NESTED LOOPS

In many cases we may use loop statement inside another looping statement. This type of looping is called nested looping.

TYPES OF NESTED LOOP:

- NESTED FOR LOOP
- NESTED WHILE LOOP
- NESTED DO-WHILE LOOP

NESTED FOR LOOP:

A for loop inside another for loop is called nested for loop.

SYNTAX:

for (initialization; condition; increment/decrement)

{

for (initialization; condition; increment/decrement)

{

body of the loop;

}

}

The inner loop runs as many times as there is the limit of the condition of the external loop. This loop runs as long as the condition in the parenthesis is true.

Ex:

```
#include<stdio.h>
int main() {
   int i, j;

   for (i = 1; i <= 4; i++) {
      for (j = 1; j <= 3; j++) {
         printf("*");
      }
      printf("\n");
   }
   return 0;
}
```

Output:

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

NESTED WHILE LOOP:

Using while loop within while loop is said to be nested while loop. It is used to execute a set of code for multiple times as long as the condition is true. Once the condition becomes false the code block is no longer executed.

SYNTAX:

while(test condition)

{

while (test condition)

{

Body of the inner loop;

}

Body of the outer loop;

}

Ex:

```
#include<stdio.h>
int main() {
   int i, j;
   i = 1;
   while (i <= 4) {

      j = 1;
      while (j <= 3) {
         printf("*");
         j++;
      }
      printf("\n");
      i++;
   }
   return 0;
}
```

Output:

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

NESTED DO-WHILE LOOP:

A do-while loop inside another dowhile loop is called nested do-while loop.

SYNTAX:

do

{

do

{

Body of the inner loop;

} while (condition of the inner loop);

Body of the outer loop;

} while (condition of the outer loop);

Ex:

```
#include<stdio.h>
int main() {
   int i, j;
   i = 1;
   do {

      j = 1;
      do {
         printf("*");
         j++;
      } while (j <= 3);
      printf("\n");
      i++;
   } while (i <= 4);
   return 0;
}
```

Output:

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

EXERCISE 1: Write c programs that prints the following patterns (Hint: you need nested loops).

1.

```
**#**
**#**
**#**
**#**
```

2.

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

3.

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

EXERCISE 2: Write a program to accept any 20 numbers and display factorial of each number.

EXERCISE 3: Write c program to evaluate the following equation:

$$\sum_{i=1}^{i=100} \sum_{j=1}^{j=i} j^2 * i^2$$

EXERCISE 4:

```c
#include <stdio.h>
int main()
{
    int i;
    int x=1;
    for (i=1;i<=5;i++)
    {
        printf("%d ", (i*x)+1);
    }
    return 0;
}
```

1. Type, compile, and run the above code, write down the output.
2. Modify the code to produce the following :

   2 3 4 5 6
   3 5 7 9 11
   4 7 10 13 16
   5 9 13 17 21

3. Modify the code to prints eight rows (keeping the same pattern).
4. Modify the code to prints eight columns (keeping the same pattern).

EXERCISE 5: Write a C program that Finds the **sum of the perfect square** numbers between 1 and 1000. (The perfect square number is an integer that is the square of an integer).