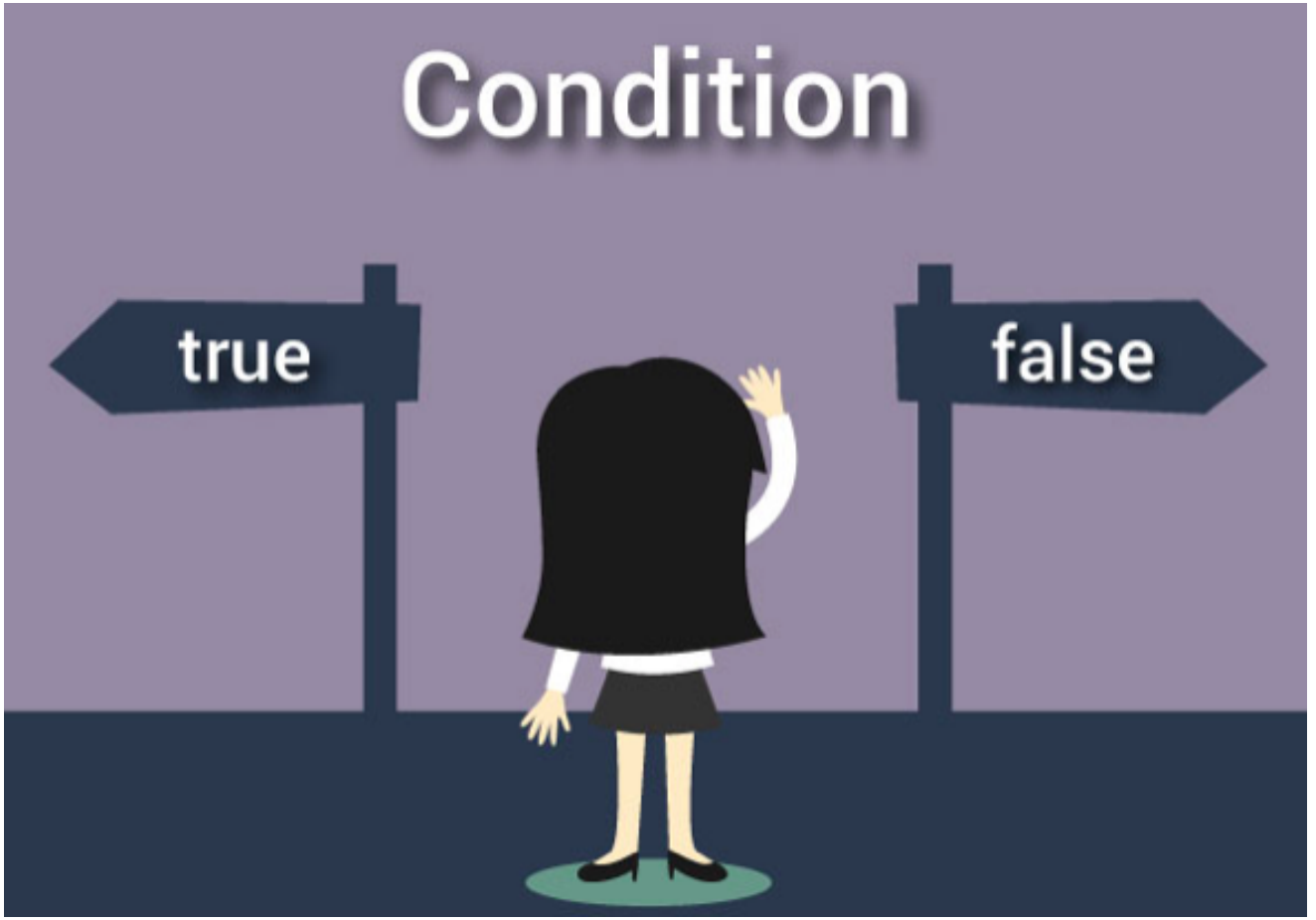
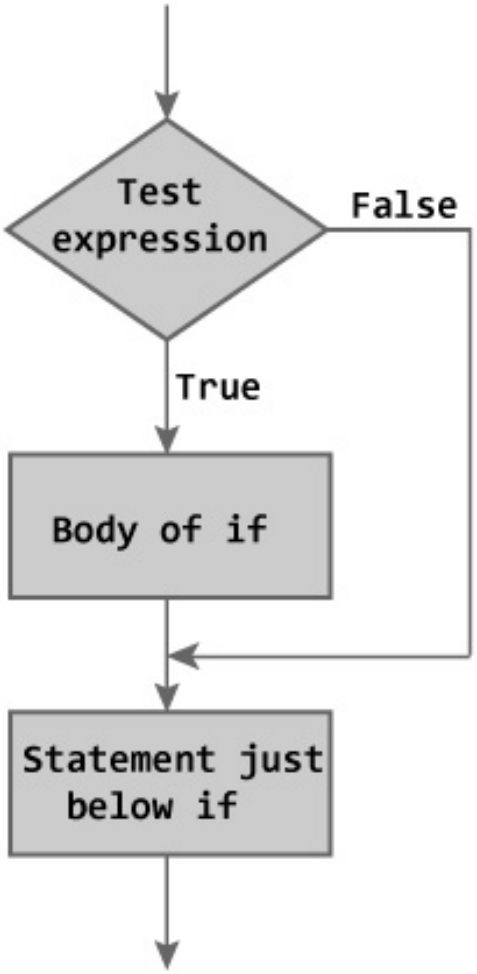
C Programming

Kasun De Zoysa



In programming, decision making is used to specify the order in which statements are executed.

C if statement




```
if (testExpression)
{
    // statements
}
```

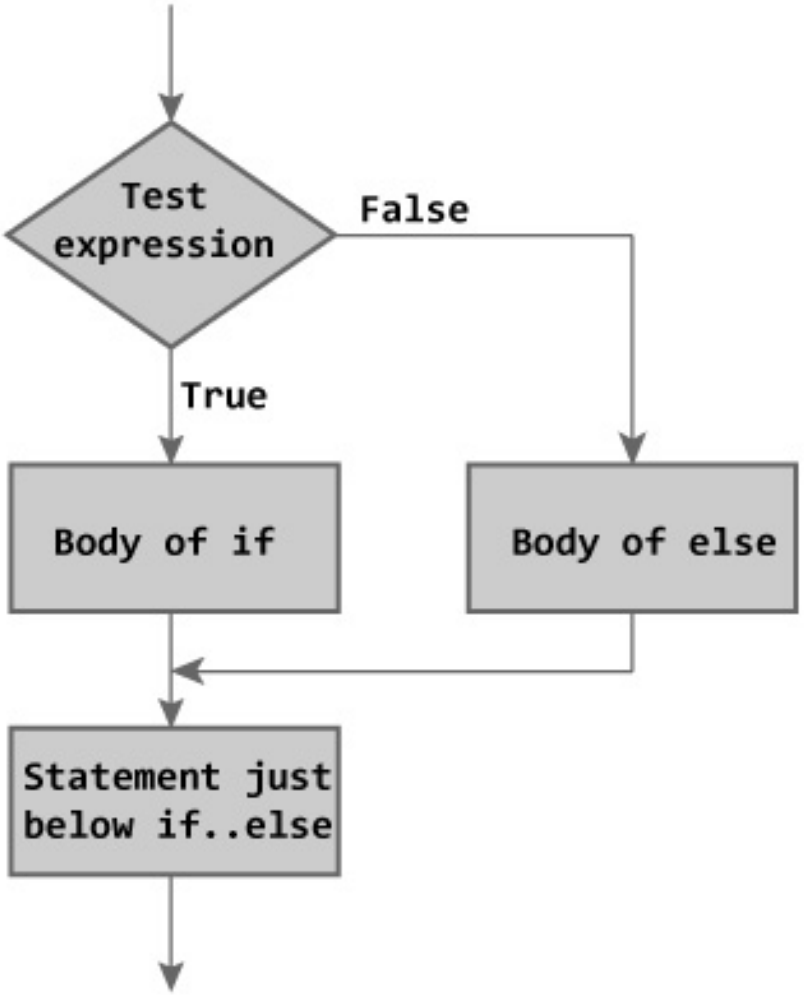
The if statement evaluates the test expression inside the parenthesis.

If the test expression is evaluated to true (nonzero), statements inside the body of if is executed.

If the test expression is evaluated to false (0), statements inside the body of if is skipped from execution.

C if...else statement

```
if (testExpression) {
     // codes inside the body of if
}
else {
     // codes inside the body of else
}
```



If test expression is true, codes inside the body of if statement is executed and, codes inside the body of else statement is skipped.

If test expression is false, codes inside the body of else statement is executed and, codes inside the body of if statement is skipped.

Odd and Even Numbers

```c
// Program to check whether an integer entered by the user is odd or even

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

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

Nested if...else statement (if...else if....else Statement)

```
if (testExpression1)
{
    // statements to be executed
}
else if(testExpression2)
{
    // statements to be executed
}
else if (testExpression 3)
{
    // statements to be executed
}
.
.
else
{
    // statements to be executed
}
```

Bank Interest

Develop the function interest. It consumes a deposit amount and produces the actual amount of interest that the money earns in a year.

The bank pays
a flat 4% for deposits of up to Rs. 10000,
a flat 5% per year for deposits of up to Rs. 100000,
a flat 10% per year for deposits of up to Rs. 1000000,
and a flat 15% for deposits of more than Rs. 1000000.

C switch...case Statement

The if..else..if ladder allows you to execute a block code among many alternatives.

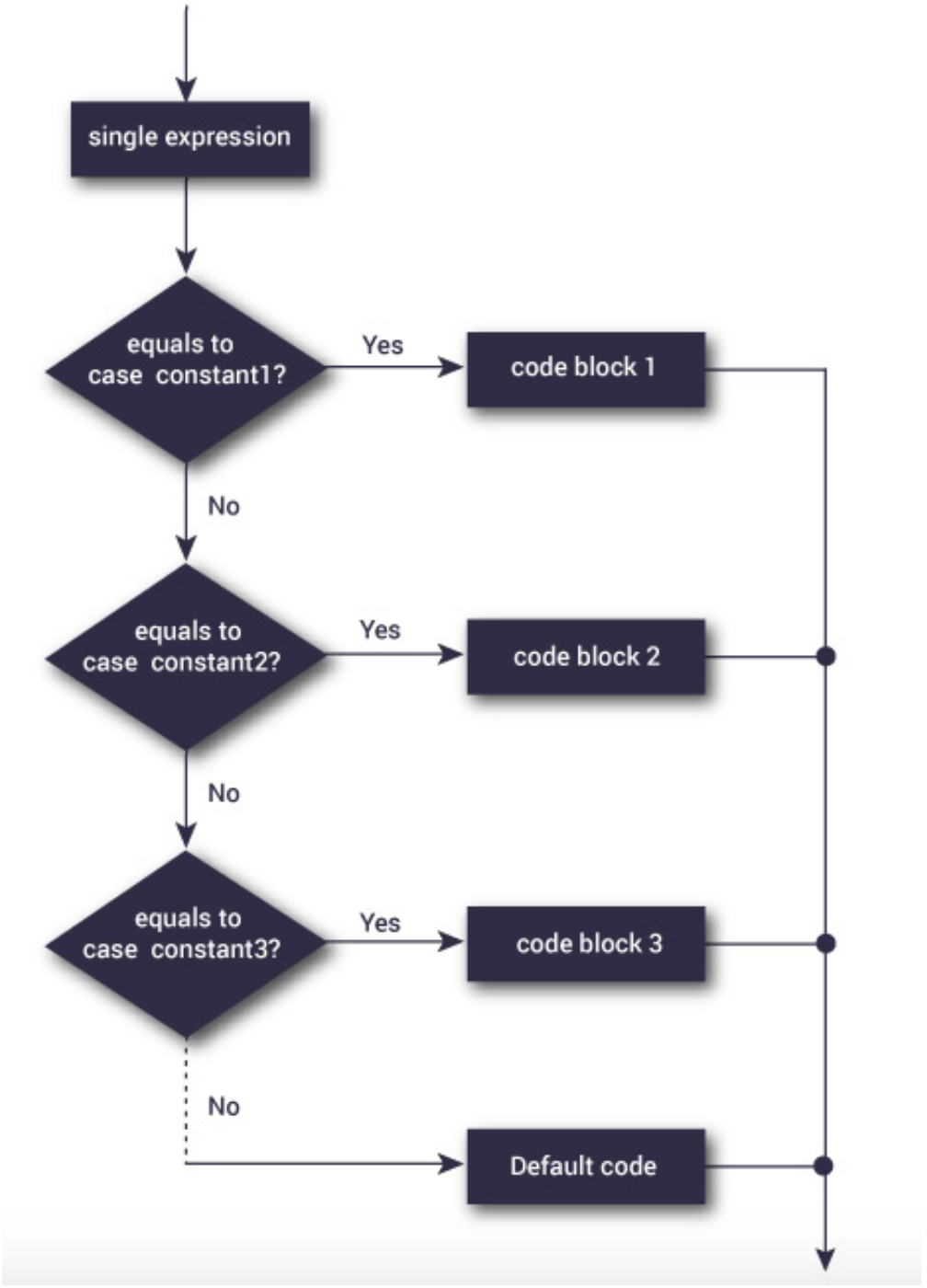
If you are checking on the value of a single variable in if...else...if, it is better to use switch statement.

The switch statement is often faster than nested if...else.

Syntax of switch...case

```
switch (n)
{
     case constant1:
          // code to be executed if n is equal to constant1;
          break;

     case constant2:
          // code to be executed if n is equal to constant2;
          break;
          .
          .
          .
     default:
          // code to be executed if n doesn't match any constant
}
```

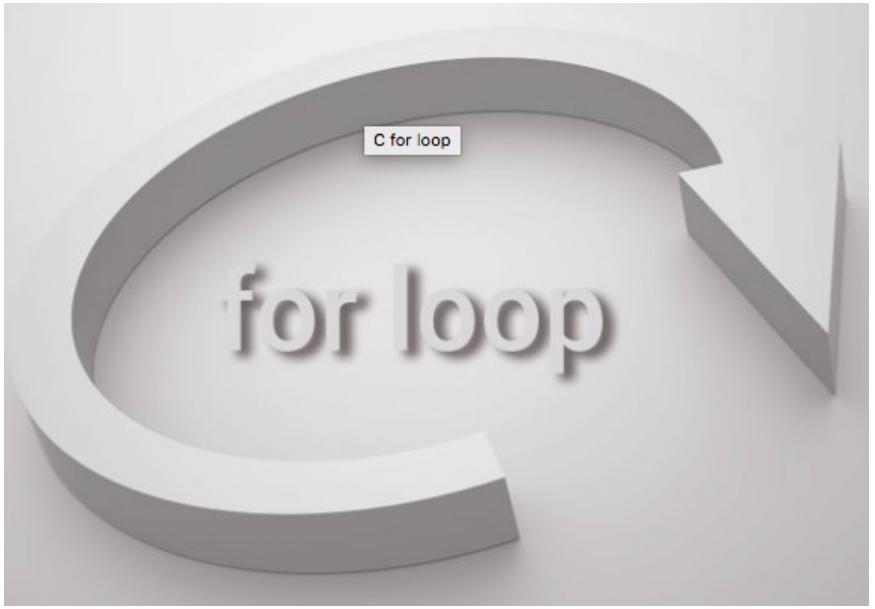

single expression
equals to
case constant1?
Yes
code block 1
No
equals to
case constant2?
Yes
code block 2
No
equals to
case constant3?
Yes
code block 3
No
Default code

The goto statement is used to alter the normal sequence of a C program.

```
goto label;
... .. ...
... .. ...

label:
... .. ...
... .. ...
```

When goto statement is encountered, control of the program jumps to label: and starts executing the code.



Loops are used in programming to repeat a specific block of code.

There are three loops in C programming:

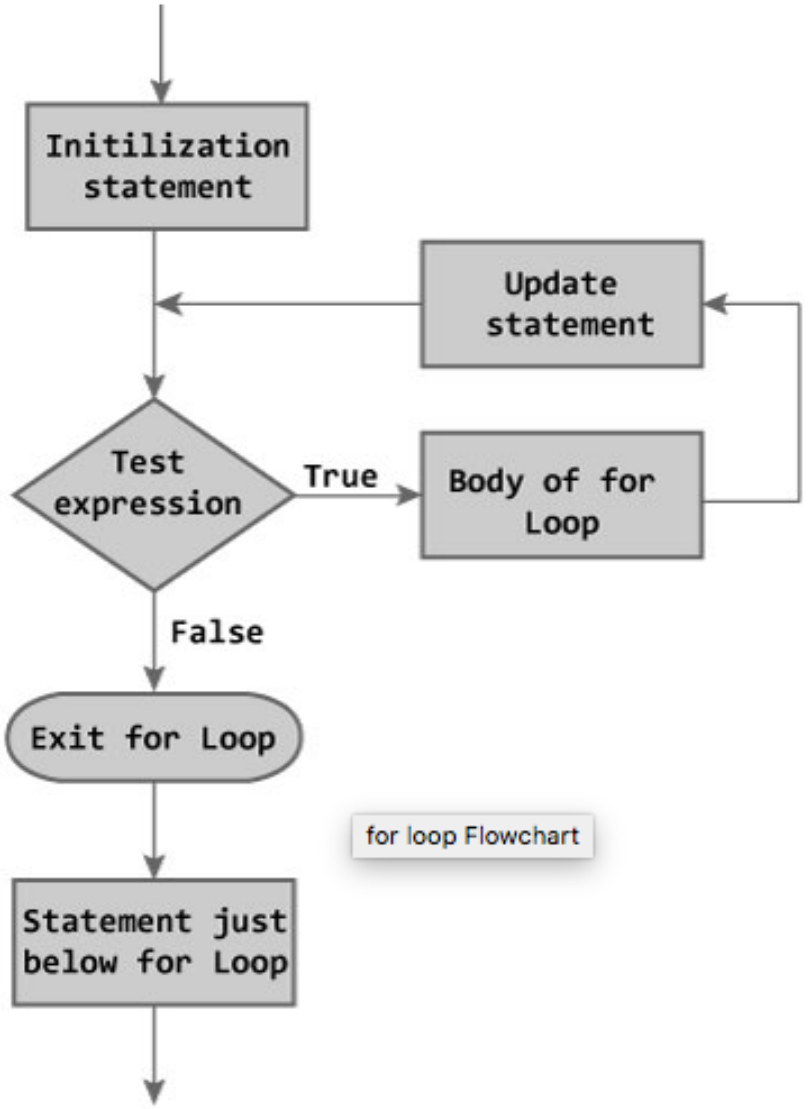
1. for loop

2. while loop

3. do...while loop

for Loop

```
for (initializationStatement; testExpression; updateStatement)
{
        // codes
}
```



for loop Flowchart

How for loop works?

The initialization statement is executed only once.

Then, the test expression is evaluated. If the test expression is false (0), for loop is terminated.

But if the test expression is true (nonzero), codes inside the body of for loop is executed and the update expression is updated.

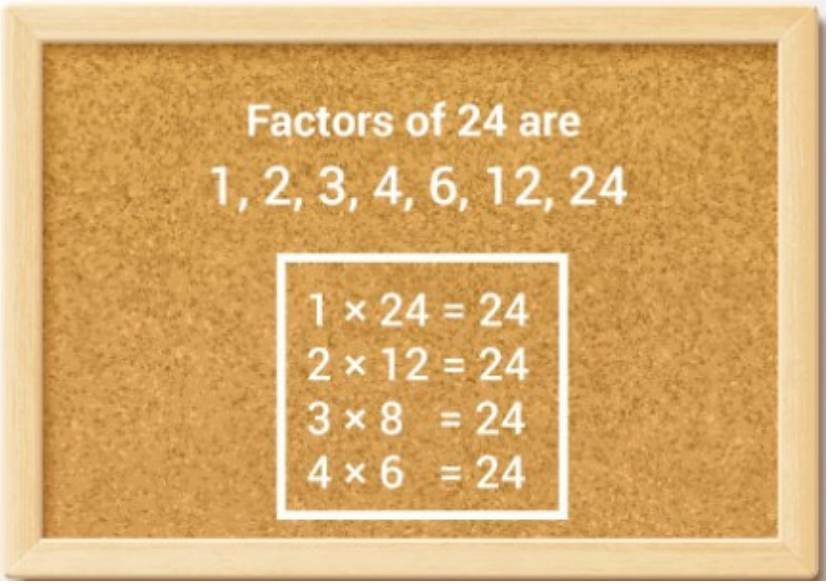
This process repeats until the test expression is false.

The for loop is commonly used when the number of iterations is known.

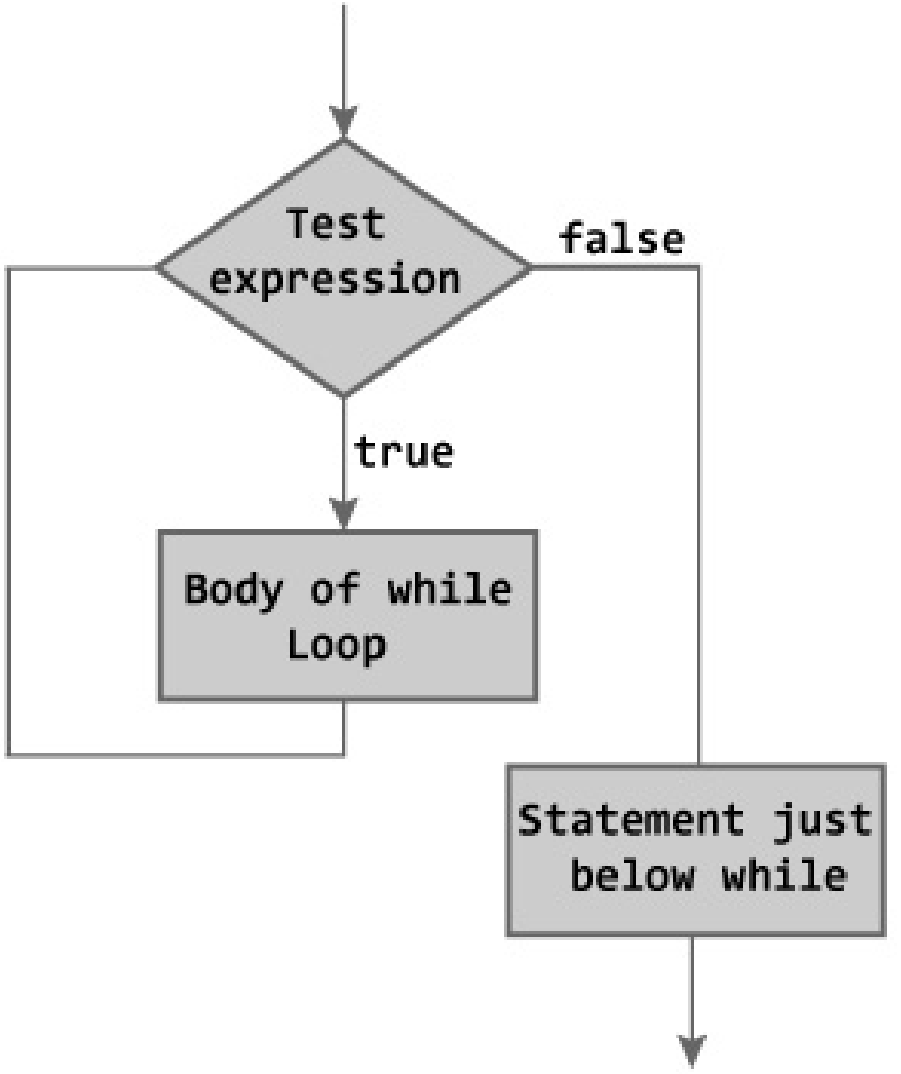
Factors??

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

  printf("Factors of %d are: ", number);
  for(i=1; i <= number; ++i) {
    if (number%i == 0) printf("%d ",i);
   }
  printf("\n");
  return 0;
}
```




```
while (testExpression)
{
     //codes
}
```



How while loop works?

The while loop evaluates the test expression.

If the test expression is true (nonzero), codes inside the body of while loop are executed

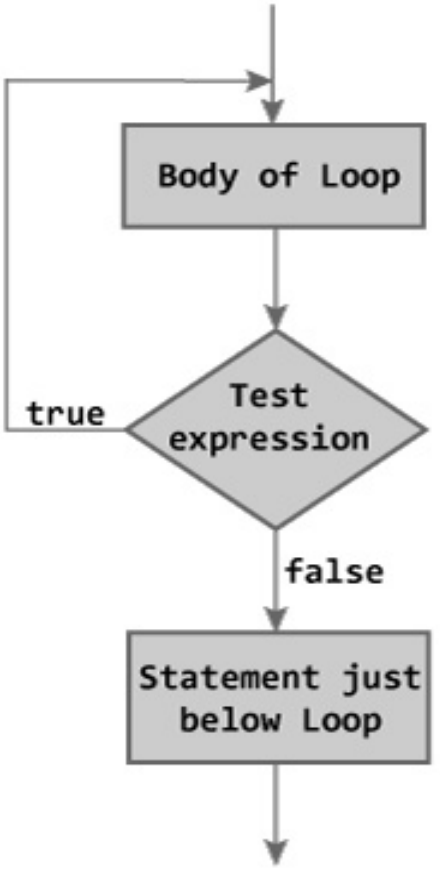
The test expression is evaluated again. The process goes on until the test expression is false.

When the test expression is false, the while loop is terminated.

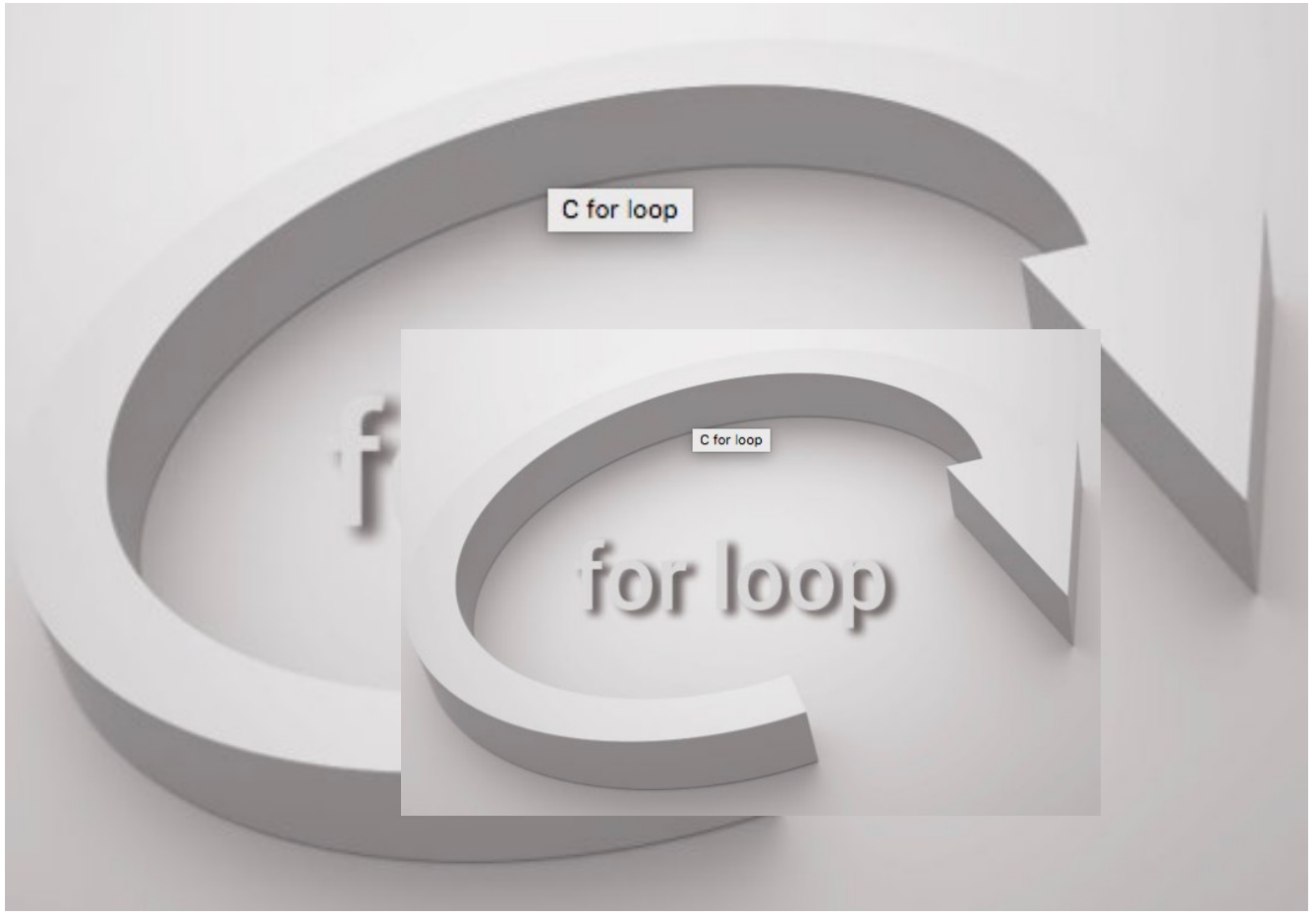
do...while loop

The do..while loop is similar to the while loop with one important difference. The body of do...while loop is executed once, before checking the test expression. Hence, the do...while loop is executed at least once.

```
do
{
    // codes
}
while (testExpression);
```



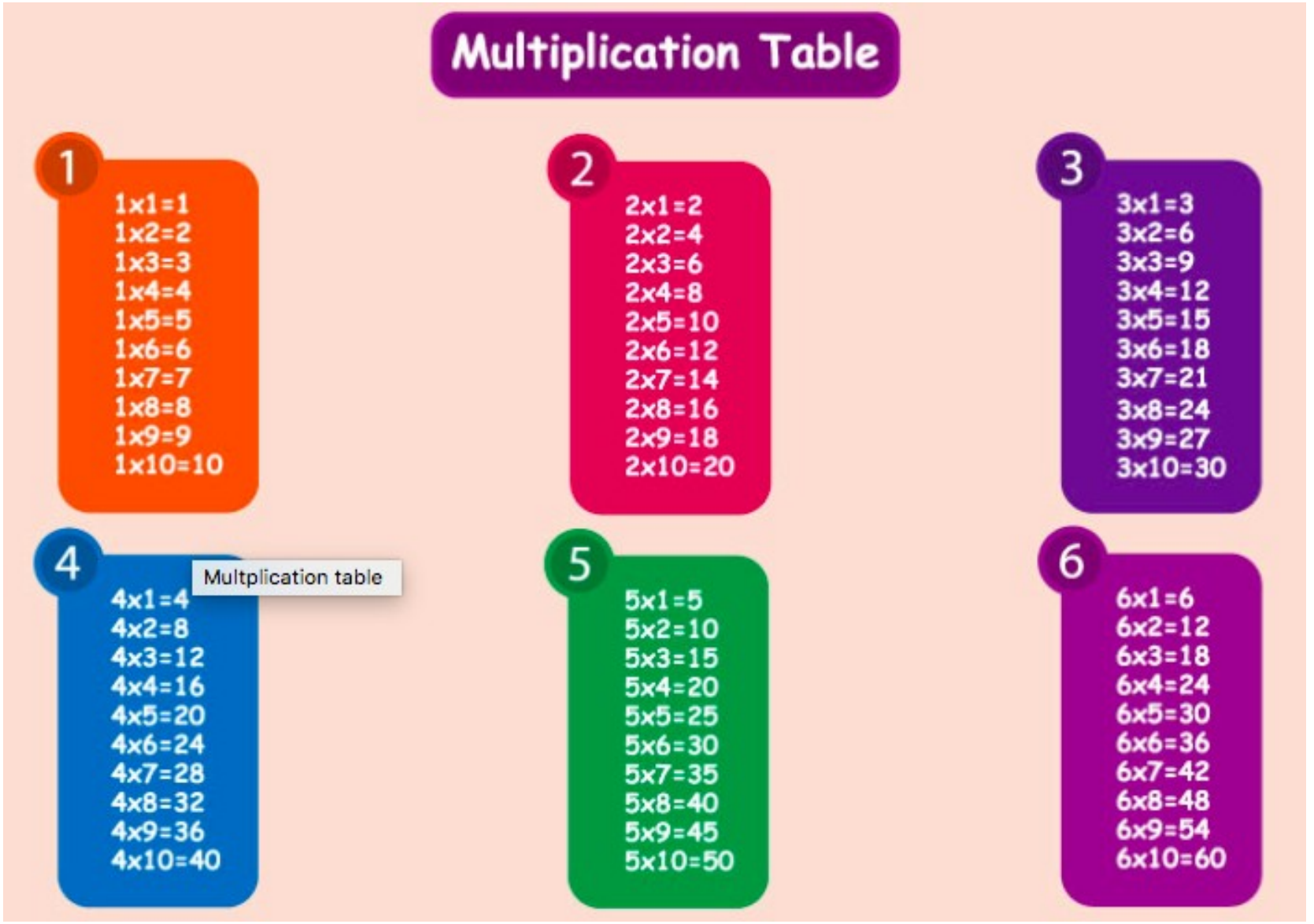
Loop inside Loop



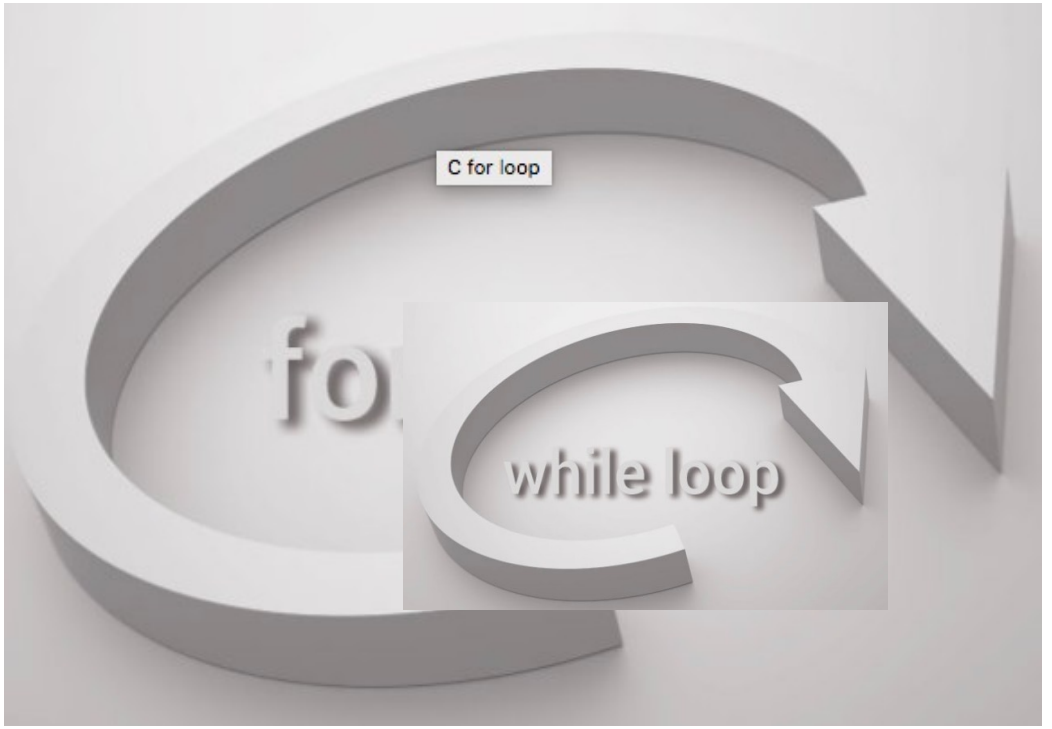
Problem

Write a function to print the following pattern by using the given character.

```
*
**
***
****
******
```

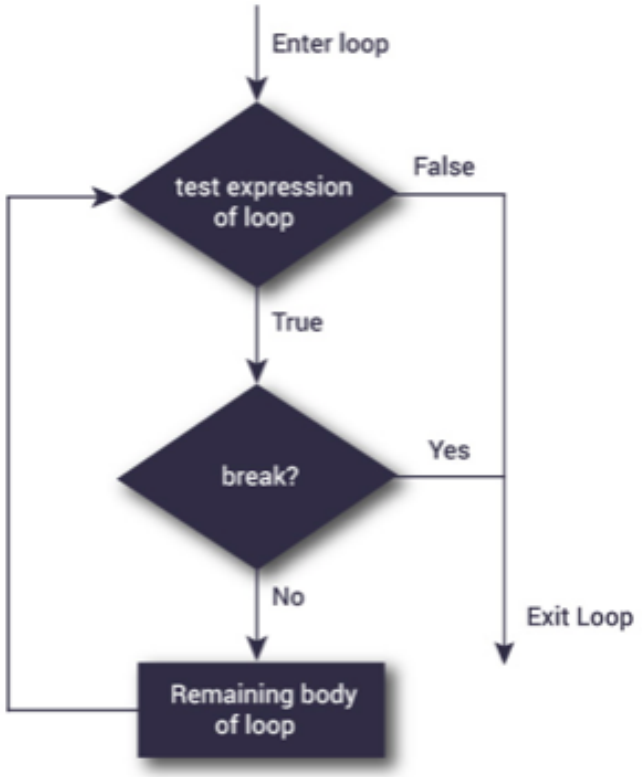

Multiplication Table
1
1x1=1
1x2=2
1x3=3
1x4=4
1x5=5
1x6=6
1x7=7
1x8=8
1x9=9
1x10=10
2
2x1=2
2x2=4
2x3=6
2x4=8
2x5=10
2x6=12
2x7=14
2x8=16
2x9=18
2x10=20
3
3x1=3
3x2=6
3x3=9
3x4=12
3x5=15
3x6=18
3x7=21
3x8=24
3x9=27
3x10=30
4
Multplication table
4x1=4
4x2=8
4x3=12
4x4=16
4x5=20
4x6=24
4x7=28
4x8=32
4x9=36
4x10=40
5
5x1=5
5x2=10
5x3=15
5x4=20
5x5=25
5x6=30
5x7=35
5x8=40
5x9=45
5x10=50
6
6x1=6
6x2=12
6x3=18
6x4=24
6x5=30
6x6=36
6x7=42
6x8=48
6x9=54
6x10=60


whi
while loop


C for loop
fo
while loop

break Statement

The break statement terminates the loop (for, while and do...while loop) immediately when it is encountered. The break statement is used with decision making statement such as if...else.

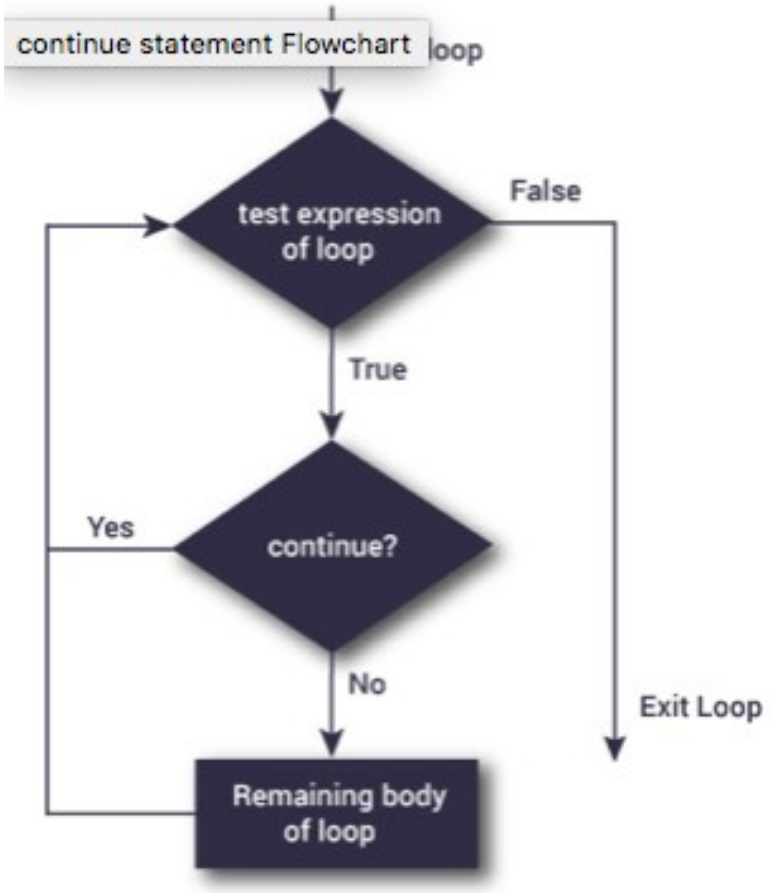



```
while (test Expression)
{
    // codes
    if (condition for break)
    {
        break;
    }
    // codes
}
```

```
for (init, condition, update)
{
    // codes
    if (condition for break)
    {
        break;
    }
    // codes
}
```

Continue Statement

The continue statement skips some statements inside the loop. The continue statement is used with decision making statement such as if...else.




```
while (test Expression)
{
    // codes
    if (condition for continue)
    {
        continue;
    }
    // codes
}
```

```
for (init, condition, update)
{
    // codes
    if (condition for continue)
    {
        continue;
    }
    // codes
}
```

Problems?

1) Write a program to take a positive integer (let say n) as an input from the user and calculate the sum of all integers up to n.

2) Write a function to determine the given number is prime number

3) Write a function to return the largest Integer from the given set of numbers.

4) Write a C Program to Reverse a Number