

Logic Building Assignment : 75

1. Write java program to print below pattern.

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

```
public class PrintPattern
{
    public static void main(String args[])
    {
        int space=0;
        for(int i=5; i>=1; i--)
        {
            //print first part of the row
            for(int j=i; j>=1; j--)
                System.out.print("*");

            //print space
            for(int j=1; j<=space; j++)
                System.out.print(" ");

            //print second part of the row
            for(int j=i; j>=1; j--)
                System.out.print("*");

            //print new lint
            System.out.println();
            space=space+2;
        }

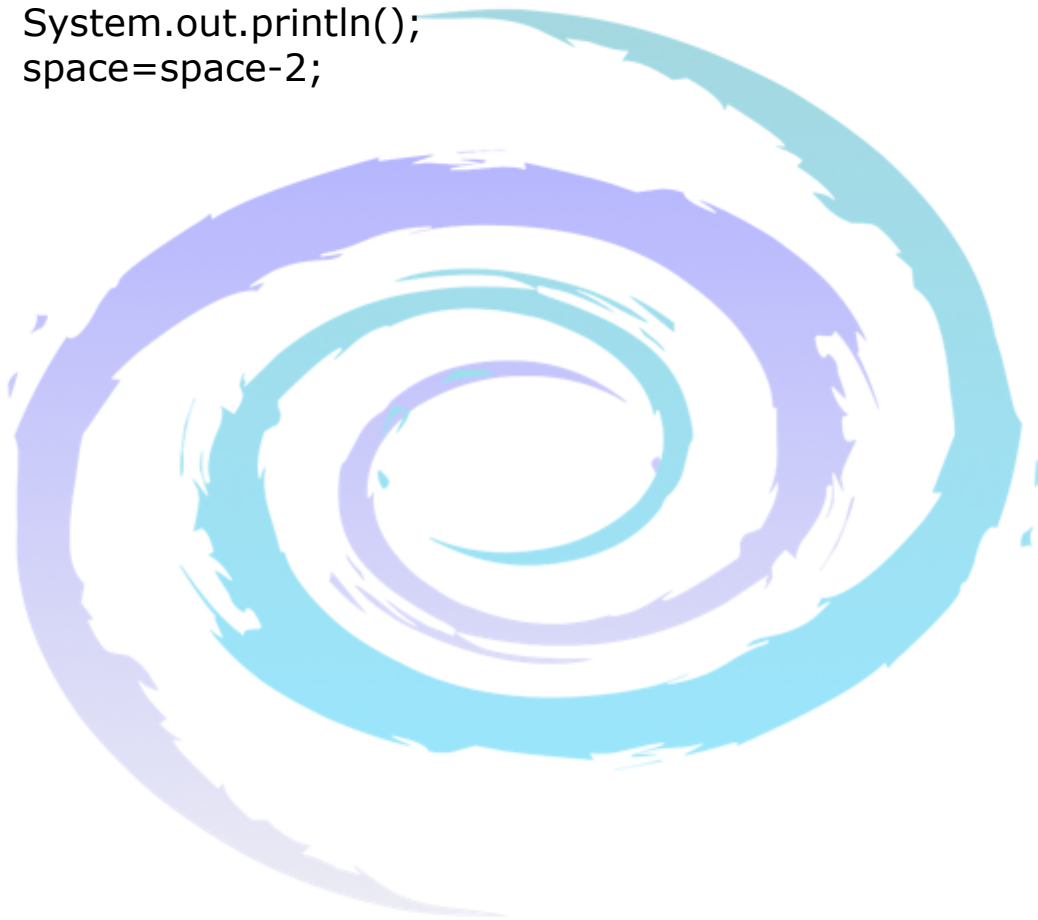
        space=8;
        for(int i=1; i<=5; i++)
        {
            //print first part of the row
            for(int j=1; j<=i; j++)
```

```
        System.out.print("*");

//print space
for(int j=1;j<=space; j++)
    System.out.print(" ");

//print second part of the row
for(int j=1; j<=i; j++)
    System.out.print("*");

//print new lint
System.out.println();
space=space-2;
    }
}
}
```



2. Write Java program to multiply two matrices

```
import java.util.Scanner;

class MatrixMultiplication
{
    public static void main(String args[])
    {
        int m, n, p, q, sum = 0, c, d, k;

        Scanner in = new Scanner(System.in);
        System.out.println("Enter the number of rows and columns of
first matrix");
        m = in.nextInt();
        n = in.nextInt();

        int first[][] = new int[m][n];

        System.out.println("Enter the elements of first matrix");

        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                first[c][d] = in.nextInt();

        System.out.println("Enter the number of rows and columns of
second matrix");
        p = in.nextInt();
        q = in.nextInt();

        if ( n != p )
            System.out.println("Matrices with entered orders can't be
multiplied with each other.");
        else
        {
            int second[][] = new int[p][q];
            int multiply[][] = new int[m][q];

            System.out.println("Enter the elements of second matrix");

            for ( c = 0 ; c < p ; c++ )
                for ( d = 0 ; d < q ; d++ )
                    second[c][d] = in.nextInt();
```

```
for ( c = 0 ; c < m ; c++ )
{
    for ( d = 0 ; d < q ; d++ )
    {
        for ( k = 0 ; k < p ; k++ )
        {
            sum = sum + first[c][k]*second[k][d];
        }

        multiply[c][d] = sum;
        sum = 0;
    }
}
```

```
System.out.println("Product of entered matrices:-");
```

```
for ( c = 0 ; c < m ; c++ )
{
    for ( d = 0 ; d < q ; d++ )
        System.out.print(multiply[c][d]+"t");

    System.out.print("\n");
}
}
```

3. Write a program which print Fibonacci series

```
#include <stdio.h>

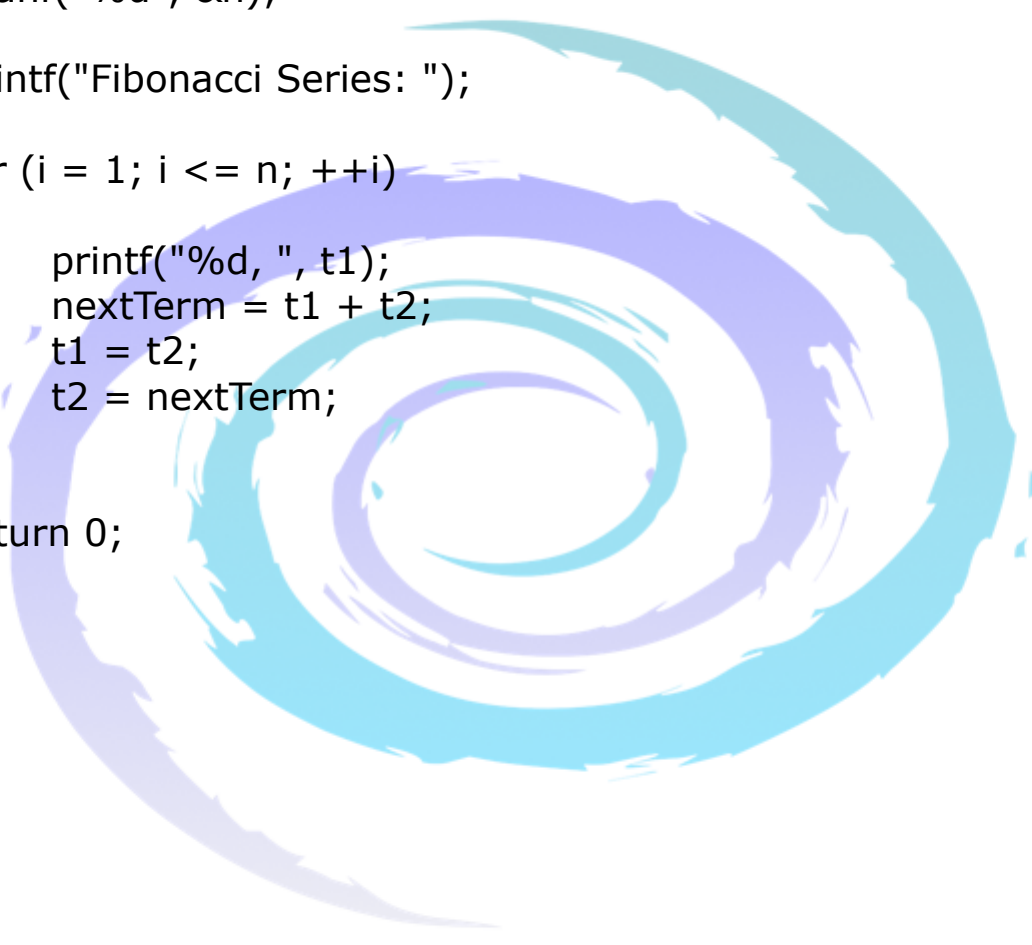
int main()
{
    int i, n, t1 = 0, t2 = 1, nextTerm;

    printf("Enter the number of terms: ");
    scanf("%d", &n);

    printf("Fibonacci Series: ");

    for (i = 1; i <= n; ++i)
    {
        printf("%d, ", t1);
        nextTerm = t1 + t2;
        t1 = t2;
        t2 = nextTerm;
    }

    return 0;
}
```



4. Java program to merge two files alternatively into third file

```
import java.io.*;

class FileMerge
{
    public static void main(String[] args) throws IOException
    {
        // PrintWriter object for file3.txt
        PrintWriter pw = new PrintWriter("file3.txt");

        // BufferedReader object for file1.txt
        BufferedReader br1 = new BufferedReader(new
        FileReader("file1.txt"));

        // BufferedReader object for file2.txt
        BufferedReader br2 = new BufferedReader(new
        FileReader("file2.txt"));

        String line1 = br1.readLine();
        String line2 = br2.readLine();

        // loop to copy lines of
        // file1.txt and file2.txt
        // to file3.txt alternatively
        while (line1 != null || line2 != null)
        {
            if(line1 != null)
            {
                pw.println(line1);
                line1 = br1.readLine();
            }

            if(line2 != null)
            {
                pw.println(line2);
                line2 = br2.readLine();
            }
        }

        pw.flush();

        // closing resources
        br1.close();
    }
}
```

```
br2.close();  
pw.close();
```

```
    System.out.println("Merged file1.txt and file2.txt  
alternatively into file3.txt");  
}  
}
```



5. Java program to count the number of characters in a file

```
import java.io.*;

public class Test
{
    public static void main(String[] args) throws IOException
    {
        File file = new File(path);
        FileInputStream fileStream = new FileInputStream(file);
        InputStreamReader input = new InputStreamReader(fileStream);
        BufferedReader reader = new BufferedReader(input);

        String line;

        // Initializing counters
        int countWord = 0;
        int sentenceCount = 0;
        int characterCount = 0;
        int paragraphCount = 1;
        int whitespaceCount = 0;

        // Reading line by line from the
        // file until a null is returned
        while((line = reader.readLine()) != null)
        {
            if(line.equals(""))
            {
                paragraphCount++;
            }
            if(!(line.equals(""))))
            {
                characterCount += line.length();

                // \s+ is the space delimiter in java
                String[] wordList = line.split("\\s+");

                countWord += wordList.length;
                whitespaceCount += countWord - 1;

                // [!?.:]+ is the sentence delimiter in java
                String[] sentenceList = line.split("[!?.:]+");
            }
        }
    }
}
```



```
        sentenceCount += sentenceList.length;
    }
}

System.out.println("Total word count = " + countWord);
    System.out.println("Total number of sentences = " +
sentenceCount);
        System.out.println("Total number of characters = " +
characterCount);
    System.out.println("Number of paragraphs = " + paragraphCount);
        System.out.println("Total number of whitespaces = " +
whitespaceCount);
    }
}
```

