

**FOUNDATION UNIVERSITY ISLAMABAD**  
**SCHOOL OF SCIENCE AND TECHNOLOGY**

**LAB MANUAL 06**

**NAME : ALISHBA HAFEEZ**

**SUBJECT: PF (programming fundamentals)**

**SECTION : 25(B)**

**DEPARTMENT : IET**

**ROLL NO : 056**

**INSTRUCTOR : Mr. SHAMSHAIR ALI**

**TASK 6.1 (A):**

**Nested ‘for’ Statement (Drawing Patterns with Nested for Loops)**

Write a program that uses for statements to print the following patterns separately, one below the other. Use for loops to generate the patterns. All asterisks (\*) should be printed by a single statement of the form cout << " "; (this causes the asterisks to print side by side).

[Hint: The last two patterns require that each line begin with an appropriate number of blanks. Extra credit: Combine your code from the four separate problems into a single program that prints all four patterns side by side by making clever use of nested for loops.]

**CODE:**

```
#include<iostream>
using namespace std;
int main()
{
    for(int i = 0; i <= 10; i++)
    {
        for(int j = 0; j <= i; j++)
        {
            cout << "*";
        }
        cout << "\n";
    }
    return 0;
}
```

## OUTPUT:

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
-----  
Process exited after 0.2814 seconds with return value 0  
Press any key to continue . . . |
```

(B):

CODE:

```
#include<iostream>  
using namespace std;  
int main()  
{  
    for(int i = 0; i <= 10; i++)  
    {  
        for(int j = 10; j > i; j--)  
        {  
            cout << " ";  
        }  
        for(int k = 0; k < i; k++)  
        {  
            cout << "*";  
        }  
        cout << "\n";  
    }  
    return 0;  
}
```

## OUTPUT:

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
-----  
Process exited after 0.2363 seconds with return value 0  
Press any key to continue . . . |
```

(C):

## CODE:

```
#include<iostream>  
using namespace std;  
int main()  
{  
    for(int i = 0; i <= 10; i++)  
    {  
        for(int j = 0; j <= i; j++)  
        {  
            cout << " ";  
        }  
        for(int k = 10; k > i; k--)  
        {  
            cout << "*";  
        }  
        cout << "\n";  
    }  
    return 0;  
}
```

## OUTPUT:

```
Process exited after 0.2299 seconds with return value 0
Press any key to continue . . .
```

(D):

## **CODE:**

```
#include<iostream>
using namespace std;
int main()
{
    for(int i = 0; i <= 10; i++)
    {
        for(int j = 10; j > i; j--)
        {
            cout << " ";
        }
        for(int k = 0; k < i; k++)
        {
            cout << "*";
        }
        cout << "\n";
    }
    return 0;
}
```

## **OUTPUT:**

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
-----  
Process exited after 0.3229 seconds with return value 0  
Press any key to continue . . . |
```

## TASK 6.2:

Write a C++ program to generate a 4-digit number using nested for loops and displays each number from 0000 to 9999. You will have to use 3 nested ‘for’ loops to generate the output.

## CODE:

```
#include <iostream>  
using namespace std;  
int main()  
{  
    for (int i = 0; i <= 9; i++)  
    {  
        for (int j = 0; j <= 9; j++)  
        {  
            for (int k = 0; k <= 9; k++)  
            {  
                for (int l = 0; l <= 9; l++)  
                {  
                    cout << i << j << k << l << endl;  
                }  
            }  
        }  
    }  
    return 0;  
}
```

## OUTPUT:

```
9988
9989
9990
9991
9992
9993
9994
9995
9996
9997
9998
9999

-----
Process exited after 3.863 seconds with return value 0
Press any key to continue . . . |
```

### TASK 6.3:

Write a C++ program to display a triangular pattern of numbers using a nested ‘for’ loop.

### CODE:

```
#include<iostream>
using namespace std;
int main()
{
    for(int i = 1; i <= 10; i++)
    {
        for(int j = 1; j <= i; j++)
        {
            cout << j << " ";
        }
        cout << endl;
    }
    return 0;
}
```

### OUTPUT:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10

-----
Process exited after 0.2814 seconds with return value 0
Press any key to continue . . . |
```

## TASK 6.4:

Repeat Task 6.1 using a nested ‘while’ loop.

(A):

CODE:

```
#include<iostream>
using namespace std;
int main()
{
    int i = 0;
    while (i <= 10)
    {
        int j = 0;
        while (j <= i)
        {
            cout << "*";
            j++;
        }
        cout << "\n";
        i++;
    }
    return 0;
}
```

OUTPUT:

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****  
-----  
Process exited after 0.2345 seconds with return value 0
Press any key to continue . . . |
```

(B):

## CODE:

```
#include<iostream>
using namespace std;
int main()
{
    int i = 10;
    while (i >= 0)
    {
        int j = 0;
        while (j <= i)
        {
            cout << "*";
            j++;
        }
        cout << "\n";
        i--;
    }
    return 0;
}
```

## OUTPUT:

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
-----
Process exited after 0.2764 seconds with return value 0
Press any key to continue . . . |
```

(C):

## CODE:

```
#include<iostream>
using namespace std;
int main()
{
    int i = 0;
    while (i <= 10)
    {
        int j = 0;
        while (j <= i)
        {
            cout << " ";
            j++;
        }
        int k = 10;
        while(k > i)
        {
            cout << "*";
            k--;
        }
        cout << endl;
        i++;
    }
    return 0;
}
```

## OUTPUT:

```
*****
 ****
 ***
 **
 *
 -----
Process exited after 0.2764 seconds with return value 0
Press any key to continue . . . |
```

(D):

## CODE:

```
#include<iostream>
using namespace std;
int main()
{
    int i = 0;
    while (i <= 10)
    {
        int j = 10;
        while (j > i)
        {
            cout << " ";
            j--;
        }
    }
    int k = 0;
    while(k < i)
    {
        cout << "*";
        k++;
    }
    cout << endl;
    i++;
}
return 0;
```

## OUTPUT:

```

*
**
***
****
*****
******
*****
*****
*****
*****
-----
Process exited after 0.2703 seconds with return value 0
Press any key to continue . . . |
```

## TASK 6.5:

Repeat Task 6.2 using nested ‘while’ loop.

### CODE:

```
#include <iostream>
using namespace std;
int main()
{
    int i = 0;
    while (i <= 9)
    {
        int j = 0;
        while (j <= 9)
        {
            int k = 0;
            while (k <= 9)
            {
                int l = 0;
                while (l <= 9)
                {
                    cout << i << j << k << l << endl;
                    l++;
                }
                k++;
            }
            j++;
        }
        i++;
    }

    return 0;
}
```

### OUTPUT:

```
9989
9990
9991
9992
9993
9994
9995
9996
9997
9998
9999

-----
Process exited after 3.675 seconds with return value 0
Press any key to continue . . . |
```

## **TASK 6.6:**

Write a C++ program to display a triangular pattern of numbers using a nested ‘while’ loop.

### **CODE:**

```
#include<iostream>
using namespace std;
int main()
{
    int i = 1;
    while(i <= 10)
    {
        int j = 1;
        while(j <= i)
        {
            cout << j << " ";
            j++;
        }
        cout << endl;
        i++;
    }
    return 0;
}
```

### **OUTPUT:**

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10

-----
Process exited after 0.2907 seconds with return value 0
Press any key to continue . . . |
```

## **TASK 6.7:**

Write a C++ program to display a triangular pattern of numbers using a nested ‘do-while’ loop.

### **CODE:**

```
#include<iostream>
using namespace std;
int main()
{
    int i = 1;
    do
    {
        int j = 1;
        do
        {
            cout << "*";
            j++;
        }
        while(j <= i);

        cout << endl;
        i++;
    }

    while(i <= 10);

    return 0;
}
```

## OUTPUT:

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
-----  
Process exited after 0.3291 seconds with return value 0  
Press any key to continue . . . |
```

### **TASK6.8:**

Repeat Task 6.2 using nested ‘do-while’ loop.

## CODE:

```

#include <iostream>
using namespace std;
int main()
{
    int i = 0;

    do
    {
        int j = 0;
        do
        {
            int k = 0;
            do
            {
                int l = 0;
                do
                {
                    cout << i << j << k << l << endl;
                    l++;
                }
                while (l <= 9);
                k++;
            }
            while (k <= 9);
            j++;
        }
        while (j <= 9);
        i++;
    }
    while (i <= 9);
}

return 0;
}

```

## OUTPUT:

```

9990
9991
9992
9993
9994
9995
9996
9997
9998
9999

-----
Process exited after 3.407 seconds with return value 0
Press any key to continue . . .

```

## TASK 6.9:

Write a C++ program to display a triangular pattern of numbers using a nested ‘do-while’ loop.

## CODE:

```
#include <iostream>
using namespace std;
int main()
{
    int i = 1;

    do
    {
        int j = 1;
        do
        {
            cout << j << " ";
            j++;
        }
        while (j <= i);
        cout << endl;
        i++;
    }
    while (i <= 10);

    return 0;
}
```

## OUTPUT:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10

-----
Process exited after 0.2499 seconds with return value 0
Press any key to continue . . . |
```

## EXERCISE 6.1:

Write a C++ program to read an odd integer from the user. If the user enters an even number, your program should decrement it by one to generate an odd number. Using this number as the number of rows, your program should then display a diamond of \*'s on the screen. Your program should have the following interface.

### CODE:

```
#include <iostream>
using namespace std;
int main()
{
    int n;

    cout << "Enter an odd number of rows: ";
    cin >> n;

    if (n % 2 == 0)
    {
        n = n - 1;
    }

    int i, j;
    for (i = 1; i <= n; i += 2)
    {
        for (j = 1; j <= (n - i) / 2; j++)
        {
            cout << " ";
        }
        for (j = 1; j <= i; j++) {
            cout << "*";
        }
        cout << endl;
    }

    for (i = n - 2; i >= 1; i -= 2)
    {
        for (j = 1; j <= (n - i) / 2; j++) {
            cout << " ";
        }
        for (j = 1; j <= i; j++) {
            cout << "*";
        }
        cout << endl;
    }

    return 0;
}
```

## OUTPUT:

```
Enter an odd number of rows: 5
*
***
*****
 ***
 *
-----
Process exited after 1.502 seconds with return value 0
Press any key to continue . . . |
```

## EXERCISE 6.2:

Write a C++ program to produce the following multiplication table. You must use nesting of control structures.

## CODE:

```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 1; i <= 10; i++)
    {
        for (int j = 1; j <= 10; j++)
        {
            cout << i * j << "\t";
        }
        cout << endl;
    }
    return 0;
}
```

## OUTPUT:

```
1      2      3      4      5      6      7      8      9      10
2      4      6      8      10     12     14     16     18     20
3      6      9      12     15     18     21     24     27     30
4      8      12     16     20     24     28     32     36     40
5     10     15     20     25     30     35     40     45     50
6     12     18     24     30     36     42     48     54     60
7     14     21     28     35     42     49     56     63     70
8     16     24     32     40     48     56     64     72     80
9     18     27     36     45     54     63     72     81     90
10    20     30     40     50     60     70     80     90     100
-----
Process exited after 0.3329 seconds with return value 0
Press any key to continue . . . |
```

### **EXERCISE 6.3:**

Write a C++ program which reads in a number from user, stores it in a variable named ‘N’ and calculates the sum of powers using the following formula.

$$\text{Sum} = 1^1 + 2^2 + 3^3 + \dots + N^N$$

Your program should have the following interface.

### **CODE:**

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    int N;
    long long sum = 0;
    cout << "Enter a positive integer N: ";
    cin >> N;

    for (int i = 1; i <= N; i++)
    {
        sum += pow(i, i);
    }
    cout << "Sum = " << sum << endl;

    return 0;
}
```

### **OUTPUT:**

```
Enter a positive integer N: 4
Sum = 288
```

```
-----
Process exited after 2.456 seconds with return value 0
Press any key to continue . . . |
```

\*\*\*\*\*THE END\*\*\*\*\*