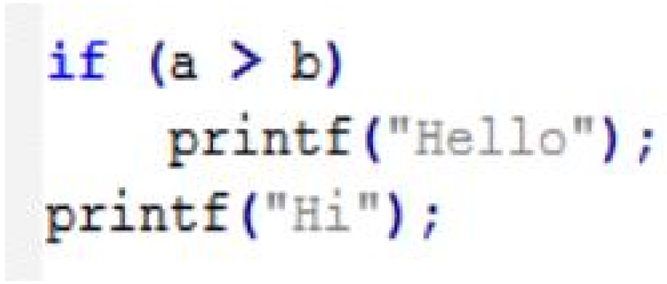
Important Note

- If the body of an if...else statement has only one statement, you do not need to use brackets {}



```
if (a > b) {
     printf("Hello");
}
printf("Hi");
```

Equivalent to



```
if (a > b)
     printf("Hello");
printf("Hi");
```

Write a program that prints the largest among three numbers.

Algorithm
1. START
2. PRINT “ENTER THREE NUMBERS”
3. INPUT A, B, C
4. MAX=A
5. IF B>MAX THEN MAX=B
6. IF C>MAX THEN MAX=C
7. PRINT “LARGEST NUMBER IS”, MAX
8. STOP

C Program

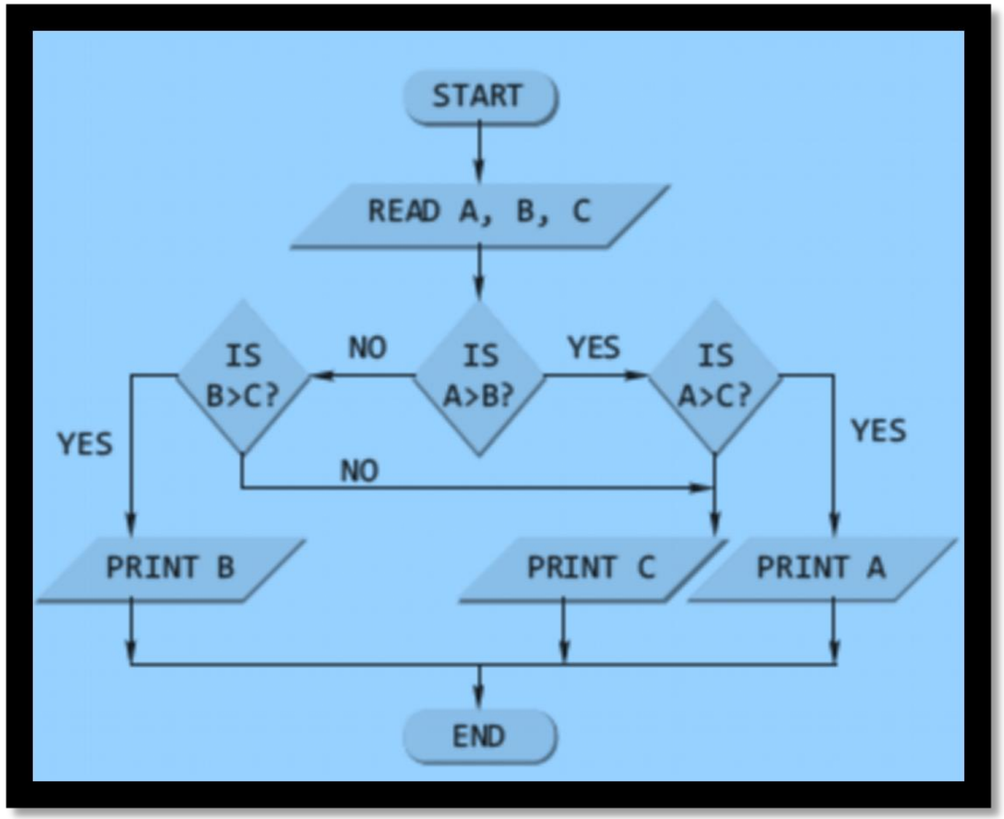
```
#include <stdio.h>
int main()
{
int a, b, c, max;
printf(“\nEnter 3 numbers”);
scanf(“%d %d %d”, &a, &b, &c);
max=a;
if(b>max)
{
  max=b;
}
if(c>max)
{
  max=c;
}
printf(“Largest No is %d”, max);
return 0;
}
```

The following program checks whether a number given by the user is zero, positive, or negative

```
#include <stdio.h>
int main()
{
   int x;
   printf(“\n ENTER THE NUMBER:”);
   scanf(“%d”, &x);
   if(x > 0)
                 {printf(“x is positive \n”)};
    else if(x == 0)
                 printf(“x is zero \n”);
    else
                  printf(“x is negative \n”);
     return 0;
}
```

```
#include <stdio.h>
int main()
{
  int a, b, c;
  printf("\nEnter the three numbers");
  scanf("%d %d %d", &a, &b, &c);
  if(a > b)
  {    if(a > c)
         printf("%d", a);
      else
          printf("%d", c);
}
 else
 {  if(b > c)
          printf("%d", b);
      else
          printf("%d", c);
}
  return 0;
}
```

A program to find the largest among three numbers using the nested if



also code for the condition a==b or b==c or a==c