Lab 07: Loop Control Structure contd.

*Further exploring the loop control structures with emphasis on the ‘for’ loop.*

1. **Loop**

The versatility of the computer lies in its ability to perform a set of instructions repeatedly. This involves repeating some portion of the program either a specified number of times until particular condition is being satisfied. there are methods by way of which we can repeat a part of program.

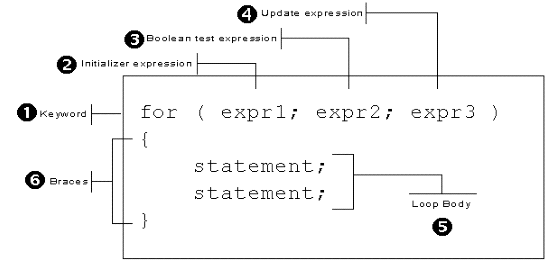
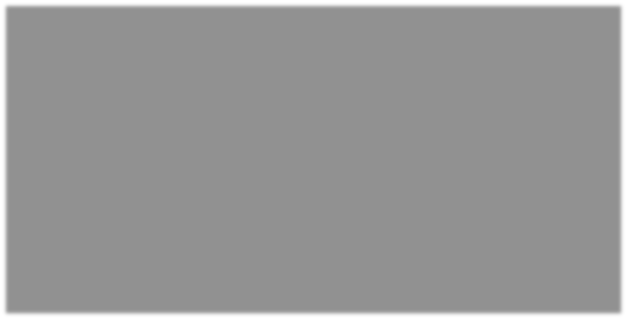
Types of loop

* + 1. Using a for statement
    2. Using a while statement
    3. Using a do while statement

# The FOR loop

The variable initialization allows you to either declare a variable and give it a value or give a value to an already existing variable. Second, the condition tells the program that while the conditional expression is true the loop should continue to repeat itself. The variable update section is the easiest way for a for loop to handle changing of the variable. It is possible to do things like x++, x = x + 10, or even x = random ( 5 ), and if you really wanted to, you could call other functions that do nothing to the variable but still have a useful effect on the code. Notice that a semicolon separates each of these sections, that is important. Also note that every single one of the sections may be empty, though the semicolons still have to be there. If the condition is empty, it is evaluated as true and the loop will repeat until something else stops it.

### 1.1.1 Syntax of the ‘for’ loop



for ( variable initialization; condition; variable update )

{

Code to execute while the condition is true

}

#### Figure 1 Syntax of for loop

**Flow chart:**

.

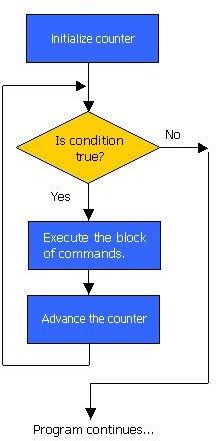


Figure 2 Flow chart of for loop

**EXAMPLE 1**

#include <stdio.h> main()

{

int x;

/\* The loop goes while x < 10, and x increases by one every loop\*/ for ( x = 0; x < 10; x++ ) {

/\* Keep in mind that the loop condition checks

the conditional statement before it loops again.

Student Name: Roll No: Section:

consequently, when x equals 10 the loop breaks. x is updated before the condition is checked. \*/

printf( "%d\n", x );

}

getch();

}

This program is a very simple example of a for loop. x is set to zero, while x is less than 10 it calls printf to display the value of the variable x, and it adds 1 to x until the condition is met. Keep in mind also that the variable is incremented after the code in the loop is run for the first time.

# Nesting of Loops

The way if statements can be nested, similarly whiles and fors can also be nested. To understand how nested loops work, look at the program given below:

### EXAMPLE 2

main( )

{

int r, c, sum ;

for ( r = 1 ; r <= 3 ; r++ ) /\* outer loop \*/

{

for ( c = 1 ; c <= 2 ; c++ ) /\* inner loop \*/

{

sum = r + c ;

printf ( "r = %d c = %d sum = %d\n", r, c, sum ) ;

} }

.

**Output:**

r = 1 c = 1 sum = 2

r = 1 c = 2 sum = 3

r = 2 c = 1 sum = 3

r = 2 c = 2 sum = 4

r = 3 c = 1 sum = 4

r = 3 c = 2 sum = 5

Here, for each value of r the inner loop is cycled through twice, with the variable c taking values from 1 to 2. The inner loop terminates when the value of c exceeds 2, and the outer loop terminates when the value of r exceeds 3.

Student Name: Roll No: Section:

As you can see, the body of the outer for loop is indented, and the body of the inner for loop is further indented. These multiple indentations make the program easier to understand. Instead of using two statements, one to calculate sum and another to print it out, we can compact this into one single statement by saying:

printf ( "r = %d c = %d sum = %d\n", r, c, r + c ) ;

The way for loops have been nested here, similarly, two while loops can also be nested. Not only this, a for loop can occur within a while loop, or a while within a for.

# Patterns using loops:

**Write a C Program to print half pyramid as using \* as shown in figure below.**

\*

\* \*

## \* \* \*

\* \* \* \*

## \* \* \* \* \*

#include <stdio.h>

main()

{

int i,j,rows;

printf("Enter the number of rows: ");

scanf("%d",&rows);

for(i=1;i<=rows;++i)

{

for(j=1;j<=i;++j)

{

printf("\* ");

}

printf("\n");

}

}

.

**Student Tasks:**

### Task 1:

Write a C Program to print half pyramid as using numbers as shown in figure below.

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

### Task 2:

Write a C program to print square star (\*) pattern series of n rows. For example if n=5 the star pattern should be printed like:

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

### Task 3:

Write a C program to print Fibonacci series up to n terms.

**Fibonacci series algorithm:**

If you look to the given series very carefully you will find a specific pattern i.e. the current number is the sum of previous two numbers.

**Fibonacci series: 0, 1, 1, 2, 3, 5, 8, 13, 21...**

1. is the sum of 0 + 1
2. is the sum of 1 + 1
3. is the sum of 2 + 1

5 is the sum of 3 + 2 and so on....

Lets suppose the first number as a i.e. a = 0 Second number as b i.e. b = 1

And lets suppose c as our current number in Fibonacci series also initialize the value with 0

i.e. c = 0

Now Fibonacci series algorithm is simple and contains only four steps.

**The Rule is xn = xn-1 + xn-2**

Step 1: Print the value of c. Step 2: a = b.

Step 3: b = c. Step 4: c = a + b.

**Fibonacci Series:**

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946,

17711, 28657, 46368, 75025, 121393, 196418, 317811, ...

**Fact:**

Fibonacci Day is November 23rd, as it has the digits "1, 1, 2, 3" which is part of the sequence.

### Task 4:

Use for-loop to print the multiplication table below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 |
| 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 |
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 |
| 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 |

### Task 5:

#### Write a C program to enter any number and print table of the given number using for loop. Format of the table should be like: If num =5

5 \* 1 =5

#### 5 \* 2 =10

...

#### ...

5 \* 10 =50

### Task 6:

Write a C program that will print the following pattern:

1\*\*\*\*\*\*

12\*\*\*\*\*

123\*\*\*\*

1234\*\*\*

12345\*\*

123456\*

1234567

### Task 7:

#### Read a positive integer value, and compute the following sequence:

* If the number is even, halve it

#### If it's odd, multiply by 3 and add 1.

Repeat this process until the value is 1, printing out each value. Finally print out how many of these operations you performed. Typical output might be:

Inital value is 9 Next value is 28 Next value is 14 Next value is 7 Next value is 22 Next value is 11 Next value is 34 Next value is 17 Next value is 52 Next value is 26 Next value is 13 Next value is 40 Next value is 20 Next value is 10 Next value is 5 Next value is 16 Next value is 8 Next value is 4 Next value is 2

Final value 1, number of steps 19

If the input value is less than 1, print an error message.

### Task 8:

. **Write a C Program to print inverted half pyramid using \* as shown below.**

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

### Task 9:

Write a C program to print rhombus star(\*) pattern series using for loop. The pattern should look like:

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

### Task 10:

Assume that the salesmen of a company are paid a commission based on the number of sales made during a week. The staff is paid a commission of $8 for sales less than 15, $12 for 15 sales, and $16 if sales exceed 15. Find the commission for each salesman, if the total number of salesmen in the company are taken input from the user.

Student Name: Roll No:

**Practice Problems:**

Section:

### Task 1:

#### . A dice is rolled n number of times. Show the possible outcomes of dice. For example, if a dice is rolled twice the possible outcomes would be:

(1,1), (1,2),(1,3), (1,4), (1,5),(1,6)

#### (2,1), (2,2),(2,3), (2,4), (2,5),(2,6)

(3,1), (3,2),(3,3), (3,4), (3,5),(3,6)

#### (4,1), (4,2),(4,3), (4,4), (4,5),(4,6)

(5,1), (5,2),(5,3), (5,4), (5,5),(5,6)

(6,1), (6,2),(6,3), (6,4), (6,5),(6,6)

### Task 4:

Write a program to calculate the result of the series accurate up to 7th digit: x+(x^3)/3!+(x^5)/5!+…

### Task 4:

Write a C Program to print triangle of characters as below

A B B

C C C

D D D D E E E E E