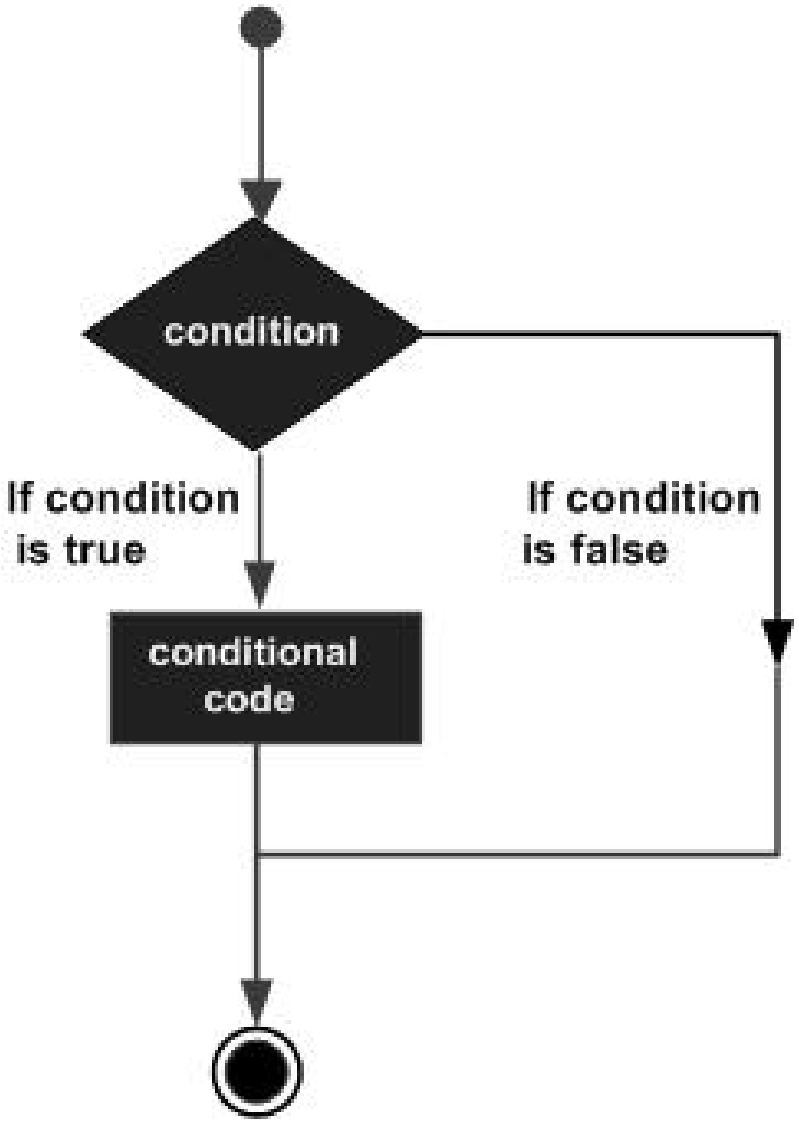
Flow control in C programming

Flow control structures require the programmer to determine one or more conditions to evaluate and test by the program, along with the commands to be executed if the condition is determined to be correct, or other commands are executed. if the condition is wrong.

Flow control structures require the programmer to determine one or more conditions to evaluate and test by the program, along with the commands to be executed if the condition is determined to be correct, or other commands are executed. if the condition is wrong.

Below is a common pattern of a common flow control structure in a programming language.



The C programming language assumes that all values **other than zero** or **not null** are **true** , if there are **zero or null** values, it is **false** .

Language C provides the following types of flow control structures.

Command Description
If statement

An **if statement** consists of a logical expression followed by one or more other commands.

If . else statement

An **if statement** can be followed by an **else statement** (optional: yes or no), which can be executed when the logical expression is false.

If command

You can use **if** or **else if statements** inside **if** or **else if else statements** .

Switch command

A **switch** command allows checking the condition of a variable before executing commands.

Command switch

You can use a **switch** command inside another **switch** command.

Conditional operator? : in C

We discussed **conditional operators?** : in the previous chapter that can be used to change the position for the **if . else statement** . It has the following general pattern:

```
bieuthuc1  ?  bieuthuc2  :  bieuthuc3 ;
```

In which Exp1, Exp2 and Exp3 are expressions. Notice the use and setting of the colon.

The value of the expression Exp1 before the sign? with **true** value, Exp2 is executed, and its value is the value of the expression. If Exp1 is **false** , Exp3 is executed and its value is the value of the expression.

If statement in C

An **if statement** in Language C contains a logical expression followed by one or more commands.

Syntax:

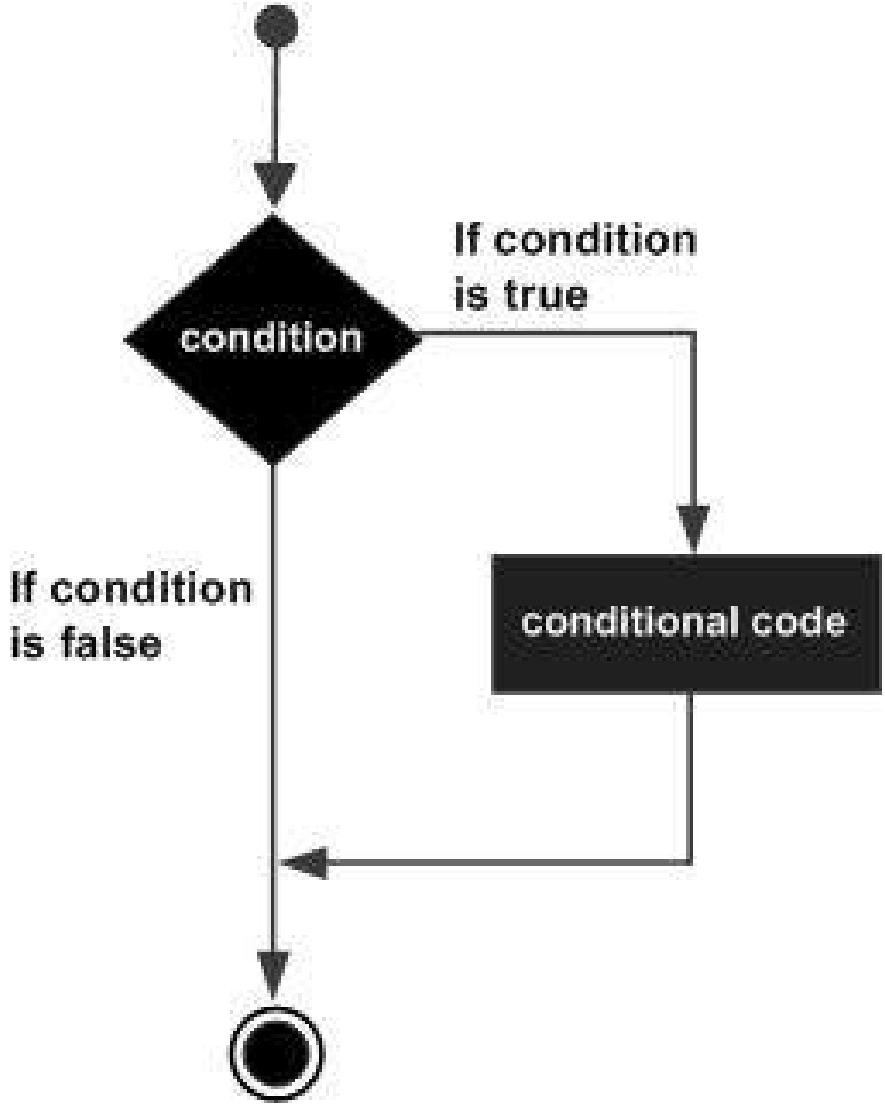
The following is the syntax of an if statement in Language C:

```
if ( bieu_thuc_boolean ) { /* cac lenh se duoc thuc thi neu bieu thuc boolean la
```

If the logical expression is evaluated as **true** , then the code block inside the if statement will be executed. If the logical expression is evaluated as **false** , then the command immediately after the if statement will be executed.

The C language assumes that any **non-zero** and **non-null values** are **true** , and if it is **zero** or **null** , then it is assumed to be false.

Diagram:



For example:

```
#include int  main  () { /* phan dinh nghia bien cuc bo */ int  a  = 15 ; /* ki
```

Compiling and executing the above C program will produce the following results:

```
a la nho hon 20
Gia tri cua a la: 15
```

If . else command in C

An **if statement** can be followed by an optional **else** command, which executes when the logical expression is false.

Syntax:

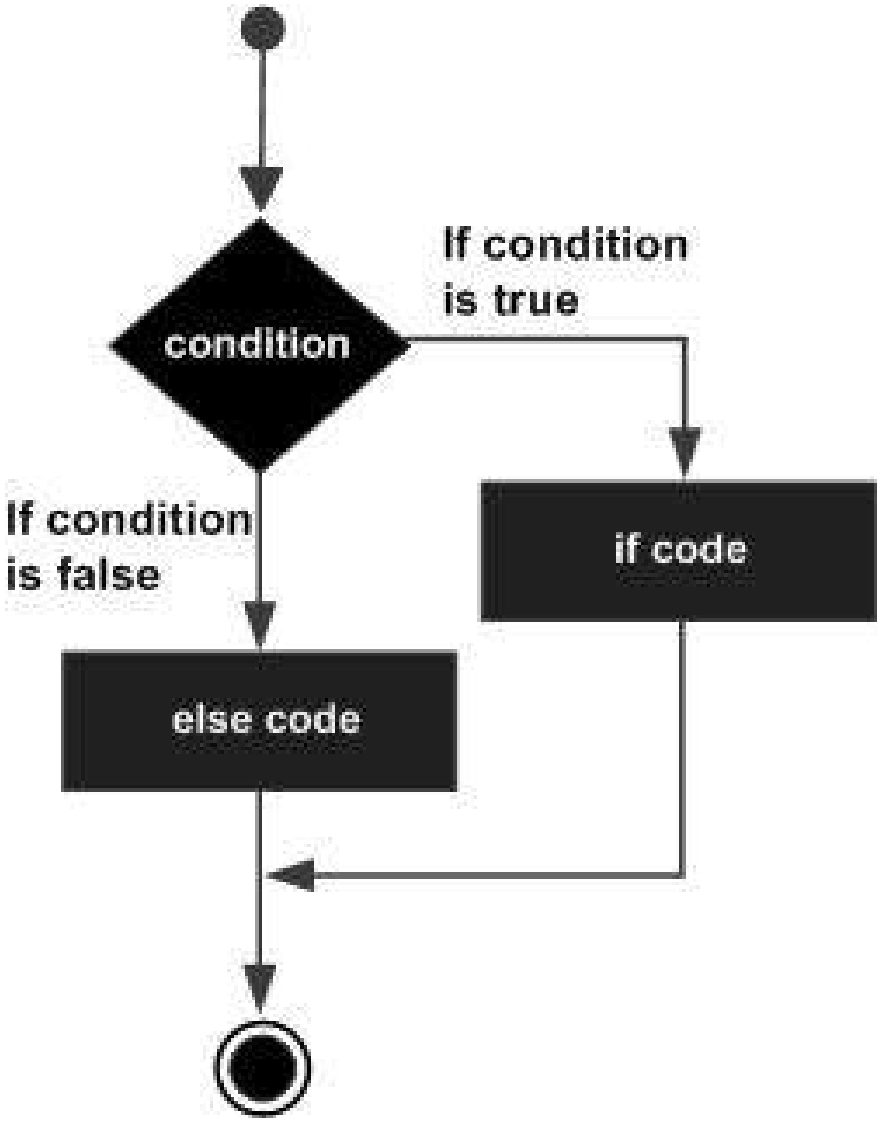
Syntax of an **if . else** command in Language C is:

```
if ( bieu_thuc_boolean ) { /* cac lenh se duoc thuc thi neu bieu thuc boolean la
```

If the logical expression is evaluated to be true, then the if block will be executed, otherwise the else block will be executed.

The C language assumes that any non-zero and non-null values ??are true, and if it is zero or null, then it is assumed to be false.

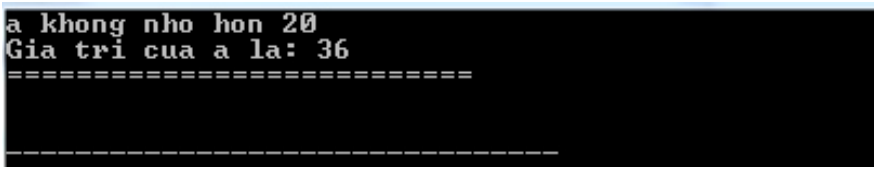
Diagram:



For example:

```
#include int  main  () { /* phan dinh nghia bien cuc bo */ int  a  = 36 ; /* kic
```

Compiling and executing the above C program will produce the following results:



If nested statement in C

It is valid to **nest** if-else statements in Language C, which means you can use an if or else statement inside another if or else statement.

Syntax:

Syntax to **nest if statements** :

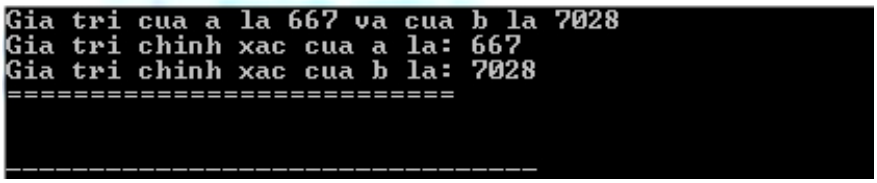
```
if (  bieu_thuc_boolean  1 ) { /* Thuc thi khi bieu thuc boolean 1 la true */ i
```

You can nested **else if . else** in the same way as if you had inserted the **if statement** .

For example:

```
#include int  main  () { /* phan dinh nghia bien cuc bo */ int  a  = 667 ; int
```

Compiling and executing the above C program will produce the following results:



Switch command in C

A **switch** command for a variable is checked equally in the list of values. Each value is called a case - the case and the variable passed are checked for each **switch case** .

Syntax:

The syntax of **switch** command in Language C is as follows:

```
 switch ( bieu_thuc ){ case  bieu_thuc_hang  :  cac_lenh ; break ; /* tuy y */ ca
case  tuy y */ default : /* tuy y */  cac_lenh ; }
```

The following rules apply to a switch command:

The expression is used in a switch command that must be of type integer or enumeration, or one of the class types in which the class has a single function that converts to an integer or enumeration type.

You can have any number of case commands in a switch. Each case is followed by the value to be compared and a colon.

bieu_thuc_hang for a case must be the same data type as the variable in the switch, and it must be constant.

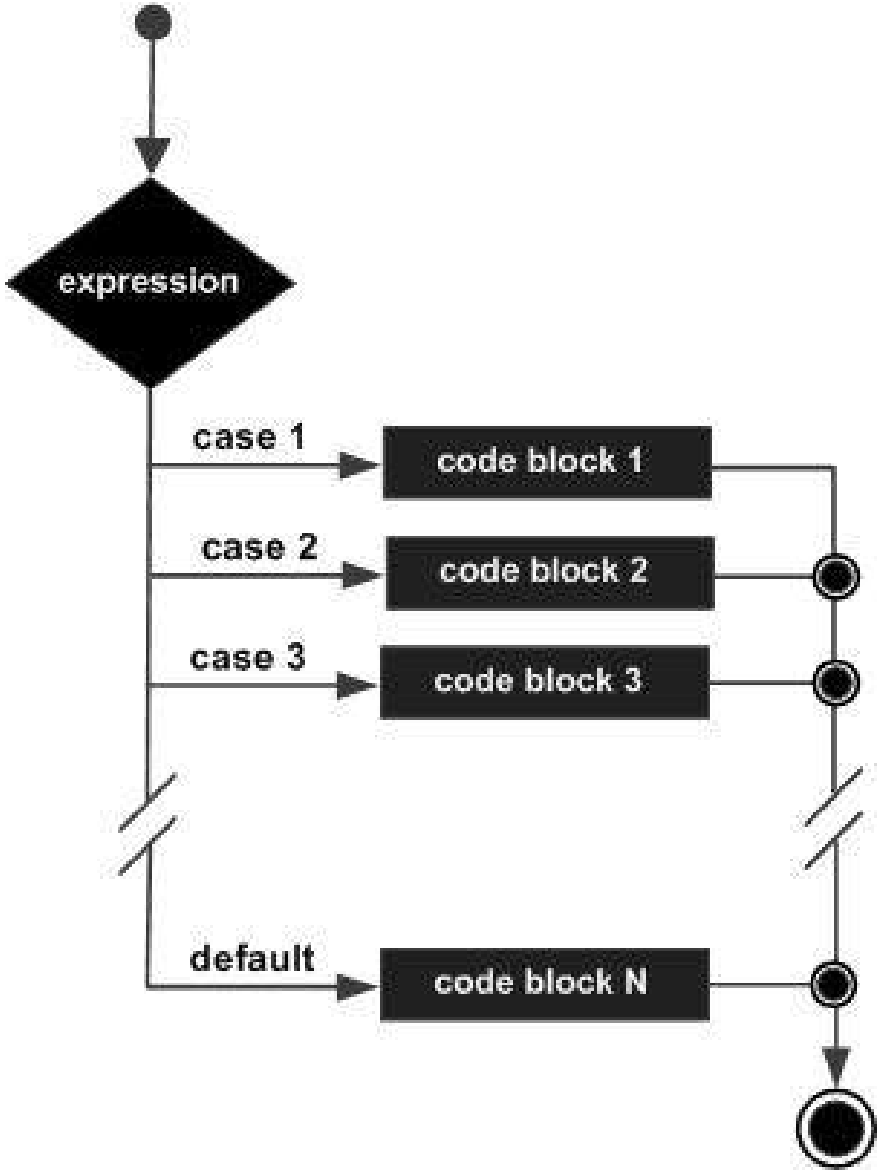
When the variable passed is balanced with a case, the following case statement will execute until a **break** statement is encountered.

When the **break** command is encountered, the switch ends, and the control line jumps to the next command line of the switch.

Not every case should contain a **break** statement. If no break statement appears, the control line will **not reach the** next case until a break statement is encountered.

A **switch** command may have an optional default (default) case, which must appear at the end of the switch. This default case can be used to perform a task when there is no true case. In this default case, there is no break command required.

Diagram:



For example:

```
#include int  main  () { /* phan dinh nghia bien cuc bo */ char  hocluc  = 'B'
```

Compiling and executing the above C program will produce the following results:

```
Gioi
Hoc luc cua sinh vien la  B
============================


--------------------------------
```

Insert switch commands in C

It is possible to have a switch command as part of the command sequence in a switch command on the outer ring. Even if the case constant inside and outside the switch command contains normal values, no conflict will occur here.

Syntax:

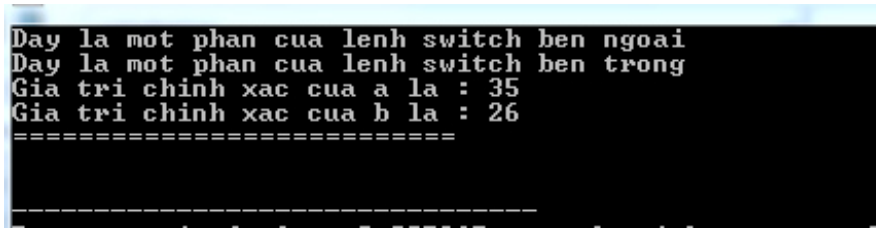
The syntax to **insert** switch commands is as follows:

```
switch ( ch1 ) { case 'A' :  printf ( "A la mot phan cua lenh switch ben ngoai"
```

For example:

```
#include int  main  () { /* phan dinh nghia bien cuc bo */ int  a  = 35 ; int  b
```

Compiling and executing the above C program will produce the following results:



According to Tutorialspoint

Previous article: Operator in programming C

Next lesson: Loop in programming C