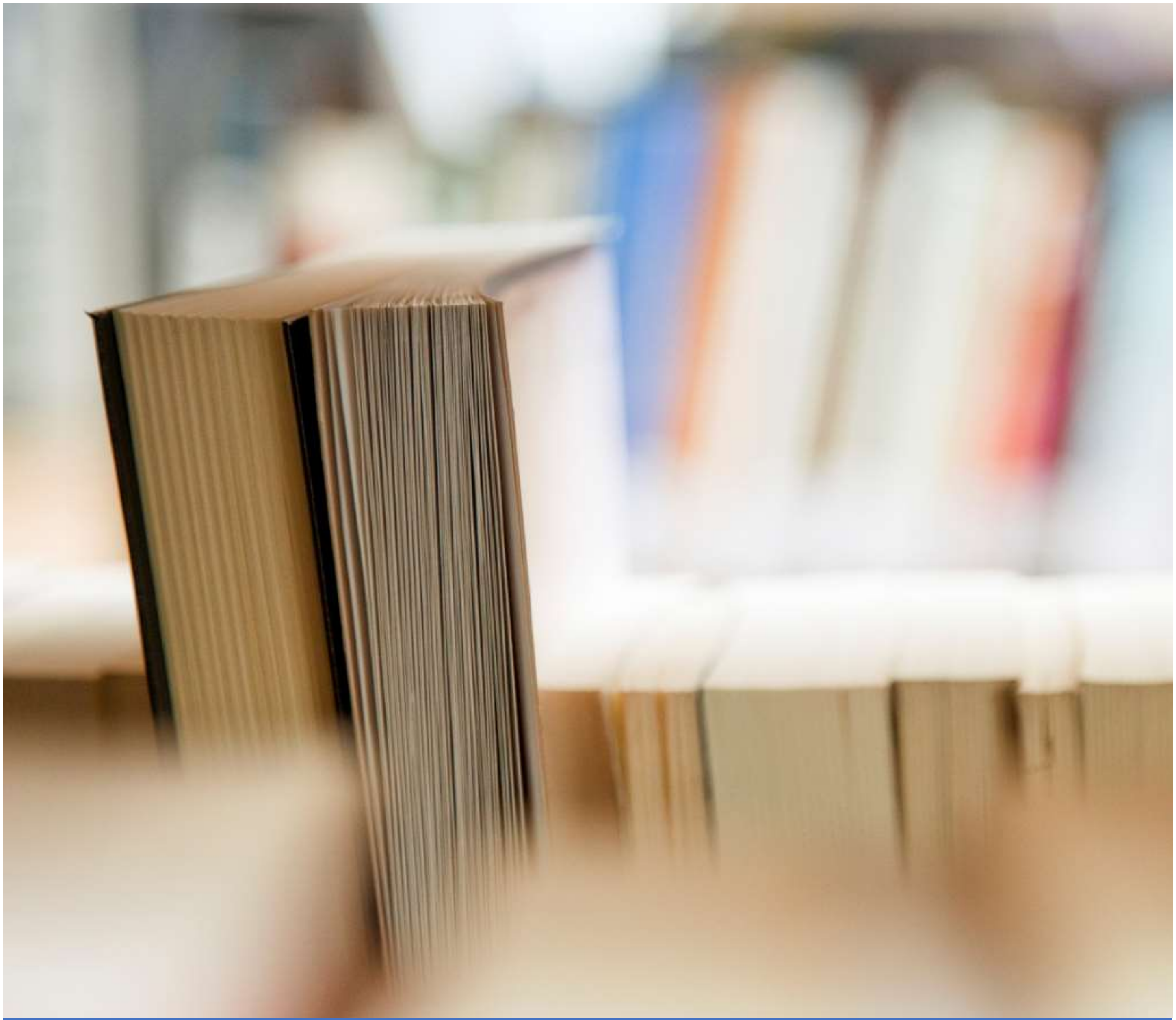


Code Logic

Ashirbad swain 8/31/20 SilentCoder

Write a Java Program to check the number is Armstrong or not?

A positive number is called Armstrong number is it equal to the sum of cubes of its digits.

It uses the concept of finding the length of a number and reverse a number.

Example: 153 = (1*1*1) +(5*5*5) +(3*3*3)
where:
(1*1*1) =1
(5*5*5) =125
(3*3*3) =27
So:
1+125+27=153

```
package ashirbad;
import java.util.Scanner;

public class ArmstrongNum
{
		public static void main(String args[])
		{
				Scanner s = new Scanner(System.in);
				System.out.println("Enter a number: ");
				int no = s.nextInt();
				int t1 = no;
				int leng = 0;
				while(t1!= 0)
				{
						t1 = t1/10;
						leng = leng + 1;
				}

				int t2 = no;
				int rem;
				int arm = 0;
				while(t2!= 0)
				{
						rem = t2%10;
						int mul = 1;
						for(int i = 1; i <= leng; i++)
						{
								mul = mul*rem;
						}
						arm = arm + mul;
						t2 = t2/10;
				}
				if(no == arm)
				{
						System.out.println(no+" is Armstrong");
				}
				else
				{
						System.out.println(no+" is not Armstrong");
```

```
                }
                s.close();
        }
}

Output:
Enter the number: 153
153 is Armstrong number
```

Write a Java program to check the given number is palindrome or not?

Palindrome number: A palindrome number is a number that is same after reverse.

For example: 545, 151, 343 etc. are palindrome number and it can also be sting like LOL, MADAM etc.

It uses the concept of reverse a number.

```
package ashirbad;
import java.util.Scanner;

public class Palindrome
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter a number : ");
                int num = sc.nextInt();
                int temp = num;
                int rem, rev = 0;
                while(temp!= 0)
                {
                        rem = temp % 10;
                        rev = rev * 10 + rem;
                        temp = temp/10;
                }
                if(num == rev)
                {
                        System.out.println(num+" is palindrome");
                }
                else
                {
                        System.out.println(num+" is not palindrome");
                }
                sc.close();
        }
}
```

```
Output: Enter the number: 121
        121 is palindrome number
```

Write program to check the number is perfect or not?

Perfect number: If the sum of its positive divisors excluding the number itself is equal to that number. For example, 28 is a perfect number because 28 is divisible by 1, 2, 4, 7, 14 and 28 and the sum of these values are: 1 + 2 + 4 + 7 + 14 = 28 (Remember, we have to exclude the number itself. That's why we haven't added 28 here). Some of the Java perfect numbers are 6, 28, 496, 8128 and 33550336 so on.

```
package ashirbad;
import java.util.Scanner;

public class Perfectum
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter a number: ");
                int num = sc.nextInt();
                int sum = 0;

                for(int i = 1; i < num; i++)
                {
                        if(num%i == 0)
                        {
                                sum = sum + i;
                        }
                }
                if(sum == num)
                {
                        System.out.println(num+" is perfect number");
                }
                else
                {
                        System.out.println(num+" is not perfect number");
                }
                sc.close();
        }
}
```

```
Output: Enter the number: 28
        28 is perfect number
```

Write a Java program to find prime number?

```java
package ashirbad;
import java.util.Scanner;
public class PrimeNum
{
        public static void main(String args[])
        {
                Scanner s = new Scanner(System.in);
                System.out.println("Enter a number : ");
                int num = s.nextInt();
                int temp = 0;

                for(int i = 2; i <= num-1; i++)
                {
                        if(num%i == 0)
                        {
                                temp = temp +1;
                        }
                }
                if(temp > 0)
                {
                        System.out.println(num+" is not prime number");
                }
                else
                {
                        System.out.println(num+" is prime number");
                }
                s.close();

        }
}
```

```
Output: Enter a number: 5
        5 is prime number.
```

Write a Java program to print fibonacci series?

Fibonacci Series: In fibonacci series, *next number is the sum of previous two numbers* for example 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 etc. The first two numbers of fibonacci series are 0 and 1.

It uses the concept of Swapping.

```
package ashirbad;
import java.util.Scanner;

public class Fibonacci
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the number : ");
                int num = sc.nextInt();
                int a = 0, b = 1;
                System.out.print(a+" "+b);

                int c;
                for(int i = 1; i <= num; i++)
                {
                        c = a + b;
                        System.out.print(" "+c);
                        a = b;
                        b = c;
                }
                sc.close();
        }
}
```

```
Output:  0 1 1 2 3 5 8 13 21 34 55 89
```

Write a Java program to print factorial of a number?

Factorial Program in Java: Factorial of n is the *product of all positive descending integers*. Factorial of *n* is denoted by n!

For example:

4! = 4*3*2*1 = 24

5! = 5*4*3*2*1 = 120

```
package ashirbad;
import java.util.Scanner;

public class Factorial
{
            public static void main(String args[])
            {
                        Scanner s = new Scanner(System.in);
                        System.out.println("Enter a number : ");
                        int num = s.nextInt();
                        int fact = 1;

                        for(int i = 1; i <= num; i++)
                        {
                                    fact = fact * i;
                        }
                        System.out.println("The factorial of "+num+" is " +fact);
                        s.close();
            }
}
```

```
Output:     Enter a Number: 5
                        Factorial of 5 is 120
```

Write a Java program to swapping two numbers using third variable and without using third variable?

```
//using third variable
package ashirbad;

public class SwappingOne
{
        public static void main(String args[])
        {
                int a = 10, b = 20;
                int temp;

                temp = a;
                a = b;
                b = temp;

                System.out.println("a : "+a);
                System.out.println("b : "+b);

        }
}
```

```
//without using third variable
package ashirbad;

public class SwappingTwo
{
        public static void main(String args[])
        {
                int a = 10, b = 20;

                a = a + b;
                b = a - b;
                a = a - b;

                System.out.println("a : "+a);
                System.out.println("b : "+b);
        }
}
```

```
Output:         a : 20
                b : 10
```

Write a Java program to check leap year?

To determine whether a year is a leap year, follow these steps:
1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
4. The year is a leap year (it has 366 days).
5. The year is not a leap year (it has 365 days).

```
package ashirbad;
import java.util.Scanner;

public class LeapYear
{

        public static void main(String[] args)
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the year : ");
                int year = sc.nextInt();

                if((year%400 == 0) || (year%4 == 0 && year%100!= 0))
                {
                        System.out.println(year + " is a leap year");
                }
                else
                {
                        System.out.println(year + " is not a leap year");
                }
                sc.close();
        }

}
```

```
Output: Enter the year:
        2013
        2013 is not a leap year
```

Write a Java program to reverse a number?

```java
package ashirbad;
import java.util.Scanner;

public class ReverseNum
{
		public static void main(String args[])
		{
				Scanner sc = new Scanner(System.in);
				System.out.println("Enter a number : ");
				int num = sc.nextInt();
				int rev = 0, rem;
				while(num!= 0)
				{
						rem = num%10;
						rev = rev * 10 + rem;
						num = num/10;
				}
				System.out.println("The reverse of the number is : "+rev);
				sc.close();
		}
}
```

```
Output: 321
```

Write a Java program to find power of any number?

Power of a number = $base^{exponent}$

Example:

In case of 2^3

The base number is 2
The exponent is 3
So, the power will be the result of 2*2*2

```
package ashirbad;

public class PowerOfNumber
{
        public static void main(String[] args)
        {
                int base = 2;
                int exponent = 3;
                int result = 1;

                while(exponent > 0)
                {
                        result = result * base;
                        exponent--;
                }
                System.out.println("The Power is : "+result);
        }

}
```

```
Output: 8
```

Write a Java program find the given number is even or odd?

```
package ashirbad;

public class EvenOdd
{
        public static void main(String[] args)
        {
                int num = 24;

                if(num%2 == 0)
                {
                        System.out.println(num+ " is even number");
                }
                else
                {
                        System.out.println(num+ " is odd number");
                }
        }

}
```

```
Output: no is even
```

Write a Java program to find the larger of two numbers?

```
package ashirbad;
import java.util.Scanner;

public class LargerTwo
{
        public static void main(String[] args)
        {
                Scanner s = new Scanner(System.in);
                System.out.println("Enter the first numbers : ");
                int a = s.nextInt();
                System.out.println("Enter the second number : ");
                int b = s.nextInt();

                if(a > b)
                {
                        System.out.println("a is greater");
                }
                else
                {
                        System.out.println("b is greater");
                }
                s.close();
        }

}
Output: b is greater
```

Write a Java program to find larger among 3 numbers?

```
package ashirbad;
import java.util.Scanner;

public class LargeThree
{
        public static void main(String[] args)
        {
                Scanner sc = new Scanner(System.in);

                System.out.println("Enter the first number : ");
                int a = sc.nextInt();
                System.out.println("Enter the second number : ");
                int b = sc.nextInt();
                System.out.println("Enter the Third number : ");
                int c = sc.nextInt();

                if(a > b && a > c)
                {
                        System.out.println("a is greater");
                }
                else if(b < a && b > c)
                {
                        System.out.println("b is greater");
                }
                else
                {
                        System.out.println("c is greater");
                }
                sc.close();
        }

}
```

```
Output: c is greater
```

Write a java program to take input from the user and print on the screen?

```
package ashirbad;

import java.util.Scanner;

public class UserInputScanner
{
        public static void main(String args[])
        {
                int roll_no;
                String name;
                String branch;

                Scanner sc = new Scanner(System.in);

                System.out.println("Enter your name: ");
                name = sc.nextLine();
                System.out.println("Your name is: "+name);

                System.out.println("Enter your branch: ");
                branch = sc.nextLine();
                System.out.println("Branch is: "+branch);

                System.out.println("Enter your roll number: ");
                roll_no = sc.nextInt();
                System.out.println("roll number is: "+roll_no);

                sc.close();
        }
}
```

```
Output:   Enter a String: Ashirbad
                You entered a string: Ashirbad
                Enter an Integer: 27
                You entered an integer: 27
                Enter a float: 3.5
                You entered a float: 3.5
```

Write a java program to calculate the length of a number?

```
package ashirbad;
import java.util.Scanner;

public class LengthOfNumber
{
        public static void main(String args[])
        {
                Scanner s = new Scanner(System.in);
                System.out.println("Enter a number : ");
                int num = s.nextInt();
                int leng = 0;
                while(num!= 0)
                {
                        num = num/10;
                        leng = leng + 1;
                }
                System.out.println("The length of the number is : "+leng);
                s.close();

        }
}
```

```
Output:
Enter a number :

456

The length of the number is : 3
```

String programs

Write a Java program to print the second last word of the string?

Means that is if the input given string is hello world, then the output is world hello. That is the second word will be printed first and then first word will be printed at last.

```
package StringPrograms;

public class SecondLastWordString
{
        public static void main(String args[])
        {
                String testcase = "Hello Java";
                String input[] = testcase.split(" ");
                String lastword = input[input.length - 1];
                String firstword = input[input.length - 2];

                System.out.println(lastword +" "+ firstword);
        }
}
```

```
Output: World Hello
```

Write a Java program to reverse a string?

```
package StringPrograms;
import java.util.Scanner;

public class ReverseString
{
        public static void main(String[] args)
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter a word : ");
                String str = sc.nextLine();
                int leng = str.length();
                String rev = "";

                for(int i = leng-1; i >= 0; i--)
                {
                        rev = rev + str.charAt(i);
                }
                System.out.println("The reverse of "+str+" is "+rev);
                sc.close();
        }

}
```

```
Output: Reverse of Ashirbad is dabrihsA
```

Write a Java program to check two strings are anagrams or not?

Two strings are said to be anagrams if they contain same character but in different order.

Example: JAVA – AVAJ, LISTEN – SILENT

Procedure:

- Take two string as input
- Then convert the string to character array
- Then sort the array
- After sorting if the two string are equal, then two strings are anagram.

```
package StringPrograms;
import java.util.Arrays;

public class StringAnagram
{
        public static void main(String[] args)
        {
                //Declare two string
                String str1 = "Brag";
                String str2 = "Grab";

                //Converting the string into lower case
                str1 = str1.toLowerCase();
                str2 = str2.toLowerCase();

                //Checking for length of the string
                if(str1.length()!=str2.length())
                {
                        System.out.println("Both the strings are not anagram");
                }
                else
                {
                        //Converting the string into character array
                        char[] string1 = str1.toCharArray();
                        char[] string2 = str2.toCharArray();

                        //Sorting the arrays using in-built function sort()
                        Arrays.sort(string1);
                        Arrays.sort(string2);

                        //compare both sorted arrays using function equals()
                        if(string1.equals(string2))
                        {
                                System.out.println("Both the strings are anagram");
                        }
                        else
                        {
                                System.out.println("Both the strings are not anagram");
                        }
                        }}}
```

Write a Java Program to compare two strings?

We use equals () method to compare two strings but we can't use == for content comparison because it is used to address/reference comparison.

.equals(b) is a pre-defined method of Java String Class which checks whether two given and initialized strings are equal or not.

Again, it is to be noted that the comparison is *case-sensitive*. So to perform a comparison which ignores case differences, you have to use *equalsIgnoreCase()*method. As it compares two strings, it considers A-Z to be the same as a-z.

```
package StringPrograms;
import java.util.Scanner;

public class StringComparison
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);

                System.out.println("Enter the first string : ");
                String str1 = sc.nextLine();
                System.out.println("Enter the second string : ");
                String str2 = sc.nextLine();

                if(str1.equals(str2))//(str1.equalsIgnoreCase(b))
                {
                        System.out.println("Both strings are equal");
                }
                else
                {
                        System.out.println("Both strings are not equal");
                }
                sc.close();
        }

}
```

```
Output:  Enter the first string :
                ashirbad
                Enter the second string :
                ASHIRBAD
                Both strings are equal
```

Write a java program to remove white space in a given string?

replaceAll() method is used to remove and replace all whitespaces in the string *sentence*.

We've used regular expression \\s that finds all white space characters (tabs, spaces, new line character, etc.) in the string. Then, we replace it with "" (empty string literal).

```
package StringPrograms;

public class RemoveWhiteSpaceString
{

        public static void main(String[] args)
        {
                String str = "My name is Ashirbad Swain";
                System.out.println("Original Sentence : " + str);

                str = str.replaceAll("//s", "");
                System.out.println("After Replacement : " + str);
        }

}
```

```
Output:  Original sentence: My name is Ashirbad Swain
                 After replacement: MynameisAshirbadSwain
```

Write a Java program to check the given string is palindrome or not?

Palindrome is a string, which when read in both forward and backward way is same or of original string and reverse string are same as per alphabetical order then we say string is palindrome.

Example: madam, lol, pop etc.

Procedure:

- Input a string
- Find the reverse of a string
- Check the original string and reverse string are same or not
- If both are equal print palindrome else print given string is not palindrome.

```
package StringPrograms;
import java.util.Scanner;

public class StringPalindrome
{

        public static void main(String[] args)
        {
                System.out.println("Enter the input string : ");
                Scanner sc = new Scanner(System.in);
                String original = sc.nextLine();
                String reverse = "";

                for(int i = original.length()-1; i >= 0; i--)
                {
                        reverse = reverse + original.charAt(i);
                }
                if(original.equals(reverse))
                {
                        System.out.println("String Palindrome");
                }
                else
                {
                        System.out.println("not String Palindrome");
                }
                sc.close();
        }

}
```

```
Output:   Enter the input string: madam
                  Given string is palindrome.
```

Write a Java program to find duplicate character from the string?

- Here in this program, a Java class name *DuplStr* is declared which is having the main() method. All Java program needs one main() function from where it starts executing program. Inside the main(), the String type variable name *str* is declared and initialized with string *w3schools*. Next an integer type variable *cnt* is declared and initialized with value *0*. This cnt will count the number of character-duplication found in the given string.
- The statement: *char [] inp = str.toCharArray();* is used to convert the given string to character array with the name *inp* using the predefined method *toCharArray()*. The *System.out.println* is used to display the message "Duplicate Characters are as given below:". Now the for loop is implemented which will iterate from zero till string length. Another nested for loop has to be implemented which will count from *i+1* till length of string.
- Inside these two nested structures for loops, you have to use an if condition which will check whether *inp[i]* is equal to *inp[j]* or not. If the condition becomes true prints inp[j] using System.out.println() with a single incrementation of variable *cnt* and then *break* statement will be encountered which will move the execution out of the loop.

```
package StringPrograms;
public class DuplicateCharacter
{

        public static void main(String[] args)
        {
                String str = "asdfasd";
                char[] inp = str.toCharArray();
                System.out.println("Duplicate Characters are : ");

                for(int i = 0; i <= str.length(); i++)
                {
                        for(int j = i+1; j <= str.length()-1; j++)
                        {
                                if(inp[i] == inp[j])
                                {
                                        System.out.println(inp[j]);
                                        break;
                                }
                        }
                }
        }

}
Output: Duplicate Characters are : a s d
```

Write a Java program to find no of words in a string?

Procedure:

- Input a string
- Then put it in an array through split () method and count the number of words by string method.

```
package StringPrograms;

public class CountWord
{
        public static void main(String args[])
        {
                String x = "my name is ashirbad swain";
                String y[] = x.split(" ");

                System.out.println(y.length);

        }
}
```

```
Output: 5
```

Write a Java program to find no of character present in the given string?

```
package StringPrograms;

public class CountCharacter
{

        public static void main(String[] args)
        {
                String s = "my name is ashirbad swain";
                int count = 0;

                for(int i = 0; i < s.length(); i++)
                {
                        if(s.charAt(i)!=' ')
                        {
                                count++;
                        }
                }
                System.out.println("Total number of character in the string : "+count);
        }

}
```

```
Output: Total number of character in the string : 21
```

Write a Java program to find no of vowel and consonants presents in the given string ?

```
package StringPrograms;

public class countVC
{

        public static void main(String[] args)
        {
                // counter variable to store the count of vowels and consonants
                int vCount = 0, cCount = 0;
                // declare a string
                String str = "my name is ashirbad";
                // converting entire string to lower case to reduce the comparisons
                str = str.toLowerCase();

                for(int i = 0; i < str.length(); i++)
                {
                        //check whether a character is a vowel
                        if(str.charAt(i) == 'a'|| str.charAt(i) == 'e' || str.charAt(i) == 'i' ||
str.charAt(i) == 'o' || str.charAt(i) == 'u')
                        {
                                //increments the vowel counter
                                vCount++;
                        }
                        //check whether a character is a consonant
                        else if(str.charAt(i) >= 'a' && str.charAt(i) <= 'z')
                        {
                                //increment the consonant counter
                                cCount++;
                        }
                }
                System.out.println("Number of vowels : "+ vCount);
                System.out.println("Number of consonants : "+ cCount);
        }

}
```

```
Output:   Number of vowels : 6
                Number of consonants : 10
```

Write a Java program to find the frequency of a given character in the given string?

```java
package StringPrograms;

public class FrequencyCharacter
{

        public static void main(String[] args)
        {
                String str = "this website is awesome";
                char ch = 't';
                int frequency = 0;

                for(int i = 0; i < str.length(); i++)
                {
                        if(ch == str.charAt(i))
                        {
                                ++frequency;
                        }
                }
                System.out.println("Frequency of " + ch + " = " + frequency);
        }

}
```

```
Output: Frequency of t = 2
```

Array Programs

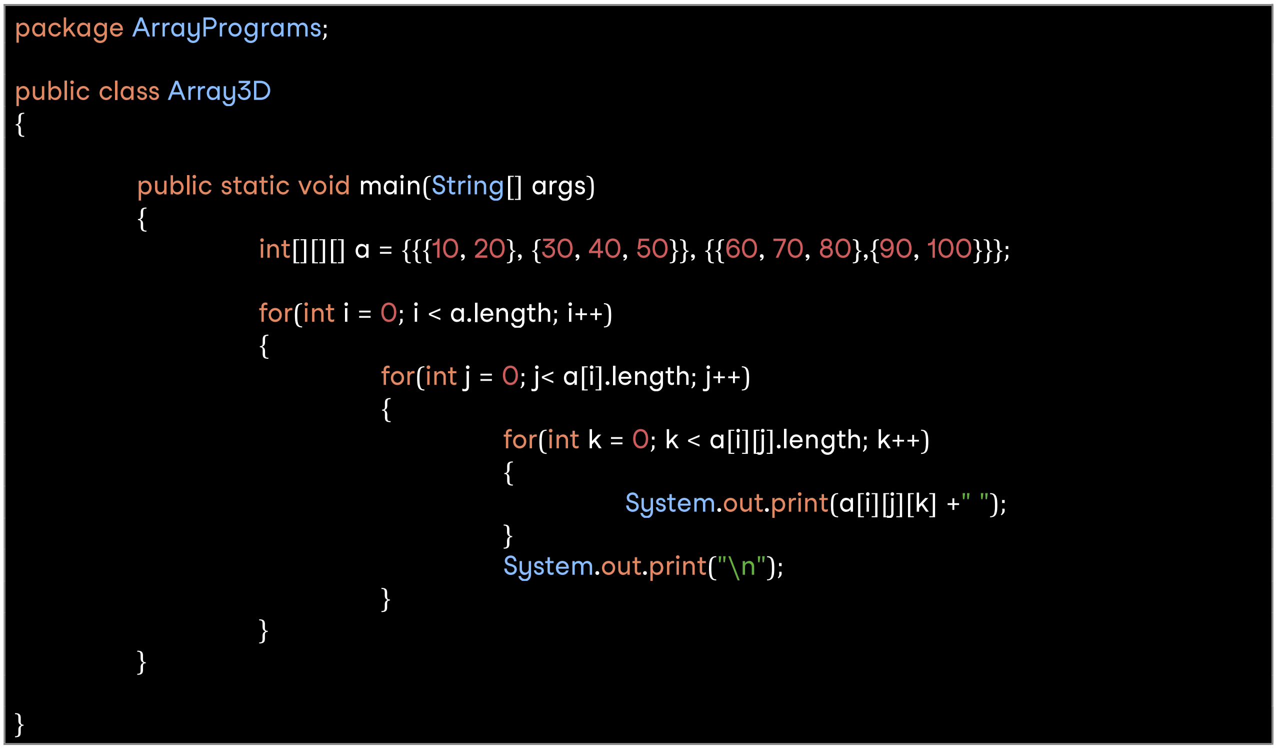
Write a Java program to print the elements of array?

```
package ArrayPrograms;

public class Array1D
{

    public static void main(String[] args)
    {
        int a[] = {10, 20, 30};

        for(int i = 0; i < a.length; i++)
        {
            System.out.print(a[i]+ " ");
        }
    }

}
```

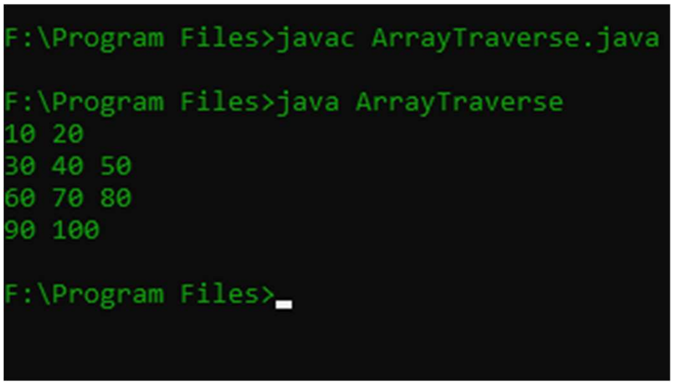
```
Output: 10 20 30
```

```
package ArrayPrograms;

public class Array2D
{

    public static void main(String[] args)
    {
        int a[][] = {{10, 20},{30, 40 ,50}};

        for(int i = 0; i < a.length; i++)
        {
            for(int j = 0; j < a[i].length; j++)
            {
                System.out.print(a[i][j]+ " ");
            }
            System.out.print("\n");
        }
    }

}
```

```
Output:   10 20
          30 40 50
```



```java
package ArrayPrograms;

public class Array3D
{

        public static void main(String[] args)
        {
                int[][][] a = {{{10, 20}, {30, 40, 50}}, {{60, 70, 80},{90, 100}}};

                for(int i = 0; i < a.length; i++)
                {
                        for(int j = 0; j< a[i].length; j++)
                        {
                                for(int k = 0; k < a[i][j].length; k++)
                                {
                                        System.out.print(a[i][j][k] +" ");
                                }
                                System.out.print("\n");
                        }
                }
        }

}
```



```
F:\Program Files>javac ArrayTraverse.java

F:\Program Files>java ArrayTraverse
10 20
30 40 50
60 70 80
90 100

F:\Program Files>_
```

Finding maximum and minimum element in the given list of arrays using java ?

```
package ArrayPrograms;

public class SearchMax
{

    public static void main(String[] args)
    {
        int a[] = {2, 5, 9, 12, 56, 78};
        int max = a[0];

        for(int i = 0; i < a.length; i++)
        {
            if(a[i] > max)
            {
                max = a[i];
            }

        }

        System.out.println("The larggest element is : "+ max);

    }
```

```
Output: The largest element is : 78
```

```
package ArrayPrograms;

public class SearchMin
{

    public static void main(String[] args)
    {
        int a[] = {3, 5, 1 , 12, 45, 23};
        int min = a[0];

        for(int i = 0; i < a.length; i++)
        {
            if(min > a[i])
            {
                min = a[i];
            }
        }

        System.out.println("The minimum element is : "+min);
    }

}
 Output: The minimum element is:1
```

Ashirbad Swain

Find the second largest element and second smallest element in the array list

```
package ArrayPrograms;

public class SecondLargest
{

    public static void main(String[] args)
    {
        int a[] = {6, 8, 2, 4, 3, 1, 5, 7};
        int temp;

        for(int i  = 0; i < a.length; i++)
        {
            for(int j = i+1; j < a.length; j++)
            {
                if(a[i] < a[j])
                {
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }

        System.out.println("Second largest element is : " +a[1]);
    }

}
```

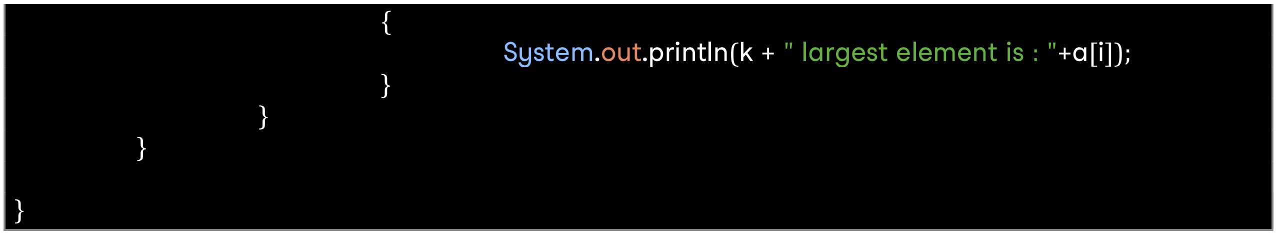
```
Output: Second largest element is : 7
```

```
package ArrayPrograms;
public class SecondSmallest
{

    public static void main(String[] args)
    {
        int a[] = {1, 8, 5, 3, 13};
        int temp;

        for(int i = 0; i < a.length; i++)
        {
            for(int j = i+1; j < a.length; j++)
            {
                if(a[i] > a[j])
                {
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
        System.out.println("Second smallest element is : "+a[1]);
    }
```

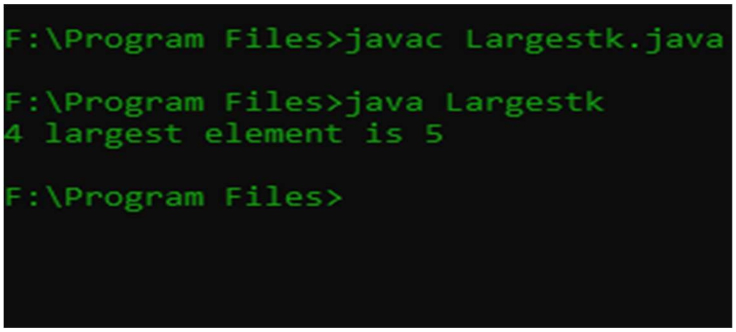
```
Output: Second smallest element is: 3
```

Find the largest and smallest element in the k^{th} position in array

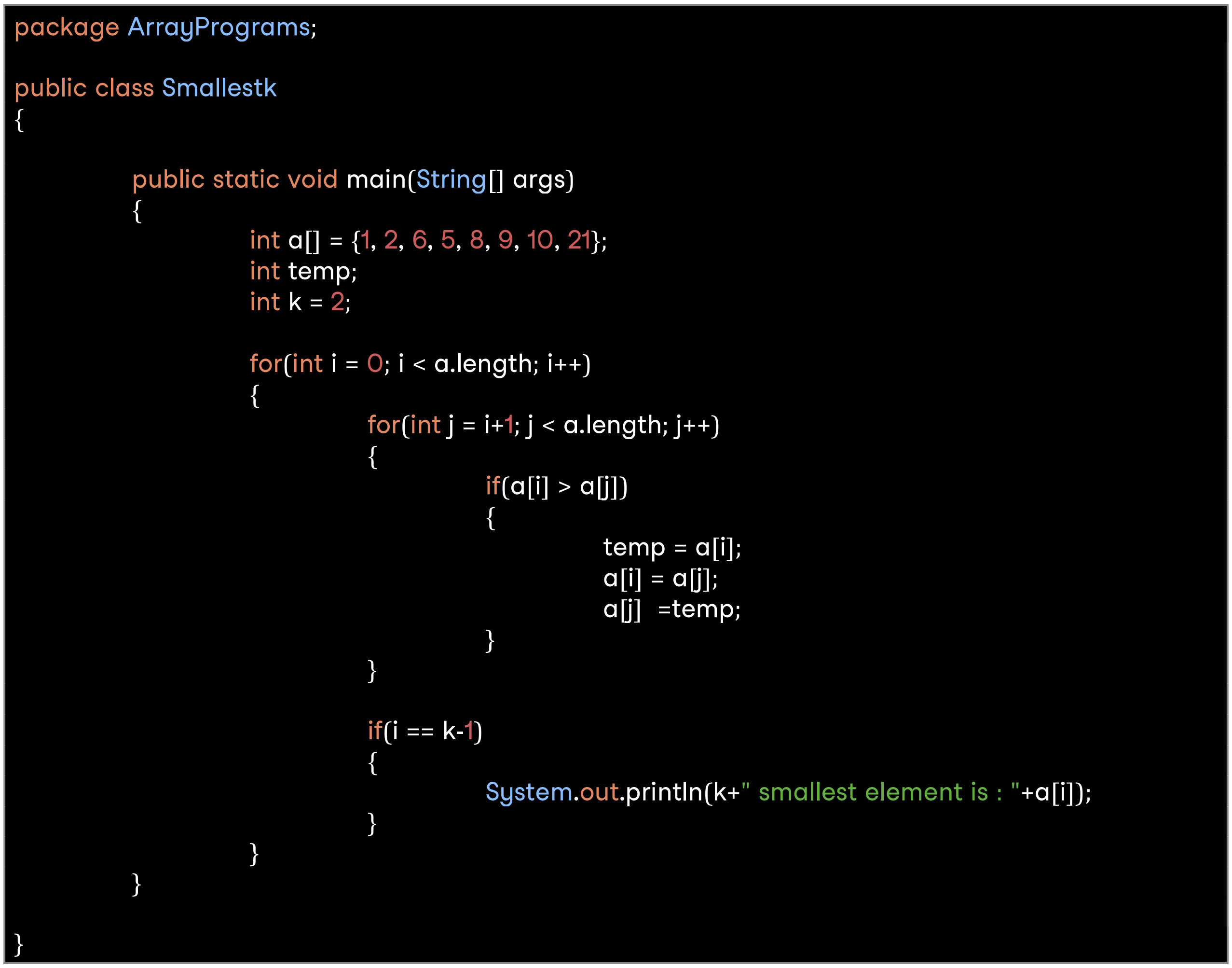
```
package ArrayPrograms;

public class Largestk
{

    public static void main(String[] args)
    {
        int a[] = {4, 5, 9 , 6 , 12, 45};
        int temp;
        int k = 4;

        for(int i = 0; i < a.length; i++)
        {
            for(int j = i+1; j < a.length; j++)
            {
                if(a[i] < a[j])
                {
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }

            if(i == k-1)
```



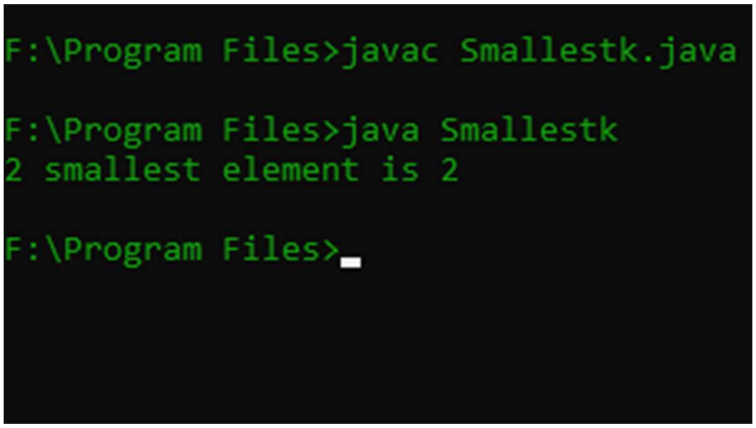
```
            {
                System.out.println(k + " largest element is : "+a[i]);
            }
        }
    }

}
```



```
F:\Program Files>javac Largestk.java

F:\Program Files>java Largestk
4 largest element is 5

F:\Program Files>
```



```
package ArrayPrograms;

public class Smallestk
{

    public static void main(String[] args)
    {
        int a[] = {1, 2, 6, 5, 8, 9, 10, 21};
        int temp;
        int k = 2;

        for(int i = 0; i < a.length; i++)
        {
            for(int j = i+1; j < a.length; j++)
            {
                if(a[i] > a[j])
                {
                    temp = a[i];
                    a[i] = a[j];
                    a[j]  =temp;
                }
            }

            if(i == k-1)
            {
                System.out.println(k+" smallest element is : "+a[i]);
            }
        }
    }

}
```



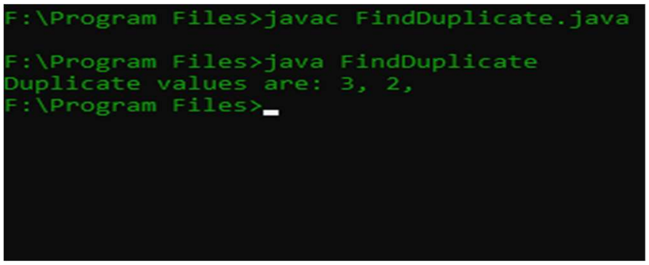
```
F:\Program Files>javac Smallestk.java

F:\Program Files>java Smallestk
2 smallest element is 2

F:\Program Files>_
```

Finding duplicate elements from the given array

By Brute force method and Set Interface method

Disadvantage: if any element appears more than 2 times it will print again the same element.

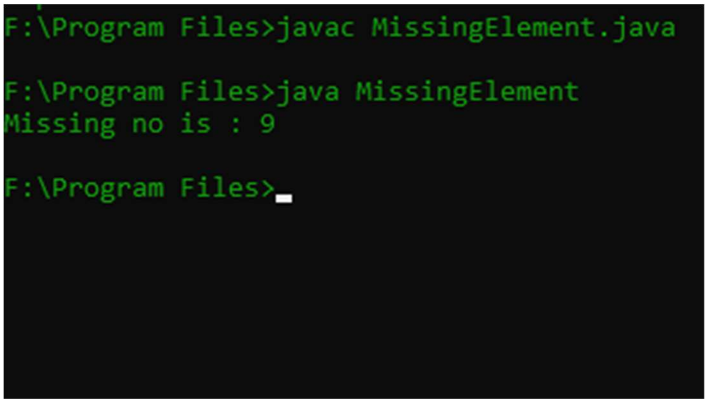
```
// by Brute Force method
package ArrayPrograms;

public class FindDuplicate
{

            public static void main(String[] args)
            {
                        int a[] = {3, 5 , 4, 3 , 2 , 2, 1};
                        System.out.println("Duplicate values are : ");

                        for(int i = 0; i < a.length-1; i++)
                        {
                                    for(int j = i+1; j < a.length; j++)
                                    {
                                                if((a[i] == a[j]) && (i!= j))
                                                {
                                                            System.out.println(a[j]+" ");
                                                }
                                    }
                        }
            }

}
```



```
F:\Program Files>javac FindDuplicate.java

F:\Program Files>java FindDuplicate
Duplicate values are: 3, 2,
F:\Program Files>_
```

Find the missing number in an array

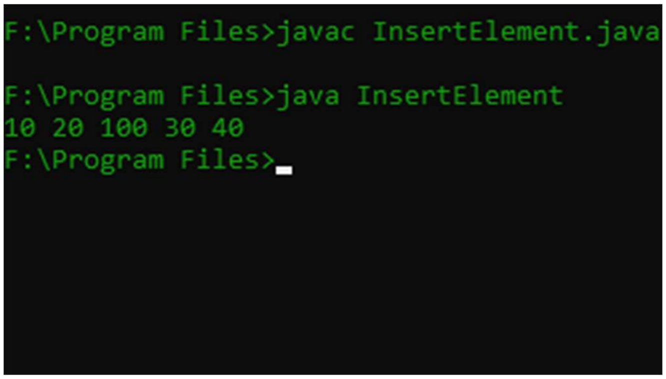
```
package ArrayPrograms;

public class FindMissingElement
{

        public static void main(String[] args)
        {
                int a[] = {1,2,3,4,5,6,7,8,10};
                int expected_no_elements = a.length+1;
                int total_sum = expected_no_elements*(expected_no_elements+1)/2;
                int sum = 0;

                for(int i = 0; i < a.length; i++)
                {
                        sum = sum + a[i];
                }

                System.out.println("Missing number is : "+(total_sum - sum));
        }

}
```



```
F:\Program Files>javac MissingElement.java

F:\Program Files>java MissingElement
Missing no is : 9

F:\Program Files>_
```

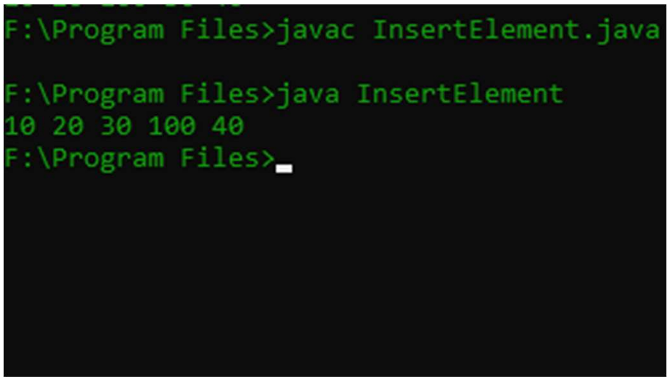
How to insert an element in the array

```
// insert element in given position

package ArrayPrograms;

public class InsertElement1
{

        public static void main(String[] args)
        {
                int a[] = {10,20,30,40,50};

                int pos = 3;
                int element = 100;

                for(int i = a.length-1; i > pos-1; i--)
                {
                        a[i] = a[i-1];
                }
                a[pos-1] = element;
                for(int i = 0; i < a.length; i++)
                {
                        System.out.println(a[i]+" ");
                }
        }

}
```



```
F:\Program Files>javac InsertElement.java

F:\Program Files>java InsertElement
10 20 100 30 40
F:\Program Files>_
```

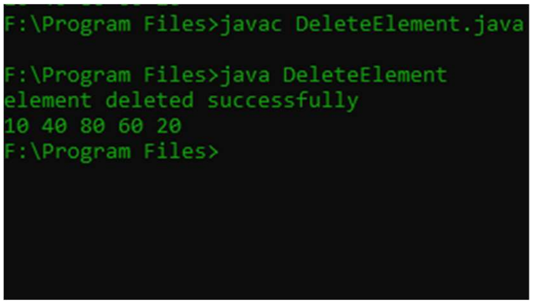
```java
// insert element in given index position

package ArrayPrograms;

public class InsertElement2
{

		public static void main(String[] args)
		{
				int a[] = {10,20,30,40,50};
				int index_pos = 3;
				int element = 100;

				for(int i = a.length-1; i > index_pos; i--)
				{
						a[i] = a[i-1];
				}

				a[index_pos] = element;
				for(int i = 0; i < a.length; i++)
				{
						System.out.println(a[i]+" ");
				}
		}

}
```



```
F:\Program Files>javac InsertElement.java

F:\Program Files>java InsertElement
10 20 30 100 40
F:\Program Files>_
```

How to delete a specific element from the array

```
package ArrayPrograms;

public class DeleteElement
{

        public static void main(String[] args)
        {
                int a[] = {10,40,30,80,60,20};
                int del_ele = 30;
                int count = 0;

                for(int i = 0; i < a.length; i++)
                {
                        if(del_ele == a[i])
                        {
                                for(int j = i; j < a.length-1; j++)
                                {
                                        a[j] = a[j+1];
                                }
                                count = count + 1;
                                break;
                        }
                }

                if(count == 0)
                {
                        System.out.println("Element not found");
                }
                else
                {
                        System.out.println("Element deleted successfully");
                        for(int i = 0; i < a.length-1; i++)
                        {
                                System.out.println(a[i]+" ");
                        }
                }
        }

}
```

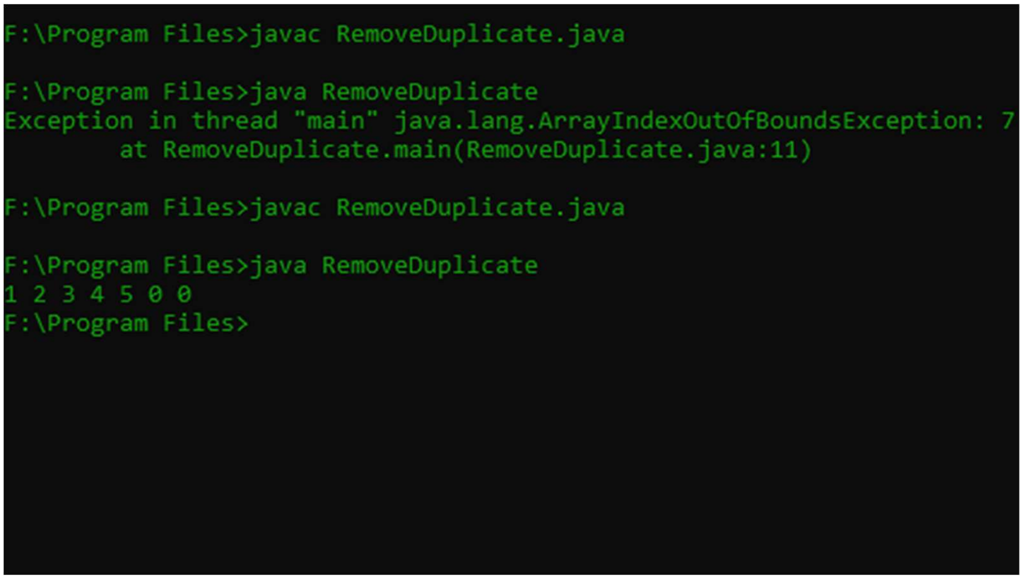


```
F:\Program Files>javac DeleteElement.java

F:\Program Files>java DeleteElement
element deleted successfully
10 40 80 60 20
F:\Program Files>
```

Ashirbad Swain

Remove Duplicate elements from array

```java
package ArrayPrograms;

public class RemoveDuplicate
{

    public static void main(String[] args)
    {
        int a[] = {1,2,2,3,4,5,5};
        int[] temp = new int[a.length];
        int j = 0;

        for(int i = 0; i < a.length-1; i++)
        {
            if(a[i]!= a[i+1])
            {
                temp[j] = a[i];
                j++;
            }
        }

        temp[j] = a[a.length-1];
        for(int i = 0; i < a.length; i++)
        {
            System.out.println(temp[i]+" ");
        }
    }

}
```



```
F:\Program Files>javac RemoveDuplicate.java

F:\Program Files>java RemoveDuplicate
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 7
        at RemoveDuplicate.main(RemoveDuplicate.java:11)

F:\Program Files>javac RemoveDuplicate.java

F:\Program Files>java RemoveDuplicate
1 2 3 4 5 0 0
F:\Program Files>
```

Sort the array elements in ascending order

```
package ArrayPrograms;

public class SortAscending
{

    public static void main(String[] args)
    {
        //Initialize array
        int[] arr = new int[] {5,2,7,4,1};
        int temp = 0;

        //Displaying elements of original array
        System.out.println("Elements of the original array : ");

        for(int i = 0; i < arr.length; i++)
        {
            System.out.println(arr[i]+" ");
        }

        //Sort the array in ascending order
        for(int i = 0; i < arr.length; i++)
        {
            for(int j = i+1; j < arr.length; j++)
            {
                if(arr[i] > arr[j])
                {
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }

        //Displaying elements of array after sorting
        System.out.println("Elements of sorted array in ascending order : ");
        for(int i = 0; i < arr.length; i++)
        {
            System.out.println(arr[i]+" ");
        }
    }

}
```

```
Output:
Elements of the original array :
5 2 7 4 1
Elements of sorted array in ascending order :
1 2 4 5 7
```

Sort the array elements in descending order

```
package ArrayPrograms;

public class SortDescending
{

        public static void main(String[] args)
        {
                //Initialize array
                int[] arr = new int[] {5,2,7,4,1};
                int temp = 0;

                //Displaying elements of original array
                System.out.println("Elements of the original array : ");

                for(int i = 0; i < arr.length; i++)
                        {
                                System.out.println(arr[i]+" ");
                        }

                //Sort the array in descending order
                for(int i = 0; i < arr.length; i++)
                        {
                                for(int j = i+1; j < arr.length; j++)
                                {
                                        if(arr[i] < arr[j])
                                        {
                                                temp = arr[i];
                                                arr[i] = arr[j];
                                                arr[j] = temp;
                                        }
                                }
                        }

                    //Displaying elements of array after sorting
                        System.out.println("Elements of sorted array in descending order : ");
                for(int i = 0; i < arr.length; i++)
                        {
                                System.out.println(arr[i]+" ");
                        }
        }

}
```

```
Output:
Elements of the original array :
5 2 7 4 1
Elements of sorted array in descending order :
7 5 4 2 1
```

Ashirbad Swain

Sum of the array element

```
package ArrayPrograms;

public class SumofArray
{

        public static void main(String[] args)
        {
                //Initialize array
                int[] arr = new int[] {1,2,3,4,5};
                int sum = 0;

                //Loop through the array to calculate sum of elements
                for(int i = 0; i < arr.length; i++)
                {
                        sum = sum + arr[i];
                }
                System.out.println("Sum of all the elements of array : "+ sum);
        }

}
```

```
Output:
Sum of all the elements of array : 15
```

Pattern Program

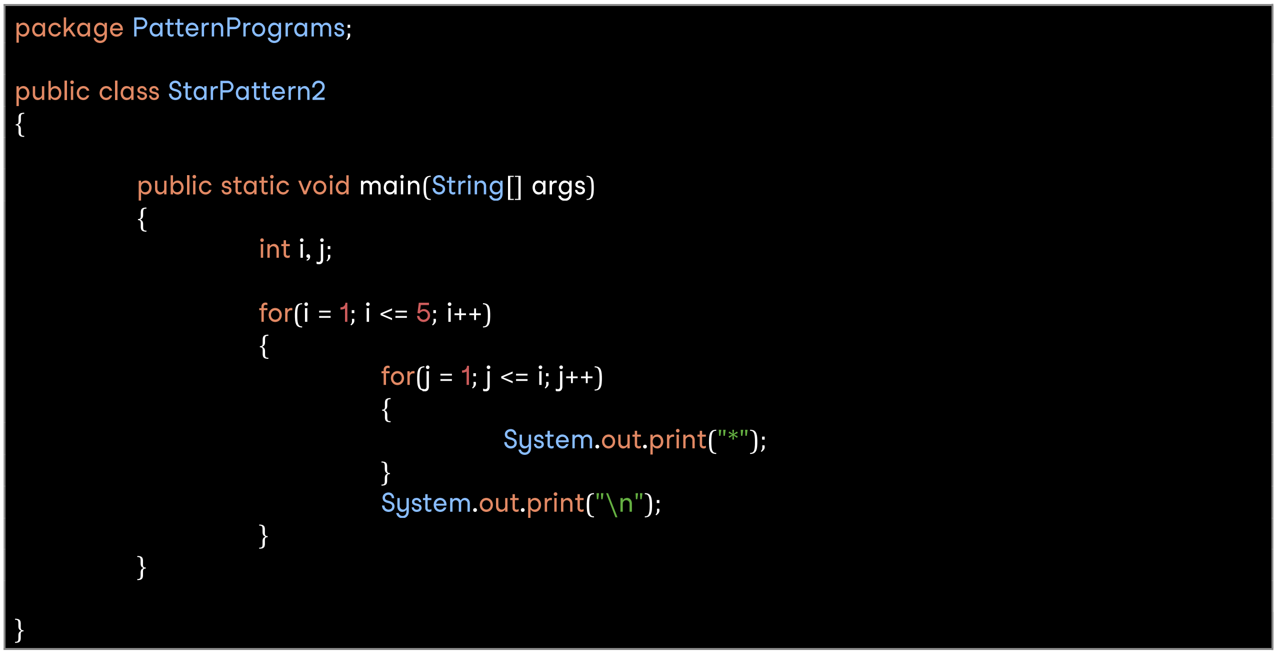
Write a Java Program to print different star pattern?

- Print ():
    - Print () method in Java is used to display a text on the console. This text is passed as parameter to this method in the form of string. This method prints the text on the console and cursor remains at the end of the text at the console. The next printing takes place from just here.
- Println ();
    - Println () method in Java is also used to display a text on the console. This text is passed as parameter to this method in the form of string. This method prints the next on the console and the cursor remains at the start of the next line at the console. The next printing takes place from next line.

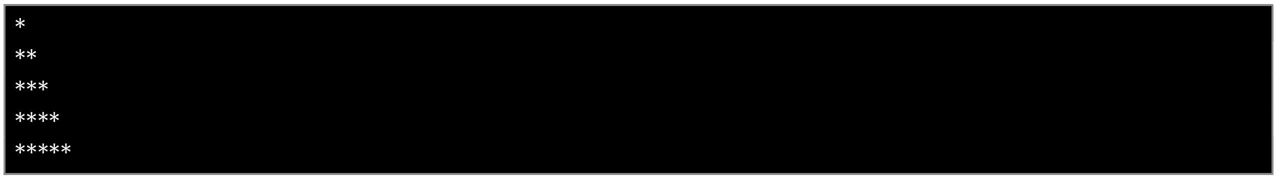
Star Patterns:

```
package PatternPrograms;

public class StarPattern1
{

    public static void main(String[] args)
    {
        int i, j;
        for(i = 1; i <= 5; i++)
        {
            for(j = 1; j <= 5; j++)
            {
                System.out.print("*");
            }
            System.out.print("\n");
        }
    }

}
```

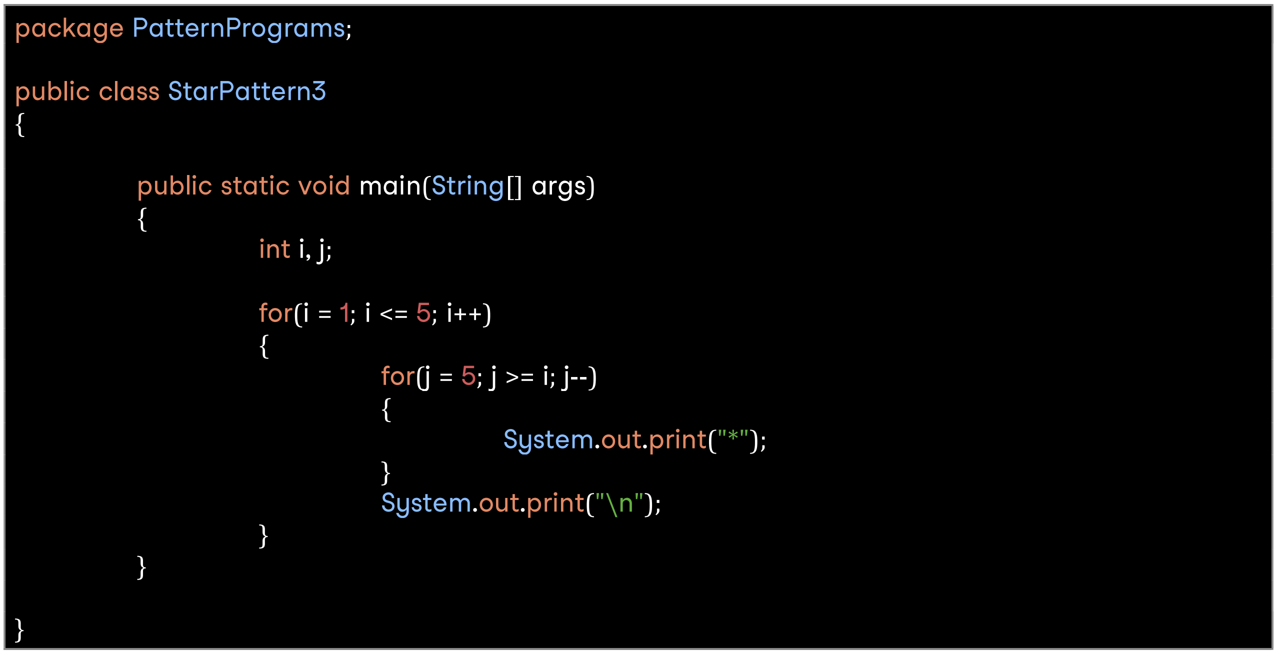
```
Output:
*****
*****
*****
*****
*****
```



```java
package PatternPrograms;

public class StarPattern2
{

    public static void main(String[] args)
    {
        int i, j;

        for(i = 1; i <= 5; i++)
        {
            for(j = 1; j <= i; j++)
            {
                System.out.print("*");
            }
            System.out.print("\n");
        }
    }

}
```



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

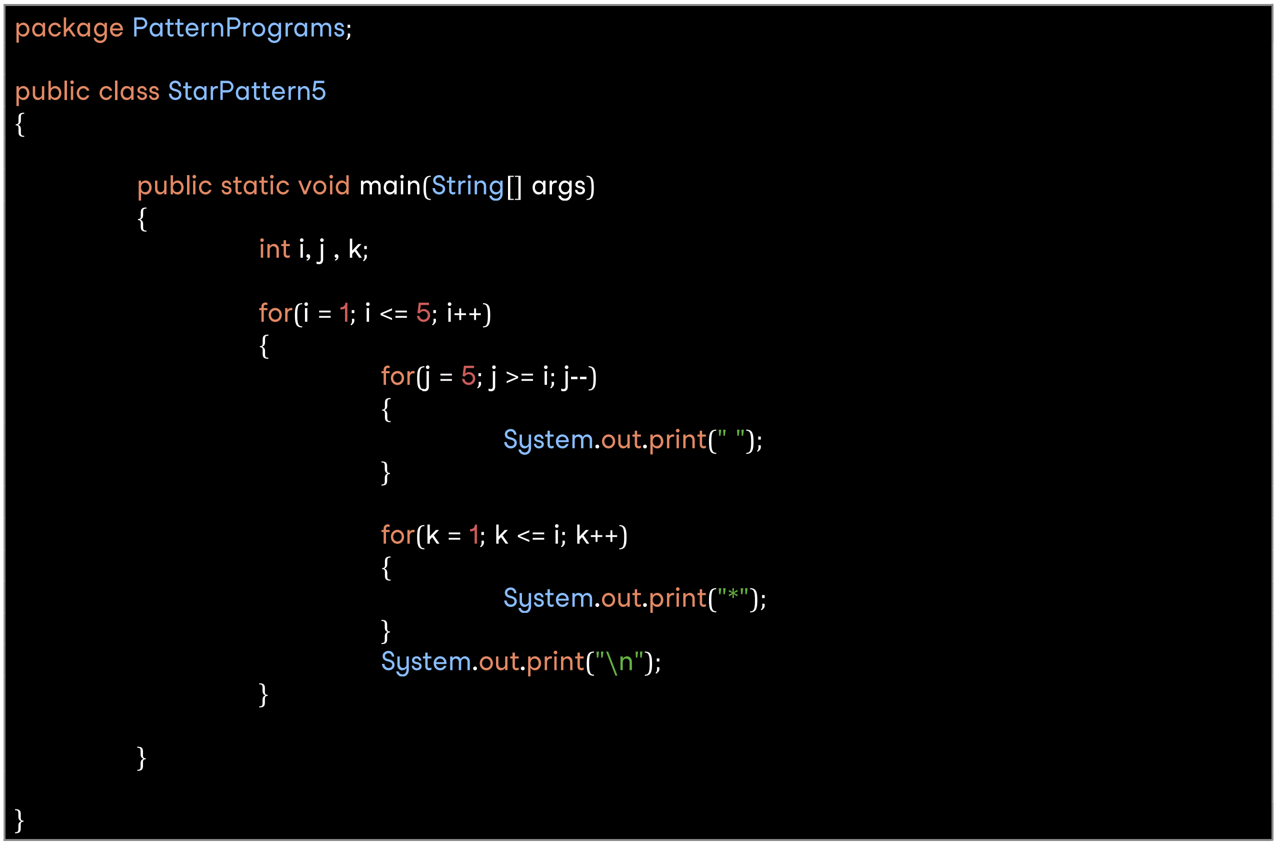


```java
package PatternPrograms;

public class StarPattern3
{

    public static void main(String[] args)
    {
        int i, j;

        for(i = 1; i <= 5; i++)
        {
            for(j = 5; j >= i; j--)
            {
                System.out.print("*");
            }
            System.out.print("\n");
        }
    }

}
```

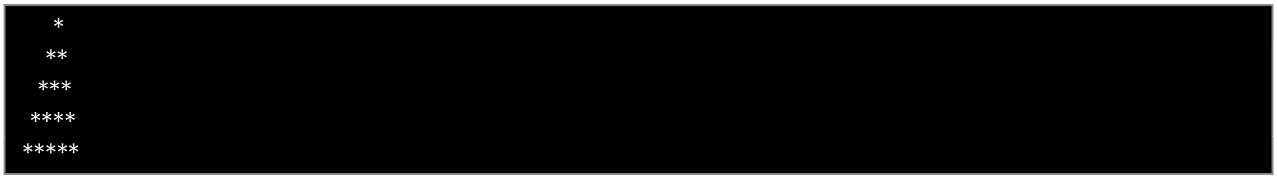
```
*****
****
***
**
*
```

```
package PatternPrograms;

public class StarPattern4
{

    public static void main(String[] args)
    {
        int i, j;

        for(i = 1; i <= 5; i++)
        {
            for(j = 1; j <= i; j++)
            {
                System.out.print("*");
            }
            System.out.print("\n");
        }

        for(i = 1; i <= 5; i++)
        {
            for(j = 4; j >= i; j--)
            {
                System.out.print("*");
            }
            System.out.print("\n");
        }
    }

}
```

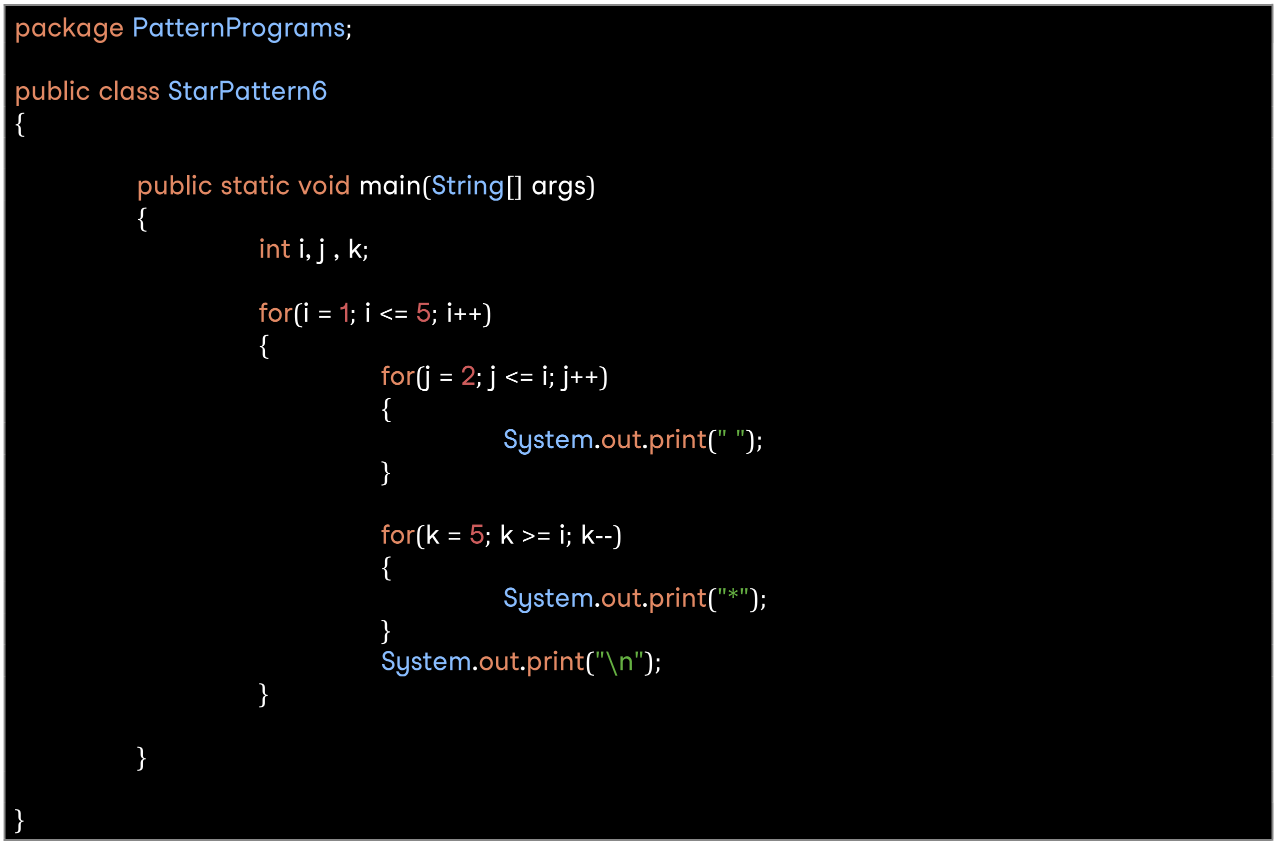
```
*
**
***
****
*****
****
***
**
*
```



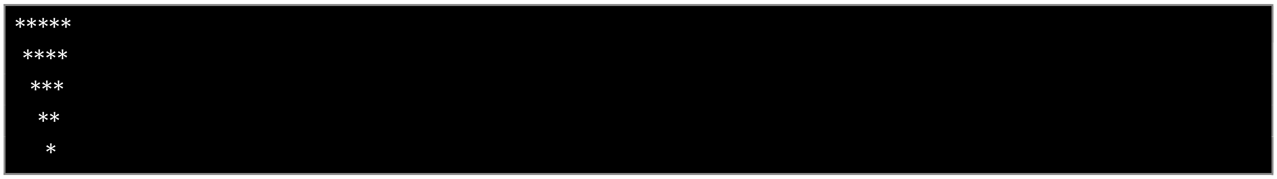
```java
package PatternPrograms;

public class StarPattern5
{

    public static void main(String[] args)
    {
        int i, j , k;

        for(i = 1; i <= 5; i++)
        {
            for(j = 5; j >= i; j--)
            {
                System.out.print(" ");
            }

            for(k = 1; k <= i; k++)
            {
                System.out.print("*");
            }
            System.out.print("\n");
        }

    }

}
```



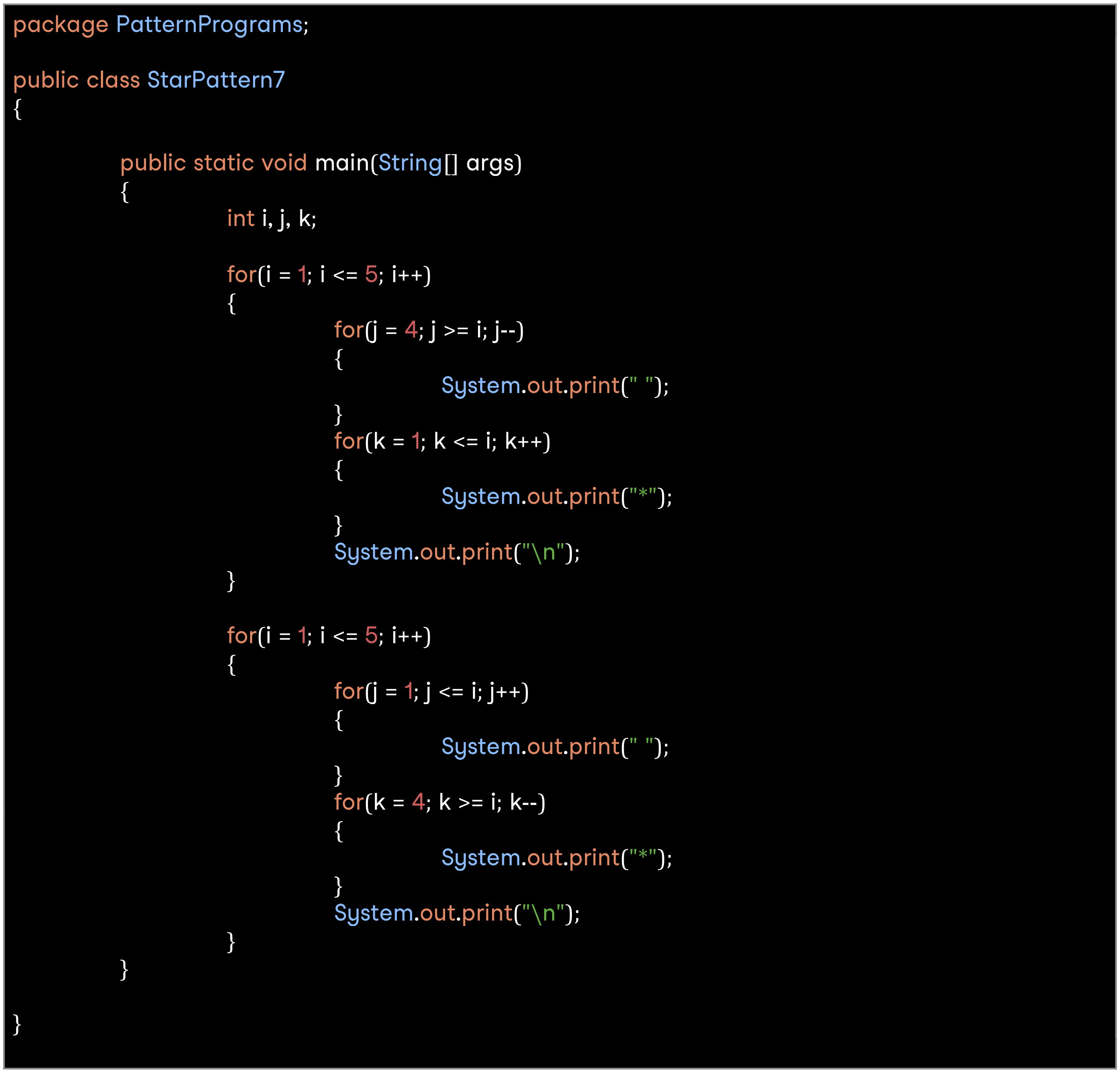
```
     *
    **
   ***
  ****
 *****
```



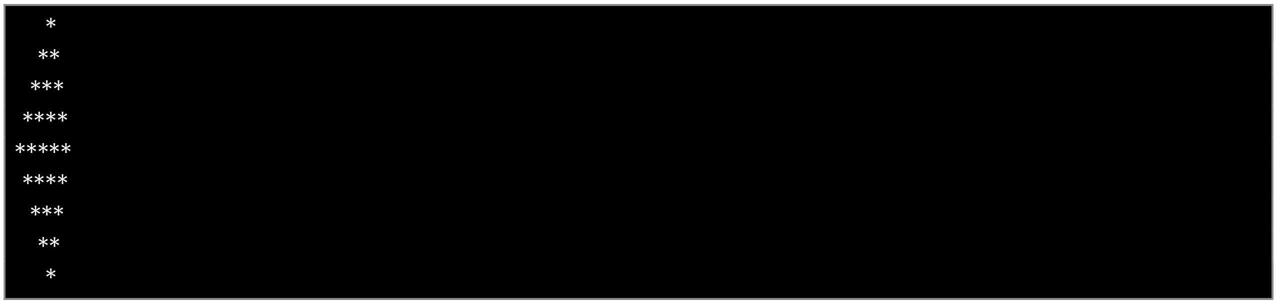
```java
package PatternPrograms;

public class StarPattern6
{

    public static void main(String[] args)
    {
        int i, j , k;

        for(i = 1; i <= 5; i++)
        {
            for(j = 2; j <= i; j++)
            {
                System.out.print(" ");
            }

            for(k = 5; k >= i; k--)
            {
                System.out.print("*");
            }
            System.out.print("\n");
        }

    }

}
```



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



```
package PatternPrograms;

public class StarPattern7
{

	public static void main(String[] args)
	{
		int i, j, k;

		for(i = 1; i <= 5; i++)
		{
			for(j = 4; j >= i; j--)
			{
				System.out.print(" ");
			}
			for(k = 1; k <= i; k++)
			{
				System.out.print("*");
			}
			System.out.print("\n");
		}

		for(i = 1; i <= 5; i++)
		{
			for(j = 1; j <= i; j++)
			{
				System.out.print(" ");
			}
			for(k = 4; k >= i; k--)
			{
				System.out.print("*");
			}
			System.out.print("\n");
		}
	}

}
```

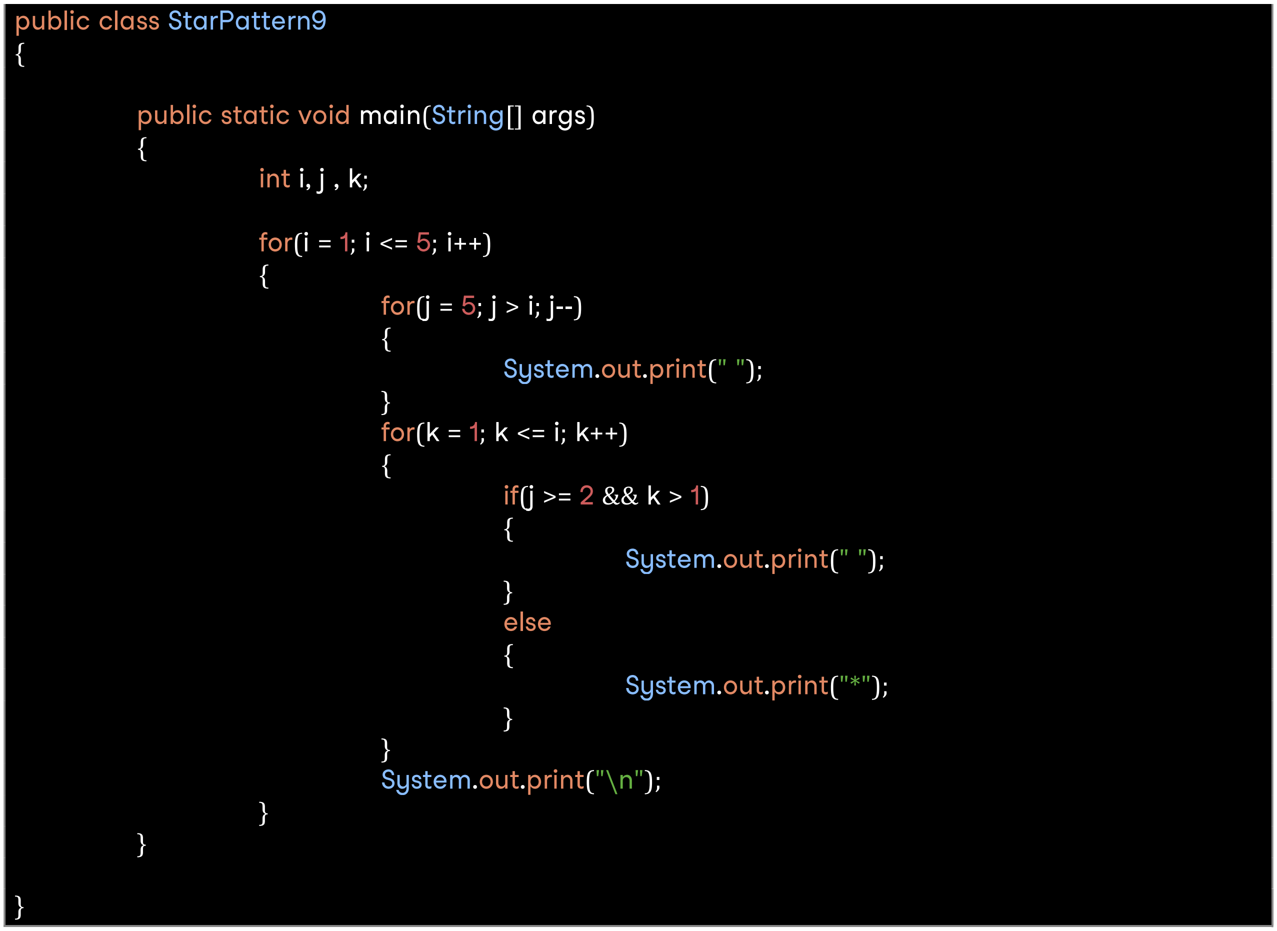


```
    *
   **
  ***
 ****
*****
 ****
  ***
   **
    *
```

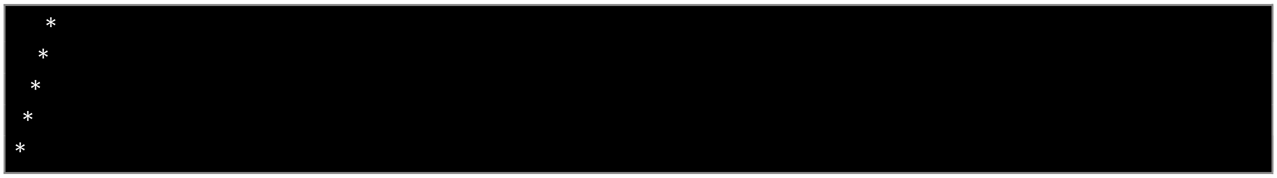
```
package PatternPrograms;

public class StarPattern8
{

    public static void main(String[] args)
    {
        int i, j;

        for(i = 1; i <= 5; i++)
        {
            for(j = 1; j <= i; j++)
            {
                if(i >= 2 && j <= i-1)
                {
                    System.out.print(" ");
                }
                else
                {
                    System.out.print("*");
                }
            }
            System.out.print("\n");
        }
    }

}
```

```
*
 *
  *
   *
    *
```

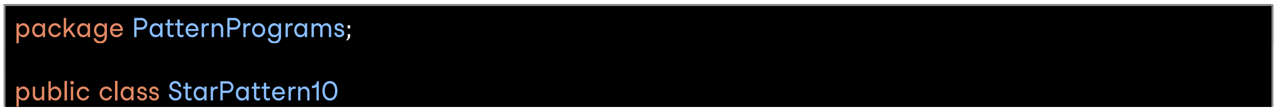
```
package PatternPrograms;
```



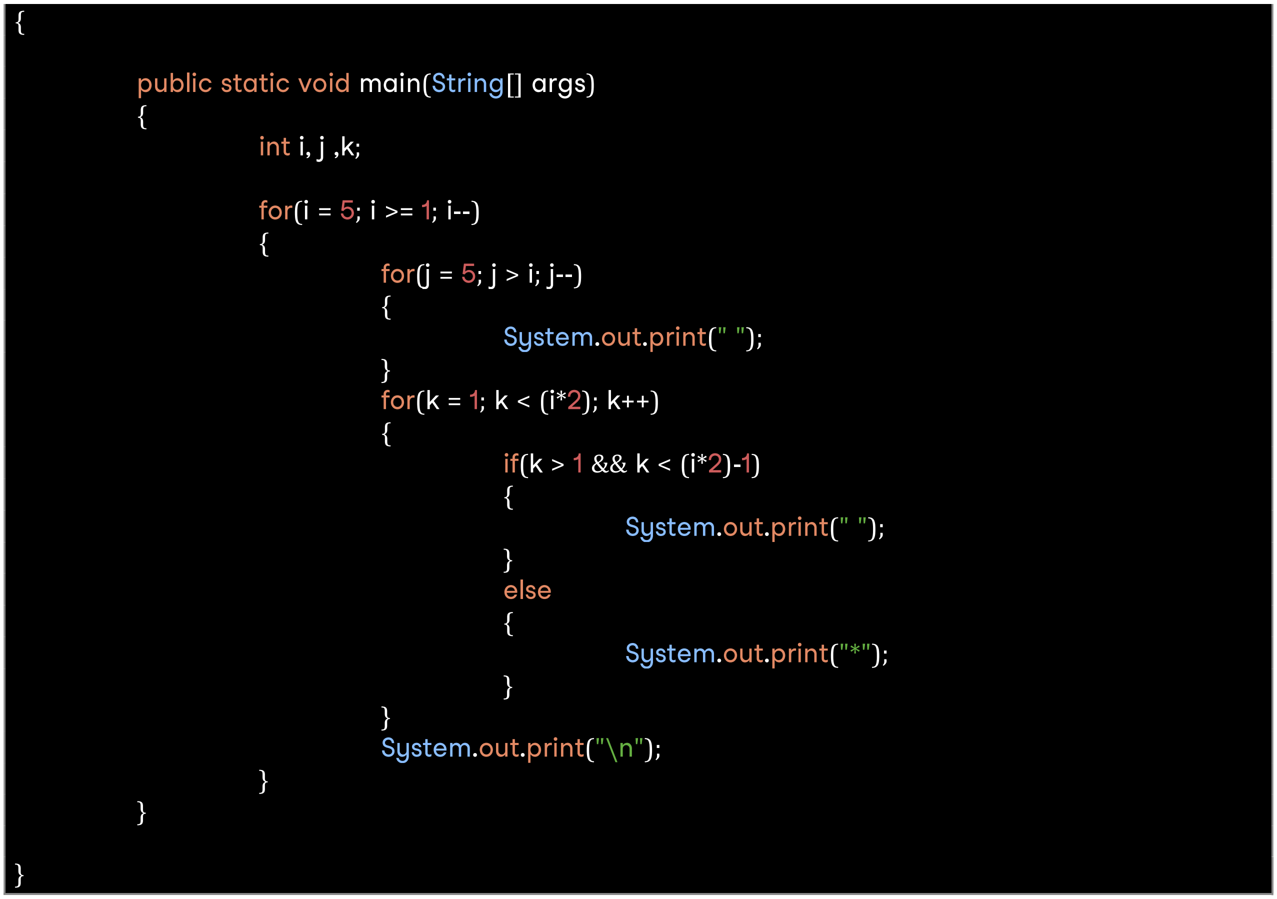
```java
public class StarPattern9
{

    public static void main(String[] args)
    {
        int i, j , k;

        for(i = 1; i <= 5; i++)
        {
            for(j = 5; j > i; j--)
            {
                System.out.print(" ");
            }
            for(k = 1; k <= i; k++)
            {
                if(j >= 2 && k > 1)
                {
                    System.out.print(" ");
                }
                else
                {
                    System.out.print("*");
                }
            }
            System.out.print("\n");
        }
    }

}
```



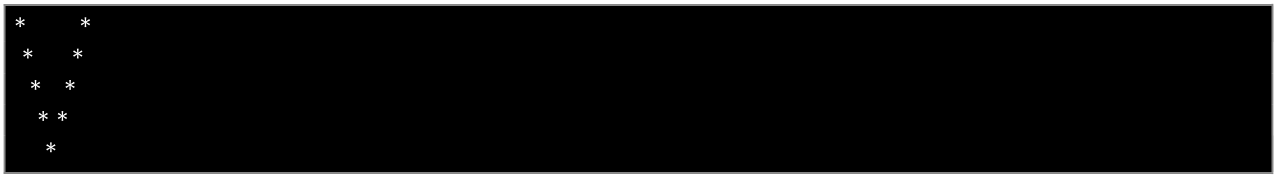
```
    *
   *
  *
 *
*
```



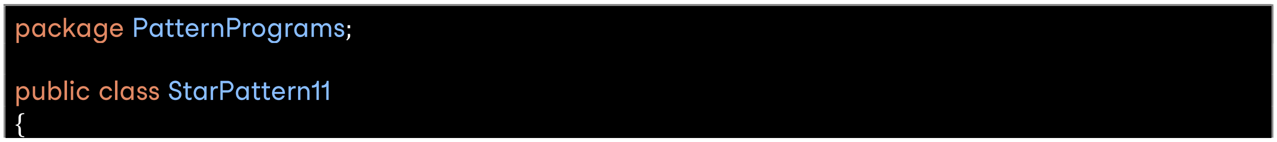
```java
package PatternPrograms;

public class StarPattern10
```



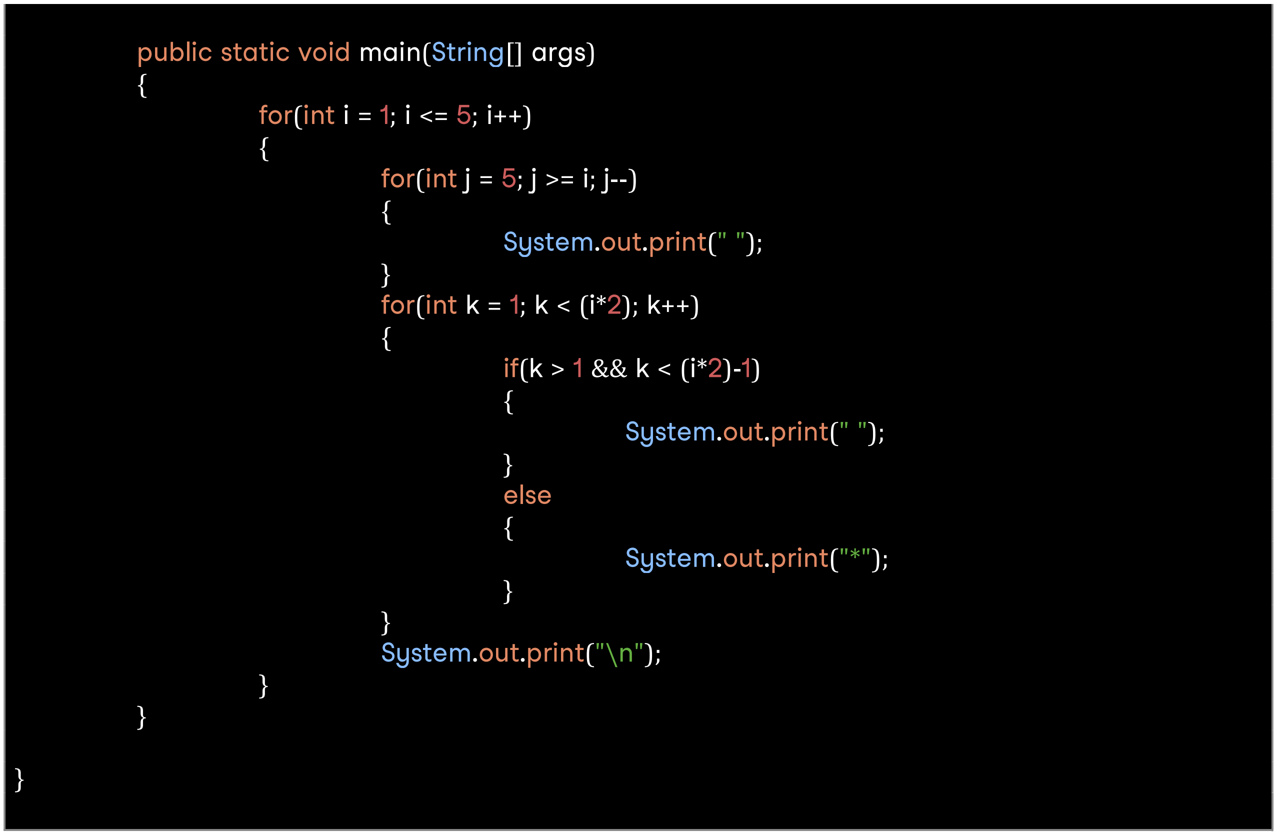
```
{
	public static void main(String[] args)
	{
		int i, j ,k;
		
		for(i = 5; i >= 1; i--)
		{
			for(j = 5; j > i; j--)
			{
				System.out.print(" ");
			}
			for(k = 1; k < (i*2); k++)
			{
				if(k > 1 && k < (i*2)-1)
				{
					System.out.print(" ");
				}
				else
				{
					System.out.print("*");
				}
			}
			System.out.print("\n");
		}
	}

}
```



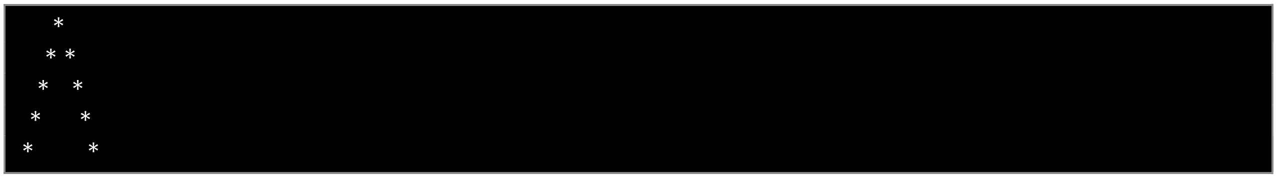
```
*       *
 *     *
  *   *
   * *
    *
```



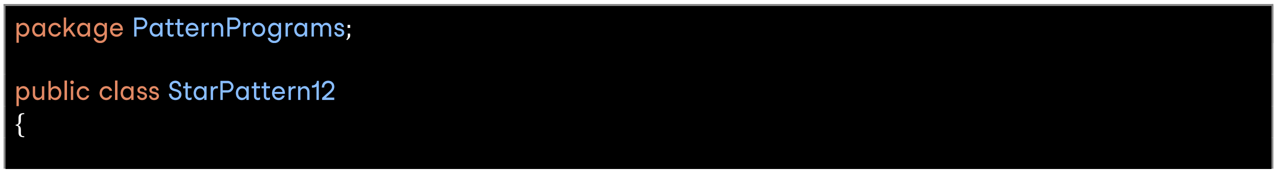
```
package PatternPrograms;

public class StarPattern11
{
```



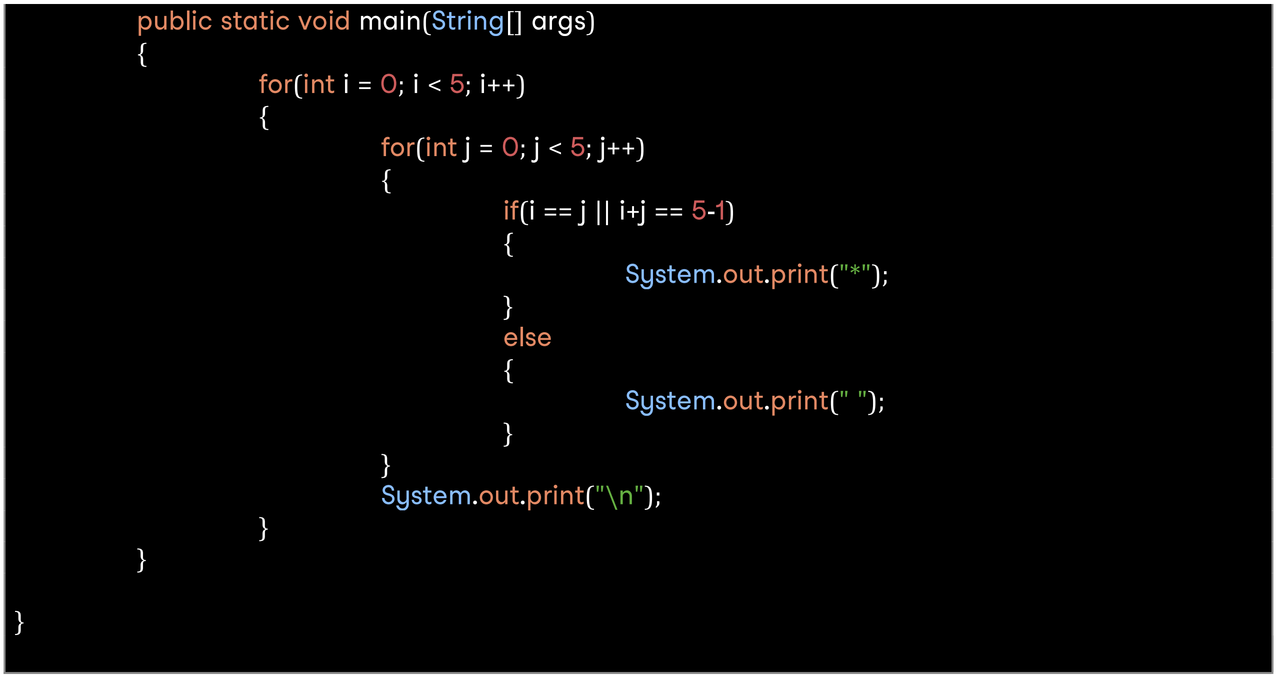
```
            public static void main(String[] args)
            {
                        for(int i = 1; i <= 5; i++)
                        {
                                    for(int j = 5; j >= i; j--)
                                    {
                                                System.out.print(" ");
                                    }
                                    for(int k = 1; k < (i*2); k++)
                                    {
                                                if(k > 1 && k < (i*2)-1)
                                                {
                                                            System.out.print(" ");
                                                }
                                                else
                                                {
                                                            System.out.print("*");
                                                }
                                    }
                                    System.out.print("\n");
                        }
            }

}
```



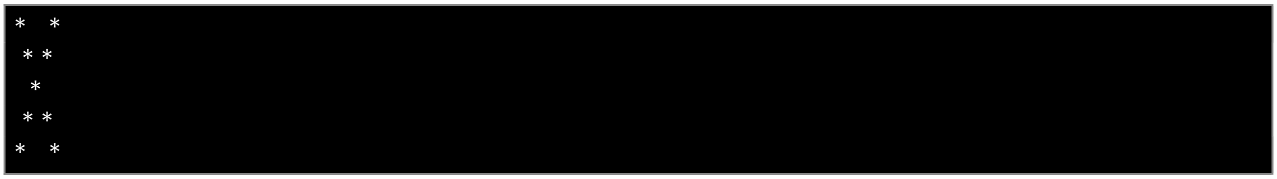
```
     *
    * *
   *   *
  *     *
 *       *
```



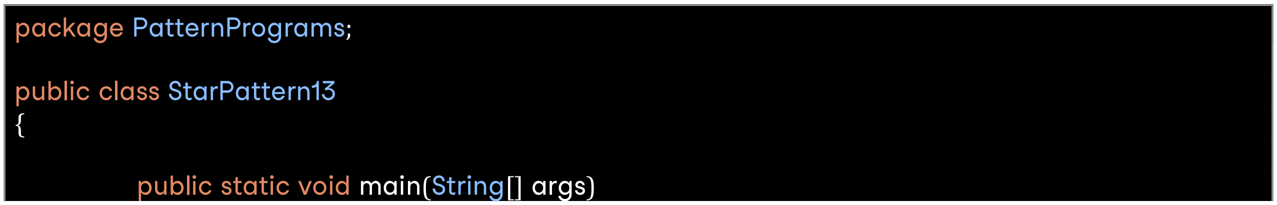
```
package PatternPrograms;

public class StarPattern12
{
```



```
	public static void main(String[] args)
	{
		for(int i = 0; i < 5; i++)
		{
			for(int j = 0; j < 5; j++)
			{
				if(i == j || i+j == 5-1)
				{
					System.out.print("*");
				}
				else
				{
					System.out.print(" ");
				}
			}
			System.out.print("\n");
		}
	}

}
```



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



```
package PatternPrograms;

public class StarPattern13
{

	public static void main(String[] args)
```

```
    {
        for(int i = 1; i <= 5; i++)
        {
            for(int j = 1; j <= 5; j++)
            {
                if(i >= 2 && j >= 2 && i <= 4 && j <= 4)
                {
                    System.out.print(" ");
                }
                else
                {
                    System.out.print("*");
                }
            }
            System.out.print("\n");
        }
    }

}
```

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

Numeric Pattern Programs:

```
package PatternPrograms;

public class NumberPattern1
{
	public static void main(String[] args)
	{
		int i, j;

		for(i = 1; i <= 5; i++)
		{
			for(j = 1; j <= i; j++)
			{
				System.out.print(i);
			}
			System.out.print("\n");
		}
	}

}
```

```
1
22
333
4444
55555
```

```
package PatternPrograms;

public class NumberPattern2
{

	public static void main(String[] args)
	{
		int i, j;

		for(i = 1; i <= 5; i++)
		{
			for(j = 1; j <= i; j++)
			{
				System.out.print(j);
			}
			System.out.print("\n");
		}
	}

}
```

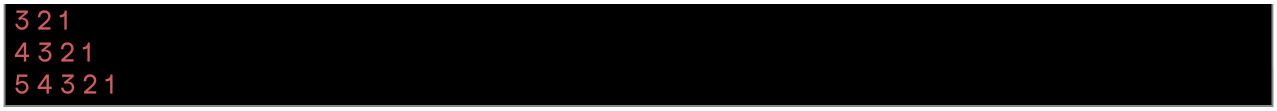
```
1
12
123
1234
12345
```

```
// Floyd's triangle
package PatternPrograms;
public class NumberPattern3
{

    public static void main(String[] args)
    {
        int i, j, count = 0;

        for(i = 1; i <= 5; i++)
        {
            for(j = 1; j <= i; j++)
            {
                count = count + 1;
                System.out.print(count+" ");
            }
            System.out.print("\n");
        }
    }

}
```

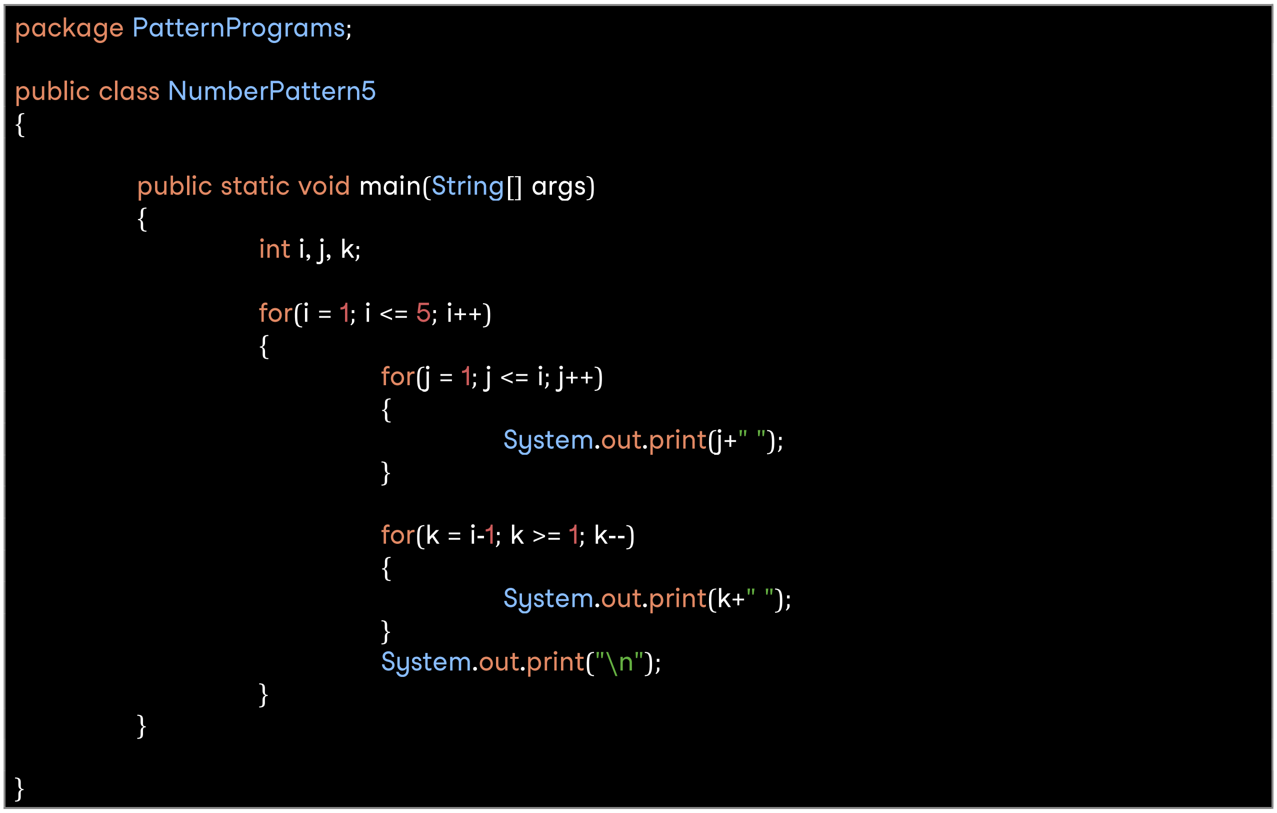
```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

```
package PatternPrograms;

public class NumberPattern4
{

    public static void main(String[] args)
    {
        int i, j;

        for(i = 1; i <= 5; i++)
        {
            for(j = i; j >= 1; j--)
            {
                System.out.print(j+" ");
            }
            System.out.print("\n");
        }
    }

}
```

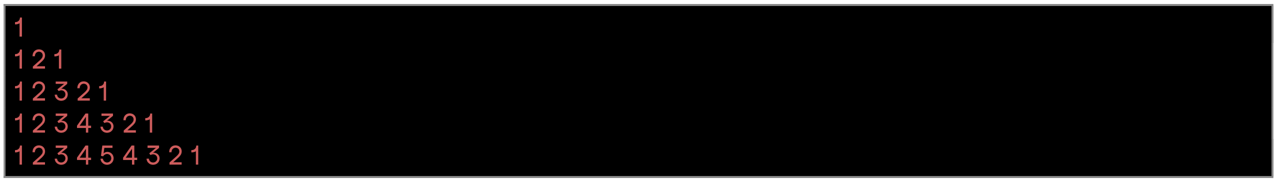
```
1
2 1
```



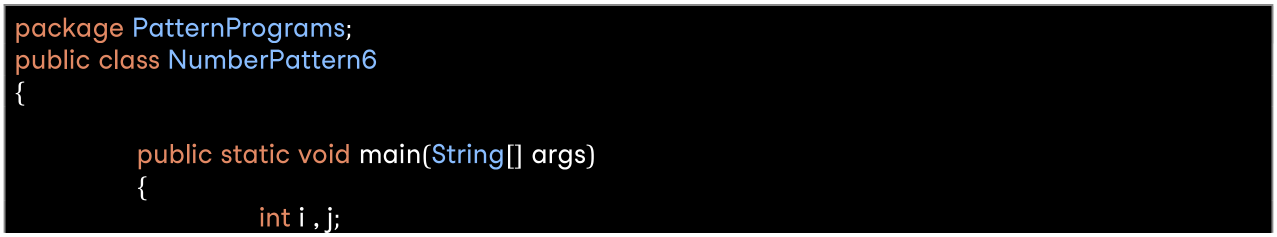
```
3 2 1
4 3 2 1
5 4 3 2 1
```



```
package PatternPrograms;

public class NumberPattern5
{

    public static void main(String[] args)
    {
        int i, j, k;

        for(i = 1; i <= 5; i++)
        {
            for(j = 1; j <= i; j++)
            {
                System.out.print(j+" ");
            }

            for(k = i-1; k >= 1; k--)
            {
                System.out.print(k+" ");
            }
            System.out.print("\n");
        }
    }

}
```



```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
```

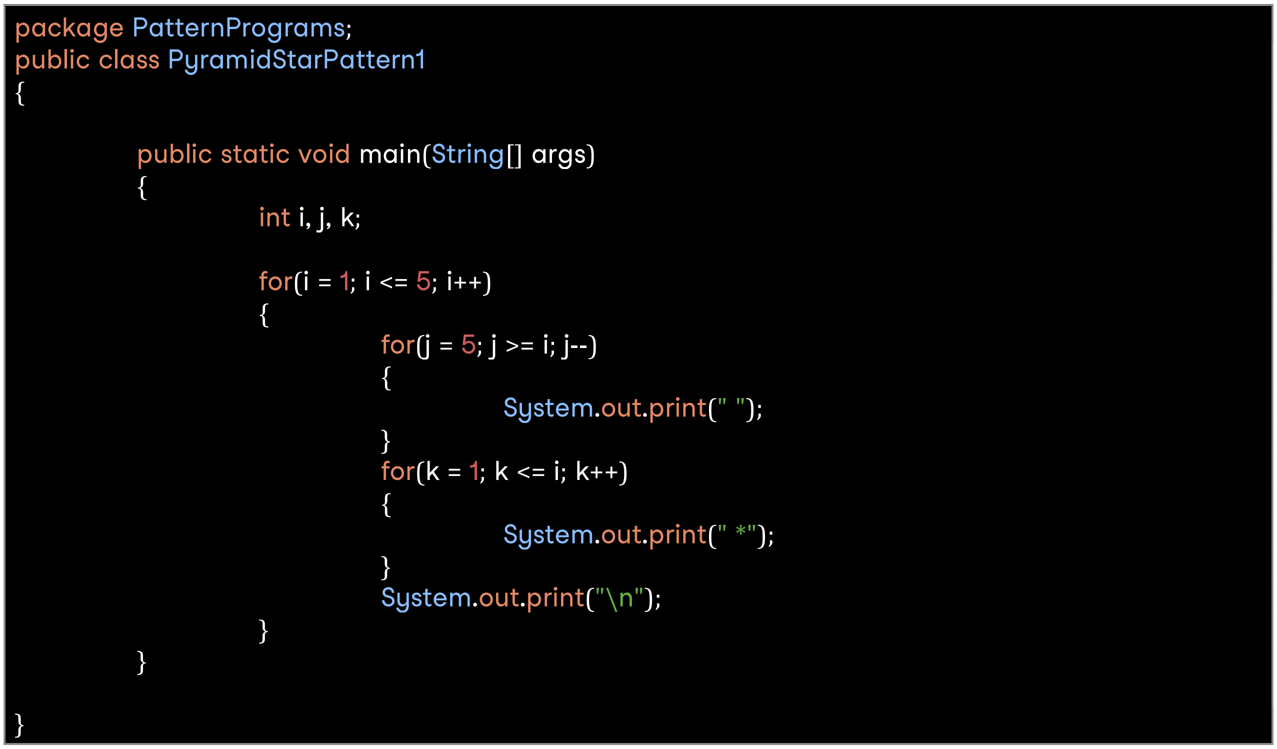


```
package PatternPrograms;
public class NumberPattern6
{

    public static void main(String[] args)
    {
        int i , j;
        for(i = 1; i <= 5; i++)
        {
            for(j = 5; j >= i; j--)
            {
                System.out.print(j + " ");
            }
            System.out.print("\n");
        }
    }

}
```

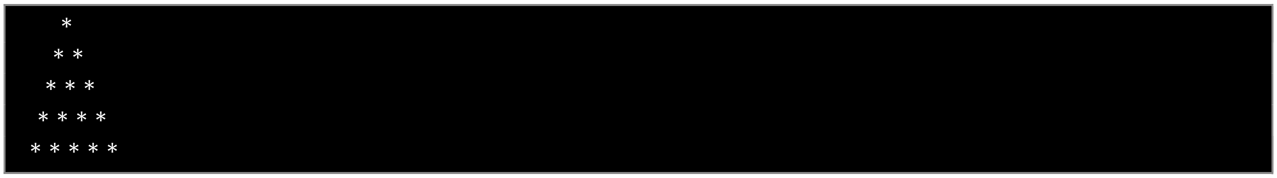
```
5 4 3 2 1
5 4 3 2
5 4 3
5 4
5
```

```
package PatternPrograms;

public class NumberPattern7
{

    public static void main(String[] args)
    {
        for(int i = 1; i <= 5; i++)
        {
            int no = i;
            for(int j = 1; j <= i; j++)
            {
                System.out.print(no + " ");
                no  = no + 5 - j;
            }
            System.out.print("\n");
        }
    }

}
```

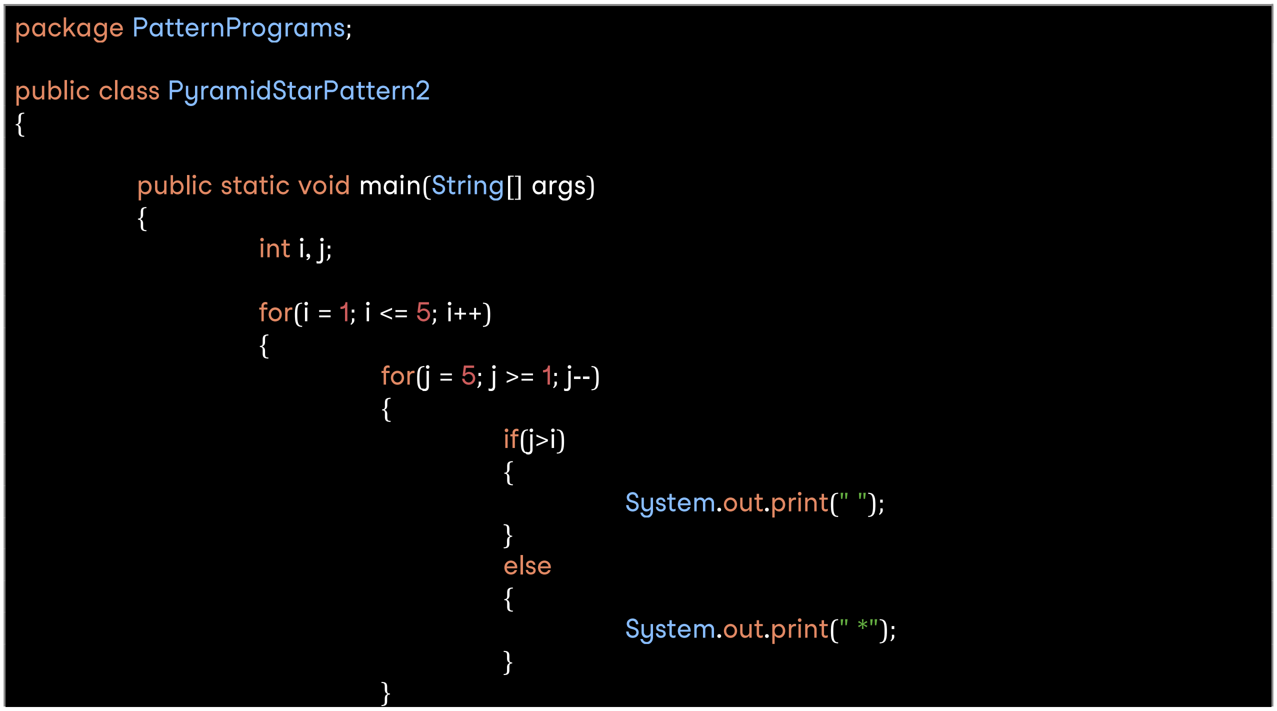
```
1
2 6
3 7 10
4 8 11 13
5 9 12 14 15
```



```
package PatternPrograms;
public class PyramidStarPattern1
{

    public static void main(String[] args)
    {
        int i, j, k;

        for(i = 1; i <= 5; i++)
        {
            for(j = 5; j >= i; j--)
            {
                System.out.print(" ");
            }
            for(k = 1; k <= i; k++)
            {
                System.out.print(" *");
            }
            System.out.print("\n");
        }
    }

}
```



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



```
package PatternPrograms;

public class PyramidStarPattern2
{

    public static void main(String[] args)
    {
        int i, j;

        for(i = 1; i <= 5; i++)
        {
            for(j = 5; j >= 1; j--)
            {
                if(j>i)
                {
                    System.out.print(" ");
                }
                else
                {
                    System.out.print(" *");
                }
            }
```

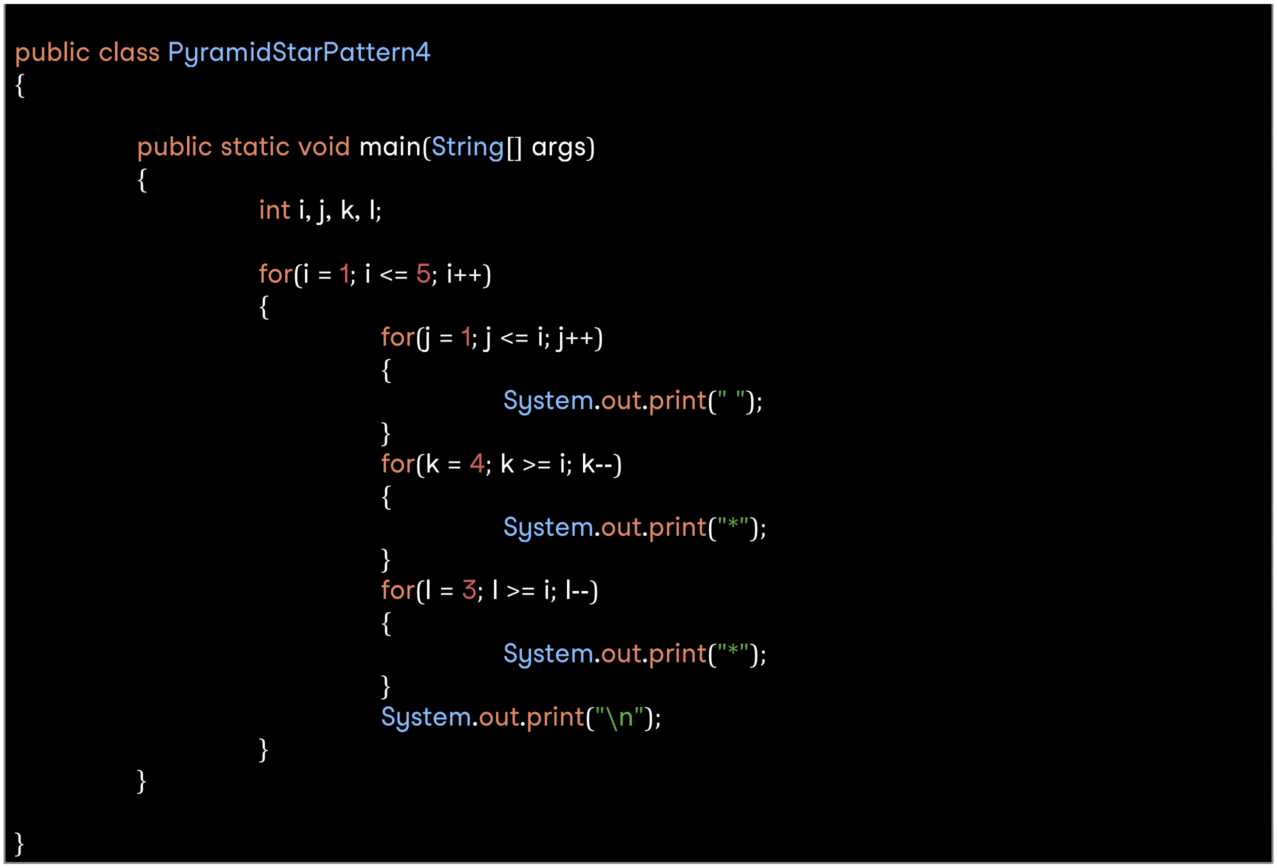
```
                        System.out.print("\n");
                }
        }

}
```

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

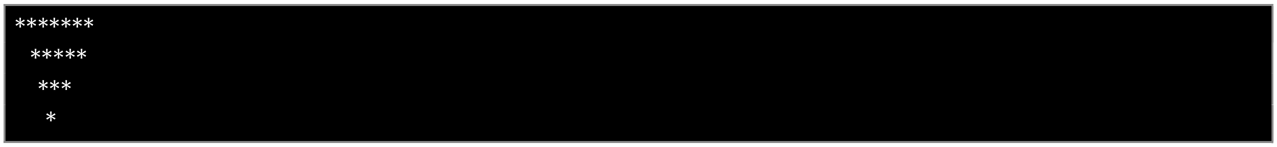
```
package PatternPrograms;

public class PyramidStarpattern3
{

        public static void main(String[] args)
        {
                int i, j, k, l;

                for(i = 1; i <= 5; i++)
                {
                        for(j = 5; j >= i; j--)
                        {
                                System.out.print(" ");
                        }
                        for(k = 1; k <= i; k++)
                        {
                                System.out.print("*");
                        }
                        for(l = 2; l <= i; l++)
                        {
                                System.out.print("*");
                        }
                        System.out.print("\n");
                }
        }

}
```

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

```
package PatternPrograms;
```



```
public class PyramidStarPattern4
{

            public static void main(String[] args)
            {
                        int i, j, k, l;

                        for(i = 1; i <= 5; i++)
                        {
                                    for(j = 1; j <= i; j++)
                                    {
                                                System.out.print(" ");
                                    }
                                    for(k = 4; k >= i; k--)
                                    {
                                                System.out.print("*");
                                    }
                                    for(l = 3; l >= i; l--)
                                    {
                                                System.out.print("*");
                                    }
                                    System.out.print("\n");
                        }
            }

}
```



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

Sorting

- ❖ Sorting is the process of arranging the data in some logical order.
- ❖ In case of numeric values, logical order may be in ascending or descending order.
- ❖ In case of alphanumeric values, it can be in dictionary order.

Bubble sort

- Bubble sort, also referred as sinking sort, is a comparison-based algorithm i.e. comparing each pair of adjacent items and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted.

```
package SortingPrograms;

public class BubbleSort
{

    public static void main(String[] args)
    {
        int a[] = {36, 19, 29, 12, 5};
        int temp;

        for(int i = 0; i < a.length; i++) // for number of round
        {
            int flag = 0;
            //for two adjacent elements
            for(int j = 0; j < a.length-1-i; j++)
            {
                if(a[j] > a[j+1])
                {
                    temp = a[j];
                    a[j] = a[j+1];
                    a[j+1] = temp;
                    flag = 1;
                }
            }
            if(flag == 0)
            {
                break;
            }
        }

        for(int i = 0; i < a.length; i++)
        {
            System.out.print(a[i] + " ");
        }
    }

}
```

```
5 12 19 29 36
```

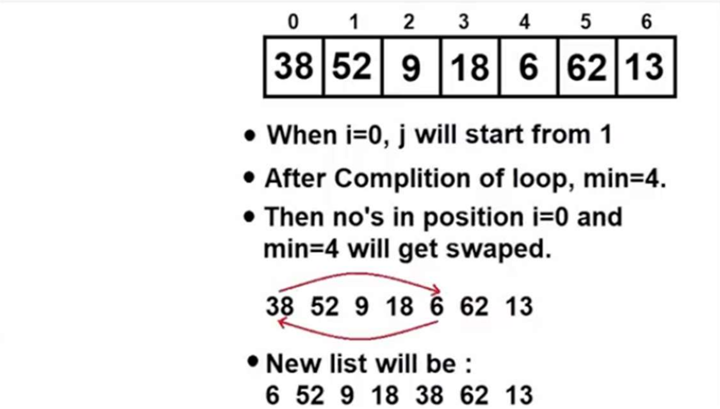
Selection Sort

- This algorithm is based on the idea of finding the minimum and maximum element in the unsorted array and then putting in its correct position or a sorted array.
- The selection sort is a combination of searching and sorting.

- It sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. In every iteration of selection sort, the minimum element from the unsorted subarray is picked and moved to sorted subarray.

```
// Number selection sort
package SortingPrograms;

public class SelectionSort
{

        public static void main(String[] args)
        {
                int a[] = {38, 52, 9, 18, 6, 62, 13};
                int min, temp;
                for(int i = 0; i < a.length; i++)
                {
                        min = i;
                        for(int j = i+1; j < a.length; j++)
                        {
                                if(a[j] < a[min])
                                {
                                        min = j;
                                }
                        }

                        temp = a[i];
                        a[i] = a[min];
                        a[min] = temp;
                }

                for(int i = 0; i < a.length; i++)
                {
                        System.out.print(a[i]+ " ");
                }
        }

}
```

```
6 9 13 18 38 52 62
```



0	1	2	3	4	5	6
38	52	9	18	6	62	13

- When i=0, j will start from 1
- After Complition of loop, min=4.
- Then no's in position i=0 and min=4 will get swaped.

- New list will be :
  6 52 9 18 38 62 13

```
// Selection sort for string

package SortingPrograms;

public class StringSelectionSort
{

            public static void main(String[] args)
            {
                        String a[]  = {"ashu", "som", "lucky", "guddy", "rahul"};
                        int min;
                        String temp = "";
                        for(int i = 0; i < a.length; i++)
                        {
                                    min = i;
                                    for(int j = i+1; j < a.length; j++)
                                    {
                                                if(a[j].compareTo(a[min]) < 0)
                                                {
                                                            min = j;
                                                }
                                    }

                                    temp = a[i];
                                    a[i] = a[min];
                                    a[min] = temp;
                        }

                        for(int i = 0; i < a.length; i++)
                        {
                                    System.out.print(a[i]+" ");
                        }
            }

}
```

```
ashu guddy lucky rahul som
```

Insertion sort

- Insertion sort is a simple sorting algorithm that works the way we sort playing cards in our hands.
- We choose one card and insert it into its correct position (ascending or descending order).

```
package SortingPrograms;

public class InsertionSort
{

    public static void main(String[] args)
    {
        int a[] = {6, 2, 1, 5, 9, 7};
        int temp, j;
        for(int i = 0; i < a.length; i++)
        {
            temp = a[i];
            j = i;
            while(j > 0 && a[j-1] > temp)
            {
                a[j] = a[j-1];
                j = j - 1;
            }

            a[j] = temp;
        }

        for(int i = 0; i < a.length; i++)
        {
            System.out.print(a[i]+" ");
        }
    }

}
```

```
1 2 5 6 7 9
```

Merge sort

- Its time complexity is (i.e., T(n) = 2T(n/2) + n) is better as compared to bubble sort, selection sort and insertion sort.
- It based on Divide and conquer algorithm.
- Divide the array into two halves. Repeatedly sort each half, them merge two halves.

```
package SortingPrograms;

public class MergeSort
{
    int[] array;
    int[] tempMergeArr;
    int length;

    public static void main(String[] args)
    {
        int[] inputArr = {48, 36, 58, 3, 12, 51, 30, 9};
        MergeSort ms = new MergeSort();
        ms.sort(inputArr);

        for(int i:inputArr)
        {
            System.out.print(i+" ");
        }
    }

    public void sort(int inputArr[])
    {
        this.array = inputArr;
        this.length = inputArr.length;
        this.tempMergeArr = new int[length];
        divideArray(0, length-1);
    }

    public void divideArray(int lowerIndex, int higherIndex)
    {
        if(lowerIndex<higherIndex)
        {
            int middle = lowerIndex+(higherIndex-lowerIndex)/2;
            divideArray(lowerIndex, middle);
            divideArray(middle+1, higherIndex);
            mergeArray(lowerIndex, middle, higherIndex);
        }
    }

    public void mergeArray(int lowerIndex, int middle, int higherIndex)
    {
        for(int i = lowerIndex; i <= higherIndex; i++)
        {
            tempMergeArr[i] = array[i];
        }

        int i = lowerIndex;
        int j = middle + 1;
        int k = lowerIndex;
```

```
                while(i <= middle && j <= higherIndex)
                {
                        if(tempMergeArr[i] <= tempMergeArr[j])
                        {
                                array[k] = tempMergeArr[i];
                                i++;
                        }
                        else
                        {
                                array[k] = tempMergeArr[j];
                                j++;
                        }
                        k++;
                }

                while(i <= middle)
                {
                        array[k] = tempMergeArr[i];
                        k++;
                        i++;
                }
        }

}
```

```
3 9 12 30 36 48 51 58
```

Quick Sort

- Quick sort sometimes called as partition-Exchange sort.
- This algorithm is also based on the divide and conquer approach.

- It reduces the space complexity.
- It based on the idea of choosing one element as pivot element and partitioning the array around it such that the left side of the pivot element contains all elements less than the pivot element and right side contains all the elements greater than the pivot.

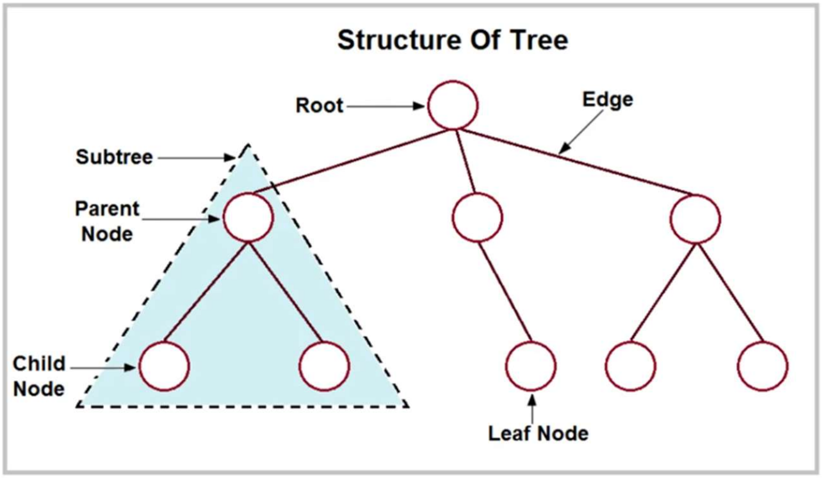
```
package SortingPrograms;

public class QuickSort
{

    public static void main(String[] args)
    {
        int arr[] = {5, 6, 85, 12, 56, 34, 9, 96};
        int leng = arr.length;

        QuickSort qsm = new QuickSort();
        qsm.quickSortRecursion(arr, 0, leng-1);
        qsm.printArray(arr);
    }

    int partition(int[] arr, int low, int high)
    {
        int pivot = arr[(low+high)/2];
        while(low <= high)
        {
            while(arr[low] < pivot)
            {
                low++;
            }
            while(arr[high] > pivot)
            {
                high--;
            }

            if(low <= high)
            {
                int temp = arr[low];
                arr[low] = arr[high];
                arr[high] = temp;

                low++;
                high--;
            }
        }

        return low;
    }

    void quickSortRecursion(int[] arr, int low, int high)
    {
        int pi = partition(arr, low, high);
        if(low < pi-1)
        {
            quickSortRecursion(arr, low, pi-1);
        }

        if(pi < high)
        {
            quickSortRecursion(arr, pi, high);
        }
    }

    void printArray(int[] arr)
    {
        for(int i:arr)
        {
            System.out.print(i+" ");
        }
    }

}
```

```
5 6 9 12 34 56 85 96
```

Heap Sort

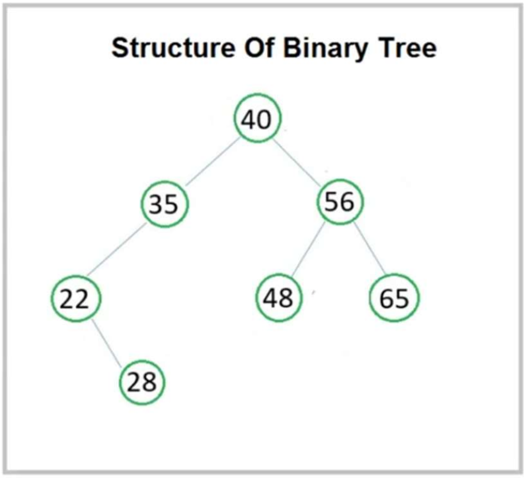
Heap sort is a completely binary tree or almost completely binary tree.

Tree is a Non-Linear Data Structure that consists of nodes with parent-child relationship.

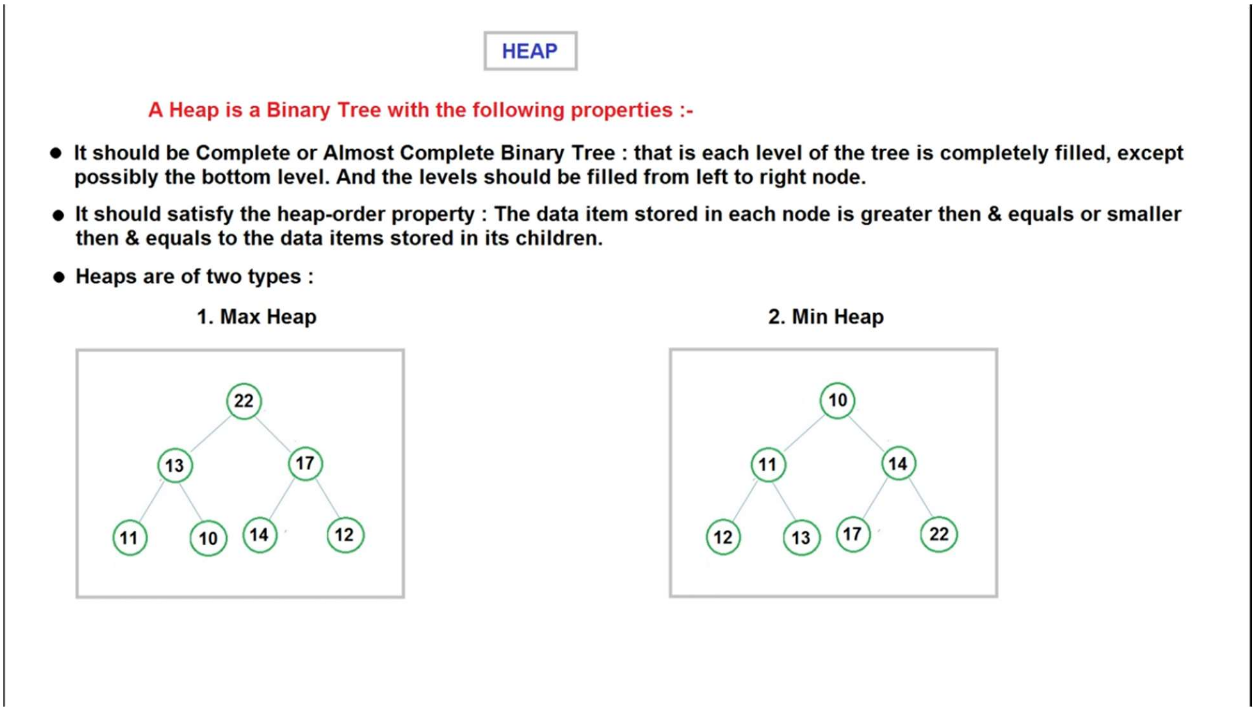


- Tree is a sequence of nodes
- There is a starting node known as root node
- Every node other than the root node has a parent node
- Nodes may have any any number of children

A Binary Tree is a Non-Linear Data Structure in which each node has maximum of two child nodes.



- Binary tree is the one in which each node has maximum of two child- node.
- The order of binary tree is ‘2’.
- Binary tree does not allow duplicate values.
- While constructing a binary, if an element is less than the value of its parent node, it is placed on the left side of it otherwise right side.
- A binary tree is shown for the element 40, 56, 35, 48, 22, 65, 28.



HEAP

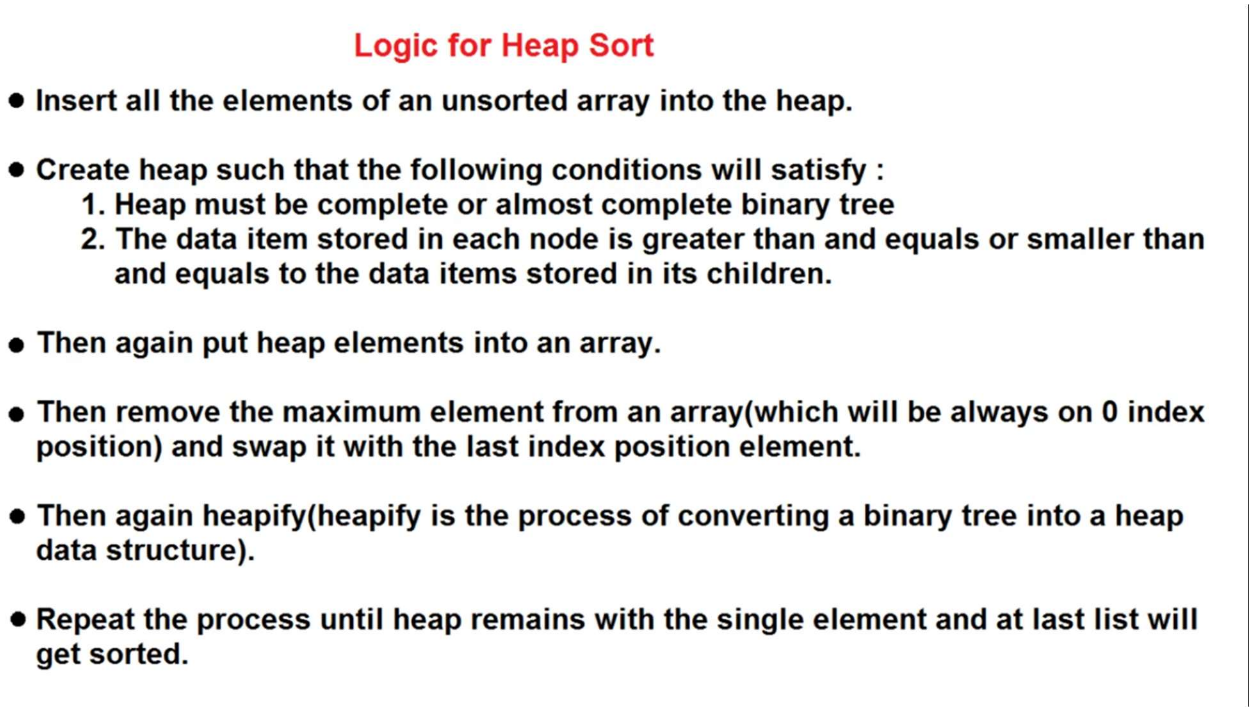
A Heap is a Binary Tree with the following properties :-

- It should be Complete or Almost Complete Binary Tree : that is each level of the tree is completely filled, except possibly the bottom level. And the levels should be filled from left to right node.
- It should satisfy the heap-order property : The data item stored in each node is greater then & equals or smaller then & equals to the data items stored in its children.
- Heaps are of two types :

1. Max Heap

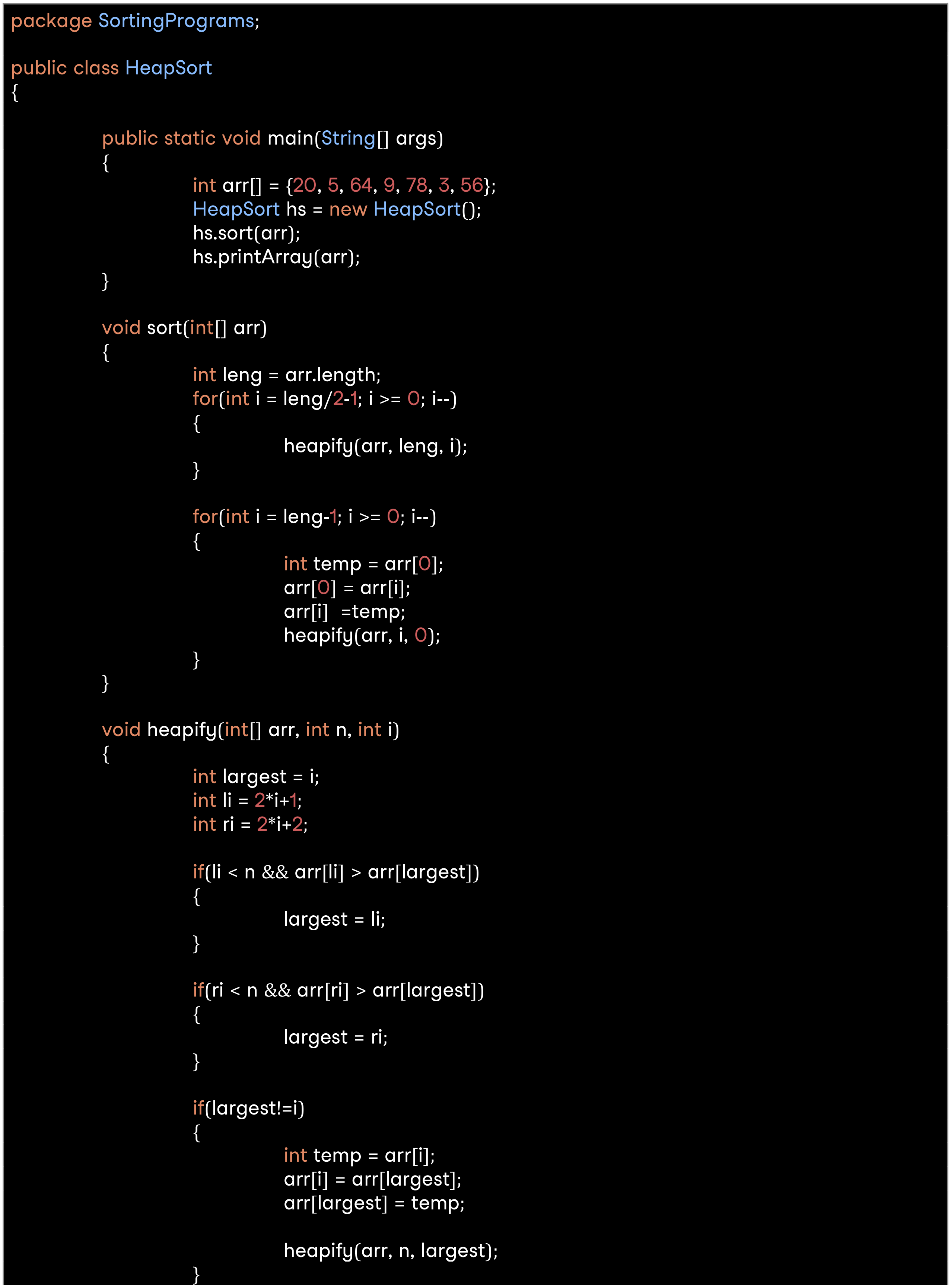
2. Min Heap

10
11
14
12
13
17
22



Logic for Heap Sort

- Insert all the elements of an unsorted array into the heap.
- Create heap such that the following conditions will satisfy :
  1. Heap must be complete or almost complete binary tree
  2. The data item stored in each node is greater than and equals or smaller than and equals to the data items stored in its children.
- Then again put heap elements into an array.
- Then remove the maximum element from an array(which will be always on 0 index position) and swap it with the last index position element.
- Then again heapify(heapify is the process of converting a binary tree into a heap data structure).
- Repeat the process until heap remains with the single element and at last list will get sorted.



```java
package SortingPrograms;

public class HeapSort
{

    public static void main(String[] args)
    {
        int arr[] = {20, 5, 64, 9, 78, 3, 56};
        HeapSort hs = new HeapSort();
        hs.sort(arr);
        hs.printArray(arr);
    }

    void sort(int[] arr)
    {
        int leng = arr.length;
        for(int i = leng/2-1; i >= 0; i--)
        {
            heapify(arr, leng, i);
        }

        for(int i = leng-1; i >= 0; i--)
        {
            int temp = arr[0];
            arr[0] = arr[i];
            arr[i]  =temp;
            heapify(arr, i, 0);
        }
    }

    void heapify(int[] arr, int n, int i)
    {
        int largest = i;
        int li = 2*i+1;
        int ri = 2*i+2;

        if(li < n && arr[li] > arr[largest])
        {
            largest = li;
        }

        if(ri < n && arr[ri] > arr[largest])
        {
            largest = ri;
        }

        if(largest!=i)
        {
            int temp = arr[i];
            arr[i] = arr[largest];
            arr[largest] = temp;

            heapify(arr, n, largest);
        }
        }

        void printArray(int[] arr)
        {
                for(int i = 0; i <= arr.length-1; i++)
                {
                        System.out.print(arr[i]+" ");
                }
        }
}
```

```
3 5 9 20 56 64 78
```

Linear Search

- Real time example:
  - Searching for a specific YouTube video
  - Searching a product in amazon or flipkart
- Linear search is a very simple search algorithm. In this type of search, a sequential search is made over all items one by one. Every item is checked and if a match is found then that particular item is returned, otherwise the search continues till the end of the data collection.

```
package SearchingPrograms;//Linear search for numbers

public class LinearSearchNumber
{

        public static void main(String[] args)
        {
                int a[] = {5, 3, 2, 7, 9, 1};
                int item = 3;
                int temp = 0;

                for(int i = 0; i < a.length; i++)
                {
                        if(a[i] == item)
                        {
                                System.out.println("Item is present "+i+" index position");
                                temp = temp + 1;
                        }
                }

                if(temp == 0)
                {
                        System.out.println("Item not found");
                }
        }

}
 Item is present 1 index position
// Linear search for string
package SearchingPrograms;
```

```
public class LinearSearchString
{

        public static void main(String[] args)
        {
                String a[] = {"ashirbad", "deepak", "rahul", "rony", "som"};
                String item = "som";
                int temp = 0;

                for(int i = 0; i < a.length; i++)
                {
                        if(a[i].equals(item))
                        {
                                System.out.println("Item is present at "+i+" index position");
                                temp = temp + 1;
                        }
                }

                if(temp == 0)
                {
                        System.out.println("Item not found");
                }
        }

}
```

```
Item is present at 4 index position
```

Binary search

- Binary search is the process of searching an element from sorted array by repeatedly dividing the search interval in half.
- Binary search is faster than linear search.
- Although binary search is a very optimized way of searching a particular element but the array must be sorted in which you want to perform the search operation process.
- If the array is not sorted in advance then we have to perform sorting first and then only we can perform binary search on that.

```
package SearchingPrograms;

public class BinarySearch
{

        public static void main(String[] args)
        {
                int[] a = {2, 5, 7, 9, 12, 14, 16, 17, 19, 20, 24, 28};
                int srch = 16;
                int li = 0;
                int hi = a.length-1;
                int mi = (li+hi)/2;
        while(li <= hi)
        {
            if(a[mi] == srch)
            {
                System.out.println("Element is at "+mi+" index position");
                break;
            }

            else if(a[mi] < srch)
            {
                li = mi+1;
            }
            else
            {
                hi = mi - 1;
            }

            mi = (li+hi)/2;
        }

        if(li > hi)
        {
            System.out.println("Element not found");
        }
    }

}
```

```
Element is at 6 index position
```

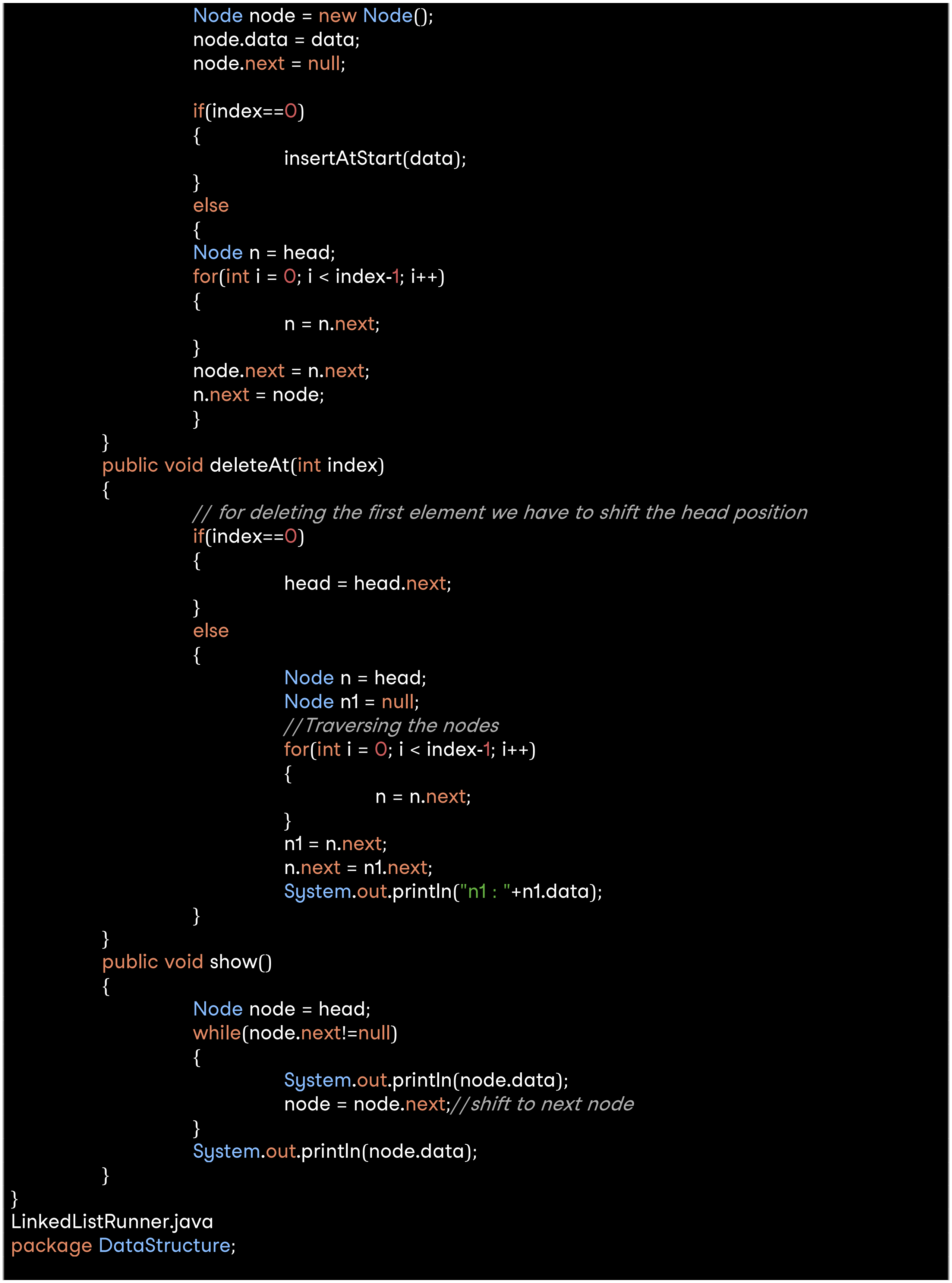
Data Structure

Linked List Implementation:

Ashirbad Swain

```
Node.java
package DataStructure;

public class Node
{
    int data;
    Node next;
}
```

```
LinkedList.java
package DataStructure;
public class LinkedList
{
    Node head; // this will refer to the first node because head is referred as first node

    //Insert a data
    public void insert(int data)
    {
        //creating a node
        Node node = new Node();
        // data will take the value and assign it into the node
        node.data = data;
        //reference to the next node
        node.next = null;

        if(head==null)
        {
            head = node;
        }
        else
        {
            Node n = head;
            //for traversing between the nodes
            while(n.next!=null)// traverse until u got a null value
            {
                n = n.next;//jumping to the next node
            }
            n.next = node;
        }
    }

    public void insertAtStart(int data)
    {
        Node node = new Node();
        node.data = data;
        node.next = null;
        node.next = head;
        head = node;
    }

    public void insertAt(int index,int data)
    {
```



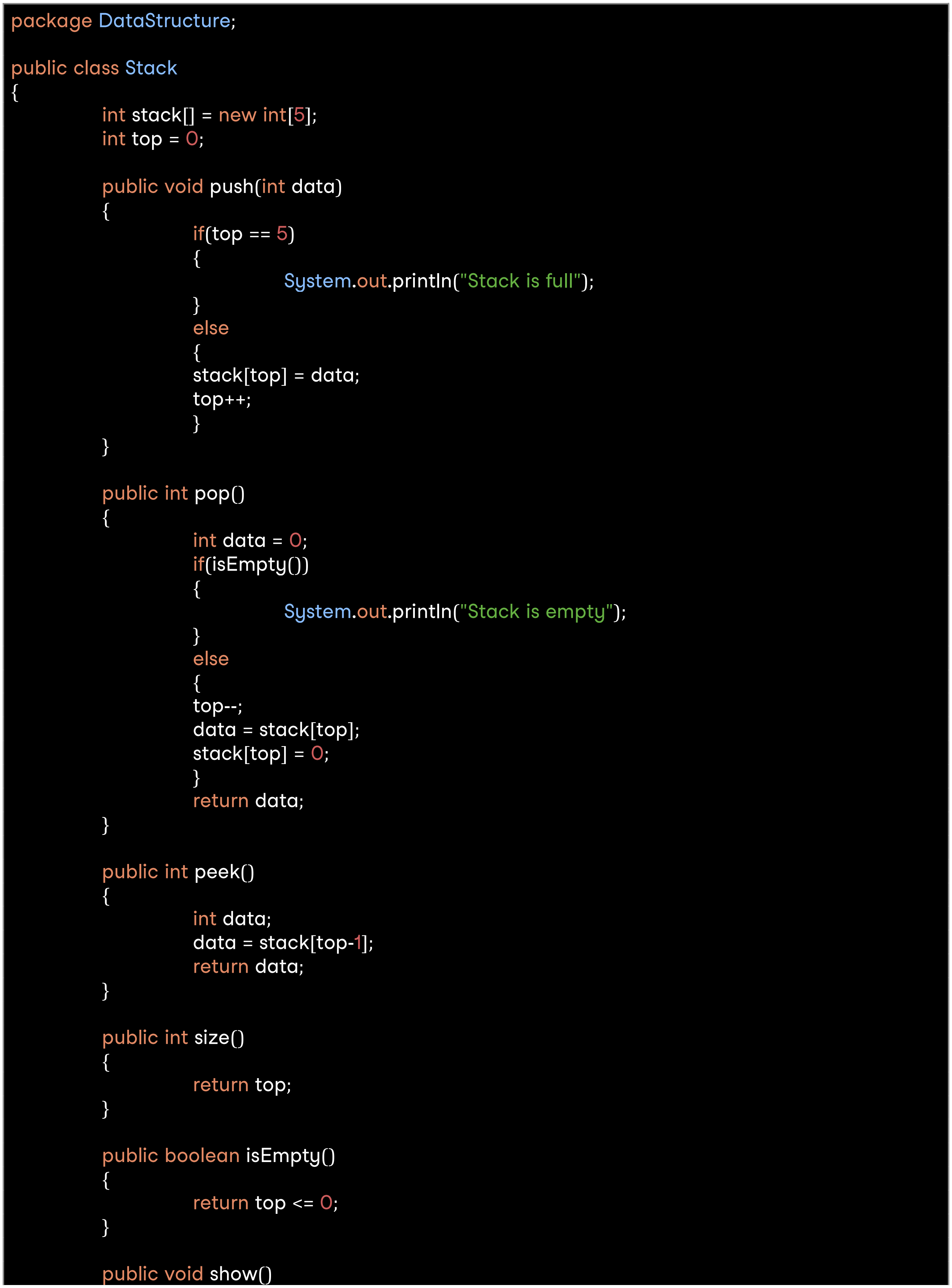
```
        Node node = new Node();
        node.data = data;
        node.next = null;

        if(index==0)
        {
            insertAtStart(data);
        }
        else
        {
        Node n = head;
        for(int i = 0; i < index-1; i++)
        {
            n = n.next;
        }
        node.next = n.next;
        n.next = node;
        }
    }
    public void deleteAt(int index)
    {
        // for deleting the first element we have to shift the head position
        if(index==0)
        {
            head = head.next;
        }
        else
        {
            Node n = head;
            Node n1 = null;
            //Traversing the nodes
            for(int i = 0; i < index-1; i++)
            {
                n = n.next;
            }
            n1 = n.next;
            n.next = n1.next;
            System.out.println("n1 : "+n1.data);
        }
    }
    public void show()
    {
        Node node = head;
        while(node.next!=null)
        {
            System.out.println(node.data);
            node = node.next;//shift to next node
        }
        System.out.println(node.data);
    }
}
LinkedListRunner.java
package DataStructure;
```

```
public class LinkedListRunner
{

        public static void main(String[] args)
        {
                LinkedList list = new LinkedList();

                list.insert(18);
                list.insert(45);
                list.insert(12);

                list.insertAtStart(25);

                list.insertAt(0, 55);

                list.deleteAt(2);

                list.show();
        }

}
```

```
n1 : 18
55
25
45
12
```

Stack:

```
Stack.java
```



```
package DataStructure;

public class Stack
{
        int stack[] = new int[5];
        int top = 0;

        public void push(int data)
        {
                if(top == 5)
                {
                        System.out.println("Stack is full");
                }
                else
                {
                stack[top] = data;
                top++;
                }
        }

        public int pop()
        {
                int data = 0;
                if(isEmpty())
                {
                        System.out.println("Stack is empty");
                }
                else
                {
                top--;
                data = stack[top];
                stack[top] = 0;
                }
                return data;
        }

        public int peek()
        {
                int data;
                data = stack[top-1];
                return data;
        }

        public int size()
        {
                return top;
        }

        public boolean isEmpty()
        {
                return top <= 0;
        }

        public void show()
        {
                for(int n : stack)
                {
                        System.out.print(n + " ");
                }
        }
```

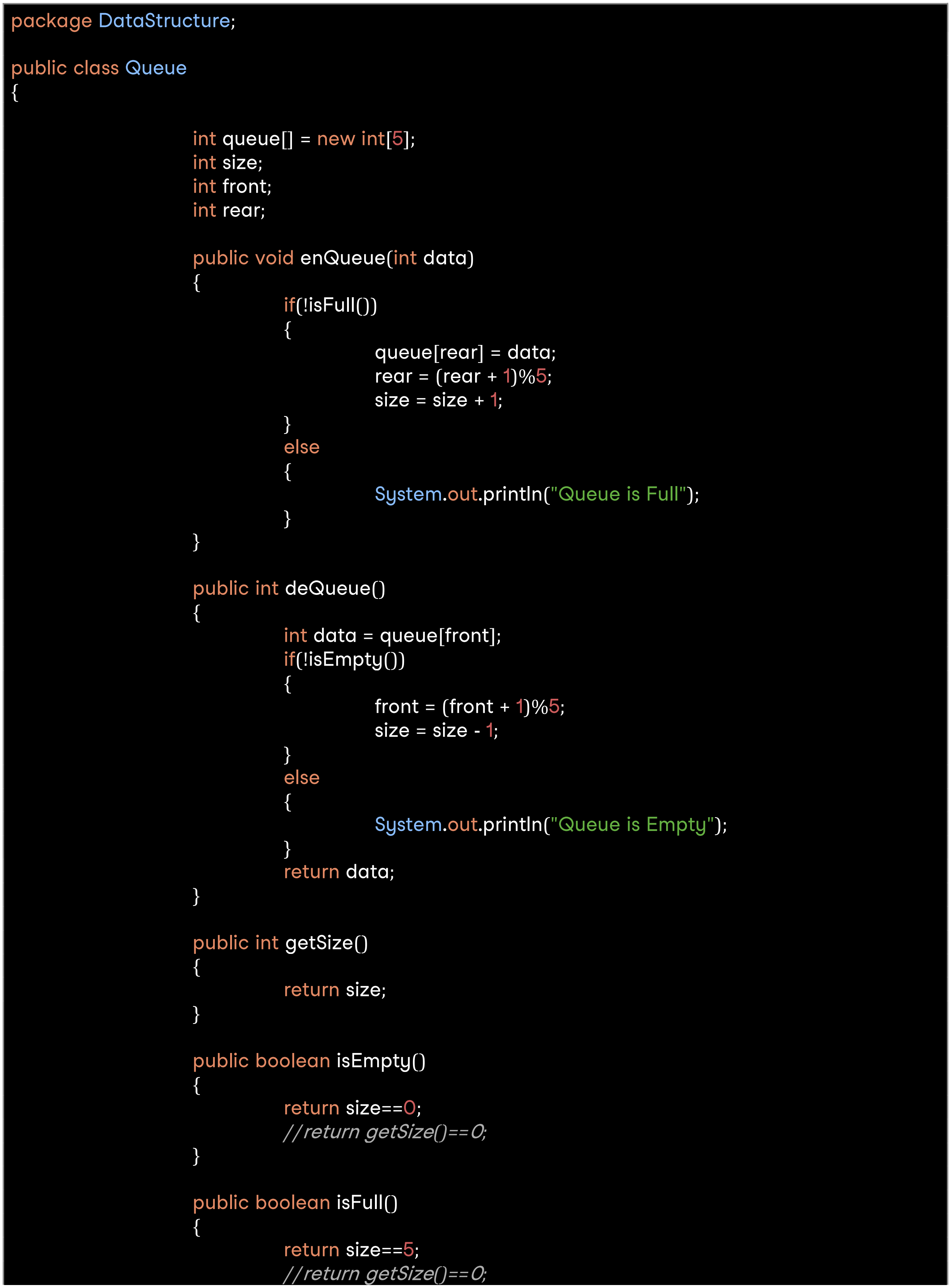
```
StackRunner.java
```

```
package DataStructure;

public class StackRunner
{
        public static void main(String args[])
        {
                Stack nums = new Stack();
                nums.push(15);
                nums.push(8);

                System.out.println(nums.peek());

                nums.push(10);

                System.out.println(nums.pop());

                System.out.println("size is " + nums.size());

                System.out.println("Empty = " + nums.isEmpty());

                nums.show();
        }
}
```

```
8
10
size is 2
Empty = false
15 8 0 0 0
```

Queue:

```
Queue.java
```



```java
package DataStructure;

public class Queue
{
        int queue[] = new int[5];
        int size;
        int front;
        int rear;

        public void enQueue(int data)
        {
            if(!isFull())
            {
                queue[rear] = data;
                rear = (rear + 1)%5;
                size = size + 1;
            }
            else
            {
                System.out.println("Queue is Full");
            }
        }

        public int deQueue()
        {
            int data = queue[front];
            if(!isEmpty())
            {
                front = (front + 1)%5;
                size = size - 1;
            }
            else
            {
                System.out.println("Queue is Empty");
            }
            return data;
        }

        public int getSize()
        {
            return size;
        }

        public boolean isEmpty()
        {
            return size==0;
            //return getSize()==0;
        }

        public boolean isFull()
        {
            return size==5;
            //return getSize()==0;
        }
        public void show()
        {
            System.out.print("Elements : ");
            for(int i = 0; i < size; i++)
            {
                System.out.print(queue[(front+i)%5] + " ");
            }
            System.out.println();

            for(int n : queue)
            {
                System.out.println(n + " ");
            }
        }
    }
```

QueueRunner.java

```
package DataStructure;

public class QueueRunner
{

    public static void main(String[] args)
    {
        Queue q = new Queue();
        q.enQueue(5);
        q.enQueue(2);
        q.enQueue(7);
        q.enQueue(3);

        q.deQueue();
        q.deQueue();

        q.enQueue(9);
        q.enQueue(1);

        System.out.println(q.isFull());

        System.out.println("Size " + q.getSize());
        q.show();
    }

}
```

```
false
Size 4
Elements: 7 3 9 1
1 2 7 3 9
```

OOPs Concept Programming Example :

If statement

```
// it execute the if block if the condition is true
//Java program to demonstrate the use of if statement
package OopsConcept;

public class IfStatement
{

        public static void main (String[] args)
        {
                // defining an 'age' variable
                int age = 20;

                //Checking the age
                if(age > 18)
                {
                        System.out.println("Age is greater than 18");
                }
        }

}
```

```
Age is greater than 18
```

If-else statement

```
// It executes the if block if condition is true otherwise else statement is executed
package OopsConcept;

public class Ifelse
{

        public static void main(String[] args)
        {
                //defining a variable
                int number = 13;
                //check if the number is divisible by 2 or not
                if(number%2 == 0)
                {
                        System.out.println("Even number");
                }
                else
                {
                        System.out.println("Odd number");
                }
        }
}
```

```
Odd number
```

Else - if statement

```
package OopsConcept;

public class NestedIfelse {

            public static void main(String[] args)
            {
                        int number = -13;

                        if(number > 0)
                        {
                                    System.out.println("POSITIVE");
                        }
                        else if(number < 0)
                        {
                                    System.out.println("NEGATIVE");
                        }
                        else
                        {
                                    System.out.println("ZERO");
                        }
            }

}
```

```
NEGATIVE
```

Switch statement

```
package OopsConcept;
public class SwitchStatement {

            public static void main(String[] args)
```

```
        {
                //Declaring a variable for switch expression
                int number = 20;
                // Switch expression
                switch(number)
                {
                //case statements
                case 10: System.out.println("Ten");
                break;
                case 20: System.out.println("Twenty");
                break;
                case 30: System.out.println("Thirty");
                break;

                //Default statement
                default: System.out.println("Not in 10, 20 or 30");
                }
        }

}

Output: Twenty
```

[For loop]

```
package OopsConcept;

public class ForLoop {

        public static void main(String[] args)
        {
                int i;
                for(i = 1; i <= 10; i++)
                {
                        System.out.println(i);
                }
        }

}
```

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

[While loop]

```
package OopsConcept;

public class WhileLoop {

        public static void main(String[] args)
        {
                int i  = 1;
                while(i <= 10)
                {
                        System.out.println(i);
                        i++;
                }
        }

}
```

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

Do-While loop

```
package OopsConcept;
public class DoWhileLoop {

        public static void main(String[] args)
        {
                int i = 1;
                do
                {
                        System.out.println(i);
                        i++;
                }
                while(i < 10);
        }

}
```

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

Break Statement

```
package OopsConcept;

public class BreakStatement {

        public static void main(String[] args)
```

```
    {
        //using for loop
        for(int i = 1; i <= 10; i++)
        {
            if(i == 5)
            {
                //breaking the loop
                break;
            }
            System.out.println(i);
        }
    }

}
```

```
 1 2 3 4
```

Continue Statement

```
package OopsConcept;

public class ContinueStatement {

    public static void main(String[] args)
    {
        //for loop
        for(int i = 1; i <= 10; i++)
        {
            if(i == 5)
            {
                //using continue statement
                continue;// it will skip the rest statement
            }
            System.out.println(i);
        }
    }

}

// as you can see the output, 5 is not printed on the console. it is because the loop is continued
when it reaches to 5
```

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

Object and Class Example

```
//Java Program to illustrate object and class
package OopsConcept;

//Creating a class named ObjectClassEx
public class ObjectClassEx
```

```
{
        // Defining field or data member or instance variable
                        int id;
                        String name;

        public static void main(String[] args)
        {
                // Creating an object or instance of class ObjectClassEx
                ObjectClassEx s1 = new ObjectClassEx();

                //Printing values of the object
                System.out.println(s1.id);//accessing member through reference variable
                System.out.println(s1.name);
        }

}
```

```
0
null
```

Initialization of an object (3 ways)

```
package OopsConcept;

//Object Initialization through reference variable by dot operator
class Animal
{
        String color;
        int age;
        public static void main(String[] args)
        {
                Animal dog = new Animal();
                dog.color() = "black";
                dog.age() = 10;
                System.out.println(dog.color+" "+dog.age);
        }
}
```

```
black 10
```

```
package OopsConcept;

//Object Initialization through method
class Animal
{
        String color;
        int age;
        void initObj(String c, int a)
        {
                color = c;
                age = a;
        }
        void display()
        {
                System.out.println(color+" "+age);
        }
        public static void main(String[] args)
        {
                Animal dog = new Animal();
                dog.initObj("black",10);
                dog.display();
        }
}
```

```
black 10
```

Constructor Example (Default & Parameterized Constructor)

```
//Java Program to create and call a Default Constructor
class Bike1
{
        //Creating a default constructor
        Bike1()
        {
                System.out.println("Bike is created");
        }
        public static void main(String args[])
        {
                //calling the default constructor
                Bike1 b = new Bike1();
        }
}
```

```
 Bike is created
```

```
package OopsConcept;

//Java Program to create and call a parameterized constructor
class Student
{
        int id;
        String name;
        //Creating a parameterized constructor
        Student(int i, String n)
        {
                id = i;
                name = n;
        }

        //Method to display the values
        void display()
        {
                System.out.println(id+" "+name);
        }

        public static void main(String args[])
        {
                //creating objects and passing values
                Student s1 = new Student(111, "Karan");
                Student s2 = new Student(222, "Aryan");
                s1.display();
                s2.display();
        }
}
```

```
111 Karan
222 Aryan
```

Constructor Overloading

```
//Java Program to overload constructors
class Student
{
        int id;
        String name;
        int age;

        //creating two arg. constructor
        Student(int i, String n)
        {
                id = i;
                name = n;
        }

        //creating three arg. constructor
        Student(int i, String n, int a)
        {
                id = i;
                name = n;
                age = a;
        }

        void display()
        {
                System.out.println(id+" "+name+" "+age);
        }

        public static void main(String args[])
        {
                Student s1 = new Student(111, "Kiran");
                Student s2 = new Student(222, "Aryan", 25);
                s1.display();
                s2.display();
        }
}
```

```
111 Karan 0
222 Aryan 25
```

Static Keyword Example:(as Variable, Method and Block)

```
//Java program for static variable
class Test
{
        //Declaring a static variable
        static int a = 10;
}

class Demo
{
        public static void main(String args[])
        {
                System.out.println(Test.a);
        }
}
```

```
10
```

```
package OopsConcept;

//Java Program for common static variable
class Student
{
        int studentID;
        String student_name;
        static String student_clg = "NIT";

        Student(int studentID, String student_name)
        {
                this.studentID = studentID;
                this.student_name = student_name;
        }

        void display()
        {
                System.out.println(studentID+" "+student_name+" "+student_clg);
        }

        public static void main(String args[])
        {
                Student s1 = new Student(101, "Ashirbad");
                s1.display();
                Student s2 = new Student(102, "Abinash");
                s2.display();
        }
}
```

```
101 Ashirbad NIT
102 Abinash NIT
```

Static Method

```
package OopsConcept;

//Java program for static method
class StaticMethod
{
        static void display()
        {
                System.out.println("1");
        }

        public static void main(String args[])
        {
                display();
                // or StaticMethod.display();
        }
}
Output: 1
```

Static Block

```
package OopsConcept;

//Java Program for static block
public class StaticBlock
{
        static
        {
                System.out.println("Hello");
        }
```

```
	public static void main(String args[])
	{
		System.out.println("I am in main method");
	}

	static
	{
		System.out.println("I am second static block");
	}
}
```

```
Hello
I am second static block
I am in main method
```

Inheritance Example

Single Inheritance

```
//Single Inheritance Example
class Animal
{
	void eat()
	{
		System.out.println("eating...");
	}
}

class Dog extends Animal
{
	void bark()
	{
		System.out.println("barking...");
	}
}

class TestInheritance
{
	public static void main(String args[])
	{
		Dog d = new Dog();
		d.bark();
		d.eat();
	}
}
```

```
barking...
eating...
```

Multilevel Inheritance

```
// Multilevel Inheritance
class Animal
{
        void eat()
        {
                System.out.println("eating...");
        }
}
class Dog extends Animal
{
        void bark()
        {
                System.out.println("barking...");
        }
}
class BabyDog extends Dog
{
        void weep()
        {
                System.out.println("Weeping...");
        }
}
class TestInheritance
{
        public static void main(String args[])
        {
                BabyDog d = new BabyDog();
                d.weep();
                d.bark();
                d.eat();
        }
}
```

```
weeping...
barking...
eating...
```

Hierarchical Inheritance

```
class Animal
{
        void eat()
        {
                System.out.println("eating...");
        }
}
class Dog extends Animal
{
        void bark()
        {
                System.out.println("barking...");
        }
}
class Cat extends Animal
{
        void meow()
        {
                System.out.println("meowing...");
        }
}

class TestInheritance
{
        public static void main(String args[])
        {
                Cat c = new Cat();
                c.meow();
                c.eat();
                c.bark();// Error
        }
}
```

```
meowing...
eating...
```

Multiple Inheritance (Diamond Problem)

```
// Diamond problem
class A
{
        void msg ()
        {
                System.out.println("Hello");
        }
}
class B
{
        void msg ()
        {
                System.out.println("Welcome");
        }
}
class c extends A, B
{
        public static void main (String args[])
        {
                C obj = new C();
                //Now which msg() method would be invoked?
                obj.msg();
        }
```

```
Compile Time Error
```

Method Overloading

```
package OopsConcept;

public class MethodOverloading
{
        void show(int a)
        {
                System.out.println("1");
        }
        void show(String b)
        {
                System.out.println("2");
        }

        public static void main(String args[])
        {
                MethodOverloading mo = new MethodOverloading();
                mo.show(1);
        }
}
```

```
1
```

Method Overriding

```
package OopsConcept;

class p
{
        public void property()
        {
                System.out.println("cash + land + gold");
        }
        public void marry()
        {
                System.out.println("Ananya");
        }
}

class c extends p
{
        public void marry()
        {
                System.out.println("Abhipsa");
        }
        public static void main(String args[])
        {
                c c1 = new c();
                c1.marry();
        }
}
```

```
Abhipsa
```

Super Keyword

```
class A
{
        int i = 10;
}
class B extends A
{
        int i = 20;
        void show(int i)
        {
```

```
        //super keyword
        System.out.println(super.i);
        //this keyword
        System.out.println(this.i);
        //normal
        System.out.println(i);
    }
    public static void main(String args[])
    {
        B b = new B();
        b.show(30);
    }}
```

```
Output: 10 20 30
```

Final keyword (variable, method, class)

```
//final keyword used for variable
class Test
{
    public static void main(String args[])
    {
        final int i = 10;
        i = i + 20;// it shows error because of final keyword
        System.out.println(i);
    }
}
```

```
//final keyword used for method
class Test
{
    final void m1()
    {
        System.out.println("I am in class Demo");
    }
    class Test extends Demo
    {
        void m1()
        {
            System.out.println("I am in class Test");
        }
        public static void main(String args[])
        {
            //code
        }
    }
}
```

```
//final keyword used for class
```

```
final class Demo
{
        //Code snippet
}
class Test extends Demo//here inheritance don't work due to final keyword
{
        public static void main(String args[])
        {
                //code
        }
}
```

Upcasting

```
Class A {}
Class B extends A {}
A a = new B (); //upcasting
```

Down Casting

```
Dog d = new Animal();//Compilation error
Dog d = (Dog)new Animal();
//Compiles successfully but ClassCastException is thrown at runtime

But if we use the Instanceof operator it runs successfully.
```

Runtime Polymorphism

```
class Bike
{
        void run()
        {
                System.out.println("running");
        }
}
class Splendor extends Bike
{
        void run()
        {
                System.out.println("running fast");
        }
        public static void main(String args[])
        {
                Bike b = new Splendor();//upcasting
                b.run();
        }
}
```

```
Running fast
```

Static Binding

```
//Static Binding
class Dog
{
        private void eat()
        {
                System.out.println("dog is eating...");
        }
        public static void main(String args[])
        {
                Dog d1 = new Dog();
                d1.eat();
        }

}
```

```
Output: dog is eating...
```

Dynamic Binding

```
//Dynamic Binding
class Animal
{
        void eat()
        {
                System.out.println("animal is eating...");
        }
}
class Dog extends Animal
{
        void eat()
        {
                System.out.println("dog is eating...");
        }
        public static void main(String args[])
        {
                Animal a = new Dog();
                a.eat();
        }
}
```

```
Output: dog is eating...
```

Java Abstraction

```
abstract class Vehicle
{
        abstract void start();
}
```

```
class Car extends Vehicle
{
        void start()
        {
                System.out.println("car starts with keys");
        }
}
class Scooter extends Vehicle
{
        void start()
        {
                System.out.println("Scooter starts with kick");
        }
        public static void main(String args[])
        {
                Car c = new Car();
                c.start();

                Scooter s = new Scooter();
                s.start();
        }
}
```

```
Car starts with keys
Scooter starts with kick
```

Interface

```
interface printable
{
        void print();
}
class A implements printable
{
        public void print()
        {
                System.out.println("Hello");
        }
        public static void main(String args[])
        {
                A obj = new A();
                obj.print();
        }
}
```

```
Hello
```

```
//This example shows the implementation of bank interface
interface Bank
{
        float rateOfInterest();
}
class SBI implements Bank
{
        public float rateOfInterest()
        {return 9.15f;}
}
class PNB implements Bank
{
        public float rateOfInterest()
        {return 9.7f;}
}
class TestInterface
{
        public static void main(String args[])
        {
                Bank b = new SBI();
                System.out.println("ROI:"+b.rateOfInterest());
        }
}
```

```
ROI: 9.15
```

Multiple Inheritance supported by Interface

```
Interface I1
{
        void show();
}
Interface I2
{
        void display();
}
class Test implements I1,I2
{
        public void show()
        {System.out.println("1");}
        public void display()
        {System.out.println("2");}
```

```
    public static void main(String args[])
    {
        Test t = new Test();
        t.show();
        t.display();
    }
}
```

```
1 2
```

Encapsulation

```
class Employee
{
    private int empid;
    public void setEmpid(int eid)
    {
        empid = eid;
    }
    public int getEmpid()
    {
        return empid;
    }
}
class Company
{
    public static void main(String args[])
    {
        Employee e = new Employee();
        e.setEmpid(101);
        System.out.println(e.getEmpid());
    }
}

Output: 101
```

Autoboxing

```
//Java program to convert primitive into objects
//Autoboxing example of int to Integer
public class Autoboxing
{
    public static void main(String args[])
    {
        int a = 20;
        //converting int into Integer explicitly
        Integer i = Integer.valueOf(a);
        //autoboxing, now compiler will write Integer.valueOf(a) internally
        Integer j = a;

        System.out.println(a+" "+i+" "+j);
    }
}
```

```
20 20 20
```

Unboxing

```
//Java program to convert object into primitives
//Unboxing example of Integer to int
public class Unboxing
{
		public static void main(String args[])
		{
				Integer a = new Integer(3);
				//converting Integer to int explicitly
				int i = a.intValue();
				//Unboxing, now compiler will write a.intValue() internally
				int j = a;

				System.out.println(a+" "+i+" "+j);
		}
```

```
3 3 3
```

Call by value in Java

```
//Example of call by value
class Operation
{
		int data = 50;

		void change(int data)
		{
				//changes will be in the local variable only
				data = data+100;
		}
		public static void main(String args[])
		{
				Operation op = new Operations();
```

```
                                System.out.println("before change "+op.data);
                                op.change(500);
                                System.out.println("after chnage "+op.data);
                }
}
```

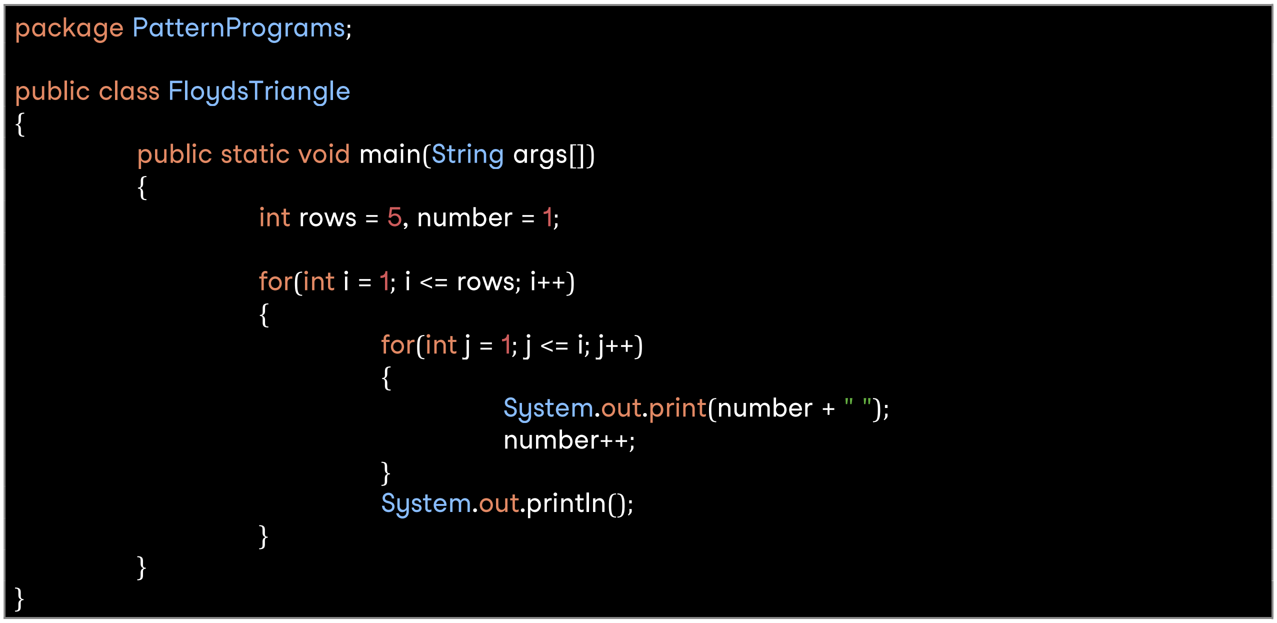
```
Output: before change 50
        After change 50
```

Pascal's Triangle

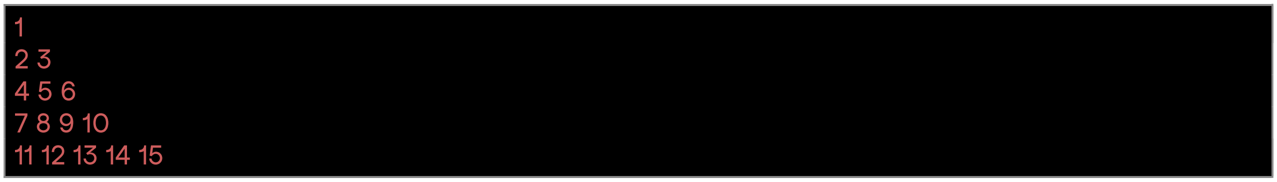
```
package PatternPrograms;

import java.util.Scanner;
public class PascalsTriangle {

  public static void main(String[] args)
{
    int no_row,c=1,blk,i,j;
    System.out.print("Input number of rows: ");
    Scanner in = new Scanner(System.in);
                                  no_row = in.nextInt();

    for(i=0;i<no_row;i++)
    {
        for(blk=1;blk<=no_row-i;blk++)
        System.out.print(" ");
        for(j=0;j<=i;j++)
        {
            if (j==0||i==0)
                c=1;
            else
               c=c*(i-j+1)/j;
            System.out.print(" "+c);
        }
        System.out.print("\n");
        in.close();
    }
}
}
```

```
Input number of rows: 5
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
```

Floyd's Triangle



```
package PatternPrograms;

public class FloydsTriangle
{
	public static void main(String args[])
	{
		int rows = 5, number = 1;

		for(int i = 1; i <= rows; i++)
		{
			for(int j = 1; j <= i; j++)
			{
				System.out.print(number + " ");
				number++;
			}
			System.out.println();
		}
	}
}
```



```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

Write a program to Print 1 to N numbers?

```
class Printnums
{
        public static void main (String[] args)
        {
                java.util.Scanner sc = new java.util.Scanner (System.in);
                System.out.println ("enter value of n");
                int n = sc.nextInt();
                for (int i = 1; i<=n ; i++)
                {
                        System.out.println (i);
                }
        }
}
```

OUTPUT:

```
enter value of n: 10
1
2
3
4
5
6
7
8
9
10
```

Write a program to Print REVERSE of N to 1 numbers?

```
class Printnums
{
        public static void main(String[] args)
        {
                java.util.Scanner sc = new java.util.Scanner(System.in);
                System.out.println ("enter value of n");
                int n=sc.nextInt();
                for(int i=n ;i>=1;i--)
                {
                        System.out.print(i);
                }
        }
}
```

OUTPUT:

```
enter value of n: 10
10 9 8  7 6 5 4 3 2 1
```

Write a program to display sum of 1 to N numbers?

```
class Sumnum
{
        public static void main(String[] args)
        {
                java.util.Scanner sc=new java.util.Scanner(System.in);
                System.out.println("enter value of n");
                int n=sc.nextInt();
                int sum=0;
                for(int i=1;i<=n ;i++)
                {
                        sum+=i;
                }
                System.out.println(sum);
        }
}
```

OUTPUT:

```
enter value of n: 10
55
```

Write a program to check given number is EVEN or ODD?

```
class EvenOdd
{
        public static void main(String[] args)
        {
                java.util.Scanner sc=new java.util.Scanner(System.in);
                System.out.println("enter the num");
                int n=sc.nextInt();
                if(n%2==0)
                        System.out.println(n+” is  even");
                else
                        System.out.println(n+" is odd");
        }
}
```

OUTPUT:

```
enter the num: 20
20 is even

F:\Practice>java Even(Command prompt)
enter the num: 11
11 is odd
```

Write a program to display PRIME NUMBERS from 1 to n?

```
class Prime
{
    public static void main (String [] args)
    {
        java.util.Scanner sc=new java.util.Scanner (System.in);
        System.out.println ("enter number");
        int n=sc.nextInt ();
         System.out.println ("Prime numbers between 1 and " + n);
    //loop through the numbers one by one
     for (int i=1; i < n; i++)
    {
           boolean isPrime = true;
            //check to see if the number is prime
            for (int j=2; j < i ; j++)
             {
                   if (i % j == 0)
                       {
                     isPrime = false;
                      break;
                        }
             }
            // print the number
            if (isPrime)
                   System.out.print (i + " ");
    }
    }
}
```

OUTPUT:

```
enter number
25
Prime numbers between 1 and 25
1 2 3 5 7 11 13 17 19 23
```

Write a program to check whether the given number is PRIME or not?

```
class Prime
{
    public static void main(String[] args)
    {
        java.util.Scanner sc=new java.util.Scanner(System.in);
        System.out.println("enter number");
        int n=sc.nextInt();
        int i;
        if(n==1)
        {
        System.out.println("Prime starts from 2");
        }
        for(i=2;i<n ;i++)
        {
            if(n%i==0)
                System.out.println("not a prime");
            break;
        }
        if(n==i)
            System.out.println("prime");

    }
}
```

OUTPUT:

```
Enter the number : 17
Prime
```

Write a program to find SUM OF PRIME numbers?

```
import java.util.Scanner;

public class SumofPrime
{
    public static void main(String[] args)
    {
        Scanner scn=new Scanner(System.in);
        System.out.println("Enter the range to print sum of prime Nos.....");
        int range=scn.nextInt();
        int sum=0;
        for(int i=1;i<=range ;i++)
        {
            if(isPrime(i))
            sum=sum+i;
        }
        System.out.println(sum);

    }

    public static boolean isPrime(int num)
    {
        if(num==1) return false;
        for(int i=2;i<num ;i++)
        {
            if(num%i==0)
            {
                return false;
            }

        }
        return true;
}
}
```

OUTPUT:

```
Enter the range to print sum of prime Nos.....
10
17
```

Write a program to display MULTIPLICATION table?

```
class Multiplication
{
        public static void main(String[] args)
        {
                java.util.Scanner sc=new java.util.Scanner(System.in);
                System.out.println("enter value of n");
                int n=sc.nextInt();
                for(int i=1;i<=10;i++)
                {
                        System.out.println(n+"*"+i+"="+(n*i));
                }
        }
}
```

Output:

```
enter value of n: 2
2*1=2
2*2=4
2*3=6
2*4=8
2*5=10
2*6=12
2*7=14
2*8=16
2*9=18
2*10=20
```

Write a program to display MULTIPLICATION TABLES?

```
class Tables
{
        public static void main(String[] args)
        {
                java.util.Scanner sc=new java.util.Scanner(System.in);
                System.out.println("enter value of n");
                int n=sc.nextInt();
                for(int i=1;i<=n ;i++)
                {
                        for (int j=1;j<=10 ;j++ )
                        {
                                System.out.print(j+"*"+i+"="+j*i+"\t");
                        }
                }
                System.out.println();
        }
}
```

OUTPUT:

```
enter value of n: 5

1*1=1   2*1=2   3*1=3   4*1=4   5*1=5
1*2=2    2*2=4          3*2=6   4*2=8   5*2=10
1*3=3   2*3=6   3*3=9   4*3=12          5*3=15
1*4=4   2*4=8   3*4=12          4*4=16          5*4=20
1*5=5   2*5=10          3*5=15          4*5=20          5*5=25
```

```
1*6=6   2*6=12  3*6=18          4*6=24          5*6=30
1*7=7   2*7=14  3*7=21  4*7=28          5*7=35
1*8=8   2*8=16          3*8=24          4*8=32          5*8=40
1*9=9   2*9=18          3*9=27          4*9=36          5*9=45
1*10=10         2*10=20         3*10=30         4*10=40         5*10=50
```

Write program weather the number is PERFECT NUMBER or not?

Def:
Perfect number, a positive integer that is equal to the sum of its proper divisors. The smallest perfect number is 6, which is the sum of 1, 2, and 3.

```
import java.util.*;
class  Perfectnumber
{
        public static void main(String[] args)
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("enter a number");
                int num=sc.nextInt();
                int sum=1;
                for (int i=2;i<=num/2;i++ )
                {
                        if (num%i==0)
                        sum=sum+i;
                }
                if (sum==num)
                {
                        System.out.println(num+"is a  Perfect number");
                }
                else
                System.out.println(num+" is not a Perfect number");
        }
}
```

OUTPUT:

```
enter a number
6
6 is a Perfect number
```

Write a program to display RANGE of PERFECT NUMBERS?

```
import java.util.*;
class  Rangeperfectnumber
{
        public static void main(String[] args)
```

```
	{
		Scanner sc=new Scanner(System.in);
		System.out.println("enter a number");
		int n=sc.nextInt();
		for(int num=1;num<=n; num++)
		{
		int sum=1;
		for (int i=2;i<=num/2;i++ )
		{
			if (num%i==0)
			sum=sum+i;
		}
		if (sum==num)
		{
			System.out.println(num+"is a  Perfect number");
		}
		}
	}
}
```

OUTPUT:

```
enter a number
100
1is a  perfect number
6is a  perfect number
28is a  perfect number
```

Write a program to check the given number is PALINDROME or not?

```
import java.util.*;
class Palindrome
{
	public static void main(String[] args)
	{
		Scanner sc=new Scanner(System.in);
		System.out.println("enter a number");
		int n =sc.nextInt();
		int t=n;
		int rev=0;
		while (n!=0)
		{
			rev=rev*10+(n%10);
			n=n/10;
		}
		if (rev==t)
			System.out.println(t+" is a palindrome number");
```

```
                    else
                    System.out.println(t+" is not a palindrome number");


          }
}
```

OUTPUT:

```
enter a number
121
121 is a palindrome number

enter a number
143
143 is not a palindrome number
```

Write a program to find the FACTORIAL of a given number?

```
import java.util.*;
class Factorial
{
          public static void main(String[] args)
          {
                    Scanner scn=new Scanner(System.in);
                    System.out.println("enter the number");
                    int n=scn.nextInt();
                    int fact=1;
                    for (int i=1;i<=n ;i++ )
                    {
                              fact=fact*i;
                    }
                    System.out.println(fact);
          }
}
```

OUTPUT:

```
Enter the number
5
120
```

Write a program to find the FACTORIAL of a given RANGE of numbers?

```
import java.util.*;
class FactRange
{
          static int fact(int n)
          {
                    int fact=1;
                    while (n>0)
                    {
                              fact=fact*n;
```

```
                n--;
            }
            return fact;
    }
    public static void main(String[] args)
    {
            Scanner scn=new Scanner(System.in);
            System.out.println("enter the factorial range number");
            int k=scn.nextInt();
            for (int i=1;i<=k ;i++)
            {
                System.out.println(i+"!---->"+fact(i));
            }
    }
}
```

OUTPUT:

```
enter the factorial range number :7
1!---->1
2!---->2
3!---->6
4!---->24
5!---->120
6!---->720
7!---->5040
```

Write program to check the given number is STRONG or not?

Def: Strong numbers are the **numbers** whose sum of factorial of digits is equal to the original **number.**
Example: 145 is a **strong number.**

```
import java.util.*;
class Strongnumber
{
    static int fact(int n)
    {
            int fact=1;
            while (n>0)
            {
                fact= fact*n;
                n--;
            }
            return fact;
    }
    public static void main(String[] args)
    {
            Scanner sc=new Scanner(System.in);
            System.out.println("enter a number");
            int n =sc.nextInt();
            int num=n;
            int sum=0;
            int t=num;
            while (num!=0)
```

```
        {
            int r=num%10;
            sum=sum + fact(r);
            num=num/10;
            }

        if (sum==t)
            System.out.println(t+" is a strong number");
        else
            System.out.println(t+" not a strong number");
    }
}
```

OUTPUT:

```
enter a number
143
143not a strong number
```

Write program weather to find range of STRONG NUMBER?

```
import java.util.*;
class Strongnumber
{
    static int fact(int n)
    {
        int fact=1;
        while (n>0)
        {
            fact= fact*n;
            n--;
        }
        return fact;
    }
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter a Range");
        int n =sc.nextInt();
        for (int i=1;i<=n ;i++ )
        {
        int num=i;
        int sum=0;
        int t=num;
        while (num!=0)
        {
            int r=num%10;
            sum=sum + fact(r);
            num=num/10;
            }

        if (sum==t)
            System.out.println(t+ " is a strong number");
        }
    }
}
```

OUTPUT:

```
enter a Range
145
1is a strong number
2is a strong number
145 is a strong number
```

Write a program to display FIBONACCI series of a number?

Def: a series of numbers in which each number (*Fibonacci number*) is the sum of the two preceding numbers. The simplest is the series 1, 1, 2, 3, 5, 8, etc.

```
class Fibonacci
{
    static int fib(int n)
    {
        if(n==0)
            return 0;
        if(n==1)
            return 1;
        return fib(n-1)+fib(n-2);
    }
    public static void main(String[] args)
    {
        java.util.Scanner sc=new java.util.Scanner(System.in);
        System.out.println("Enter the number");
        int m=sc.nextInt();
        int f=fib(m);
        System.out.println(f);
    }
}
```

OUTPUT:

```
Enter the number
10
55
```

Write a program to display range of FIBONACCI numbers?

```
import java.util.Scanner;
public class FibonacciSeries1
{
        public static void main(String[] args)
        {
            Scanner scn=new Scanner(System.in);
            System.out.println("enter the range:.........");
            int range=scn.nextInt();
          int a=0;
          int b=1;
          int c=0;
          System.out.print(a);
          System.out.print(b);
          for (int i = 2; i <=range; i++)
          {
            c=a + b;
            if(c<=range)
            {

            //c=a + b;
            System.out.print(c);
            a=b;
            b=c;
            }
         }

        }
}
```

OUTPUT:

```
Enter the range….
50
0 1  1 2 3 5 8 13 21 34
```

Write a program to REVERSE the number?

```
import java.util.Scanner;
class Reversenum
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the number");
        int num=sc.nextInt();
        int t=num;
        int rev=0;

        while(num!=0)
        {
             rev = rev*10+(num%10);
            num = num/10;
        }
        System.out.println(rev);
    }
}
```

OUTPUT:

```
enter the number
105
501
```

Write a program to display GCD of two numbers?

```
import java.util.Scanner;
class Gcd
{
    static int gcd(int m ,int n)
    {
        if(m<n)
            return gcd(n ,m);
        if(n==0)
            return m;
        return gcd(n, m%n);
    }
    public static void main(String[] args)
    {
    Scanner sc = new Scanner(System.in);
    System.out.println(" Enter the two numbers");
    int p = sc.nextInt();
    int q = sc.nextInt();
        int a=gcd(p, q);
        System.out.println(a);
    }
}
```

OUTPUT:

```
Enter the two numbers
90
120
30
```

Write a program to check the given number is PRIME PALINDROME or not?

```
import java.util.*;
class Palindrome
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter a number");
        int n =sc.nextInt();
        int t=n;
        int rev=0;
        int i;
        while (n!=0)
        {
            rev=rev*10+(n%10);
            n=n/10;
        }
        if (rev==t)
        {
            for( i=2;i<rev ;i++)
            {
                if(rev % i==0)
                {
                    System.out.println("not a prime palindrome");
                break;
                }
            }
            if(rev==i)
            System.out.println(t+ "is a prime palindrome number");
        }
        else
        System.out.println(t+ "is not a prime palindrome number");

    }
}
```

OUTPUT:

```
enter a number
313
313 is a prime palindrome number

enter a number
103
103 is not a prime palindrome number
```

Write a Program to check the given number is ARMSTRONG or not?

Def: An Armstrong number is an integer such that the sum of the power of its digits is equal to the number itself.
For example, 371 is an Armstrong number since 3**3 + 7**3 + 1**3 = 371.
9 is an Armstrong number since 9*1= 9.

```
import java.util.Scanner;
public class Armstrong1
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the number");
        int n=sc.nextInt();
        boolean r=isArmstrong(n);
                if(r)
                    System.out.println("Given num is Armstrong");
                else
                    System.out.println("Given num is not Armstrong");
    }
    static int countDigit(int num)
    {
        int count=0;
        while(num>0)
        {
            count++;
            num=num/10;
        }
        return count;
    }
    static int pow(int n, int p)
    {
        int pw=1;
        while(p>0)
        {
            pw=pw*n;
            p--;
        }
        return pw;
    }
    static boolean isAmstrong(int x)
    {
        int nd=countDigit(x);
        int t=x;
        int sum=0;
        while(t>0)
        {
            int r=t%10;
            sum=sum+ pow(r ,nd);
            t=t/10;
        }
        if(sum==x)
            return true;
        else
            return false;
    }

}
```

OUTPUT:

```
enter the number
153
Given num is Armstrong
enter the number
1
Given num is Armstrong
```

Write a Program to display the range of ARMSTRONG numbers?

```
import java.util.Scanner;
public class Armstrong2
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the number");
        int n=sc.nextInt();
        for (int i=0;i<=n ;i++ )
        {

        boolean r=isAmstrong(i);
                if(r)
                    System.out.println(i +" is Armstrong");
        }
    }
    static int countDigit(int num)
    {
        int count=0;
        while(num>0)
        {
            count++;
            num=num/10;
        }
        return count;
    }
    static int pow(int n ,int p)
    {
        int pw=1;
        while(p>0)
        {
            pw=pw*n;
            p--;
        }
        return pw;
    }
    static boolean isAmstrong(int x)
    {
        int nd=countDigit(x);
        int t=x;
        int sum=0;
        while(t>0)
        {
            int r=t%10;
            sum=sum +pow(r ,nd);
            t=t/10;
        }
        if(sum==x)
            return true;
        else
            return false;
    }
}
```

OUTPUT:

```
enter the number: 300
0 is Armstrong
1 is Armstrong
2 is Armstrong
3 is Armstrong
4 is Armstrong
5 is Armstrong
6 is Armstrong
7 is Armstrong
8 is Armstrong
9 is Armstrong
153 is Armstrong
```

Write a program to Swap two numbers without using 3rd variable?

```
class Swap
{
        public static void main(String[] args) {
                int i=10;
                int j=20;
                i=i + j;
                j=i-j;
                i=i-j;
                System.out.println(“i=”+i);
                System.out.println(“j=”+j);
        }
}
```

OUTPUT:

```
i=20
j=10
```

Write a program to Swap two numbers with using 3rd variable?

```
class Swapv
{
        public static void main(String[] args)
        {
                int i=10;
                int j=20;
                int k;
                k=i;
                i=j;
                j=k;
                System.out.println(“i=”+i);
                System.out.println(“j=”+j);
        }
}
```

OUTPUT:

```
i=20
j=10
```

NUMBER CONVERSIONS

Write a program to convert BINARY to DECIMAL?

```
import java.util.*;
public class Bintodec
{
	public static void main(String[] args)
{
		System.out.println("enter the binary number");
		Scanner sc=new Scanner(System.in);
	long n =sc. nextLong();
	long dec=0;
	int count=0;
	while(n>0)
	{
		long r=n%10;
		dec=dec +r*pow(2,count);
		count++;
		n/=10;
	}
	System.out.println("decimal Equivalent:" +dec);
   }
	static int pow(int n, int p)
	{
	int pw=1;
	while(p>0)
	{
		pw=pw*n;
		p--;
	}
	return pw;
	}

}
```

OUTPUT:

```
enter the binary number
111100001111
decimal Equivalent:3855
```

Write a program to convert DECIMAL to BINARY?

```
import java.util.*;
public class Dectobin
{
public static void main(String[] args)
{
	System.out.println("enter the decimal number");
	Scanner sc=new Scanner(System.in);
```

```
        int n=sc.nextInt();
        String bin="";
        while(n>0)
        {
        int r=n%2;
        bin= r + bin;
        n=n/2;

        }
        System.out.println("Binary Equivalent:" + bin);
}
}
```

OUTPUT:

```
enter the decimal number
3855
Binary Equivalent:111100001111
```

Write a program to convert OCTAL to DECIMAL?

```
import java.util.*;
public class Octtodec
{
        public static void main(String[] args)
{
                System.out.println("enter the octal number");
                Scanner sc=new Scanner(System.in);
        int n =sc.nextInt();
        int dec=0;
        int count=0;
        while(n>0)
        {
                int r=n%10;
                dec=dec + r*pow(8,count);
                count++;
                n/=10;
        }
        System.out.println("decimal Equivalent:" +dec);
  }
        static int pow(int n, int p)
        {
        int pw=1;
        while(p>0)
        {
                pw=pw*n;
                p--;
        }
        return pw;
        }

}
```

OUTPUT:

```
enter the octal number
763
decimal Equivalent:499
```

Write a program to convert DECIMAL to OCTAL?

```
import java.util.*;
public class Dectooct
{
public static void main(String[] args)
{
        System.out.println("enter the decimal number");
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        String oct="";
        while(n>0)
        {
        int r=n%8;
        oct= r + oct;
        n=n/8;
        }
        System.out.println("Octal Equivalent:" + oct);
}
}
```

OUTPUT:

```
enter the decimal number
56
Octal Equivalent:70
```

Write a program to convert DECIMAL to HEXADECIMAL?

```
import java.util.*;
public class Dectohex
{
        public static void main(String[] args)
{
        System.out.println("enter the decimal number");
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        String hex="";
        while(n>0)
        {
                int r=n%16;
        switch (r)
        {
        case 10: hex='A'+ hex;
                break;
        case 11: hex='B'+ hex;
                        break;
        case 12: hex='C'+ hex;
```

```
            break;
    case 13: hex='D'+ hex;
            break;
    case 14: hex='E'+ hex;
            break;
    case 15: hex='F'+ hex;
            break;

    default: hex=r + hex;
        break;
    }
    n=n/16;
    }
    System.out.println("Hexadecimal Equivalent :"+hex);
}

    }
```

OUTPUT:

```
enter the decimal number
469
Hexadecimal Equivalent :1D5
```

Write a program to convert DECIMAL to ALL(Octal , Hexa and Binary)?

```
import java.util.*;
public class DectoAll
{
public static void main(String[] args)
{
    System.out.println("enter the number");
    Scanner sc=new Scanner(System.in);
    int n=sc.nextInt();
    System.out.println("enter the base");
    int ba=sc.nextInt();
    System.out.println(ba +"base equivalent "+Convert(n, ba));

}
static String Convert(int num, int base)
{
    String st="0123456789ABCDEF";
    String b="";
    while(num>0)
    {
        int r= num % base;
        b=st.charAt(r)+b;
        num=num/base;

    }
    return b;
}
}
```

OUTPUT:

```
enter the number: 469
enter the base: 16
16 base equivalent: 1D5

enter the number: 369
enter the base: 8
8 base equivalent : 561

enter the number: 50
enter the base: 2
2 base equivalent: 110010
```

Write a program to convert DECIMAL to HEXADECIMAL?

```
import java.util.Scanner;
class HexatoDec
{
    public static void main(String[] args)
    {
        System.out.println("enter the Hexa dec number");
        Scanner sc=new Scanner(System.in);
        String st=sc.nextLine();
        int dec = 0;
        int count = 0;
        int l = st.length();
        while(l>0)
        {
            int r=0;
            char ch=st.charAt(l-1);
            if(ch>=65 && ch<=70)
                r=ch-55;
            else if(ch>=97 && ch<=102)
                r=ch-87;
            else
                r=ch-48;
            dec=dec + r*pow(16,count);
            count++;
            l--;
        }
        System.out.println("Decimal Equivalent: "+dec);
    }
    static int pow(int n ,int p)
    {
    int pw=1;
    while(p>0)
    {
        pw=pw*n;
        p--;
```

```
            }
            return pw;
            }
}
```

OUTPUT:

```
enter the Hexa dec number:  1D5
Decimal Equivalent: 469
```

PROGRAMS on
STAR PATTERNS

Write a program to display EQUILATERAL TRIANGLE with stars?

```
import java.util.Scanner;
public class EquiTri
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the number");
            int n = sc.nextInt();
        for(int i=0;i<n ;i++)
        {
            for (int j=0;j<n-i-1;j++)
            {
                System.out.print(" ");

            }
            for(int k=0;k<=i; k++)
            {
                System.out.print("* ");
            }
            System.out.println( );


        }
    }

}
```

OUTPUT:

enter the number: 7

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

Write a program to Display INVERTED TRIANGLE with stars?

```
import java.util.Scanner;
public class InverTri
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the number");
            int n = sc.nextInt();
        for(int i=0;i<n ;i++)
        {
            for (int j=0;j<i; j++)
            {
                System.out.print(" ");

            }
```

```
                        for(int k=0;k<2*(n-i)-1;k++)
                        {
                                System.out.print("*");
                        }
                        System.out.println ( );
                }
        }

}
```

OUTPUT:

```
enter the number: 4
*******
  *****
    ***
     *
```

Write a program to display the FILLED BOX with stars?

```
class FilledBox
{
        public static void main(String[] args)
        {
                java.util.Scanner sc=new java.util.Scanner(System.in);
                System.out.println("enter value of n");
                int n=sc.nextInt();
                for(int i=1;i<n ;i++)
                {
                        for (int j=0;j<n ;j++ )
                        {
                                System.out.print("*");
                        }
                        System.out.println();
                }
        }
}
```

Output:

```
enter value of n: 7
*******
*******
*******
*******
*******
*******
```

Write a program to display the HALLOW BOX with stars?

```
class Box1
{
        public static void main(String[] args)
        {
                java.util.Scanner sc = new java.util.Scanner(System.in);
                System.out.println ("enter value of n");
                int n = sc.nextInt();
                for (int i=0;i<n ;i++ )
                {
                        for (int j=0;j<n ;j++ )
                        {
                                if (i==0||j==0||i==n-1||j==n-1)
                                {
                                        System.out.print("*");
                                }
                                else
                                {
                                        System.out.print(" ");
                                }
                        }
                        System.out.println();
                }
        }
}
```

Output:

```
enter value of n 7
* ***** *
*         *
*         *
*         *
*         *
*         *
* ***** *
```

Write a program to display the BOX and CROSS inside it with stars?

```
class Box1
{
        public static void main(String[] args)
        {
                java.util.Scanner sc=new java.util.Scanner(System.in);
                System.out.println("enter value of n");
                int n=sc.nextInt();
                for (int i=0;i<n ;i++ )
                {
                        for (int j=0;j<n ;j++ )
                        {
                                if (i==0||j==0||i==n-1||j==n-1||i==j||i+j==n-1)
                                {
                                        System.out.print("*");
                                }
                                else
                                {
                                        System.out.print(" ");
                                }

                        }
                        System.out.println();
                }
        }
}
```

OUTPUT:

```
enter value of n: 7
*******
**        **
*  *    *  *
*     *     *
*  *    *  *
**       * *
*******
```

Write a program to display CROSS mark with stars?

```
class  Cross
{
	public static void main(String[] args)
	{
		java.util.Scanner sc=new java.util.Scanner(System.in);
		System.out.println("enter value of n");
		int n=sc.nextInt();
		for(int i=1;i<n ;i++)
		{
			for (int j=0;j<n ;j++ )
			{
				if(i==j||I + j==n-1)
					System.out.print("*");
				else
					System.out.print(" ");
			}
			System.out.println();
		}

	}
}
```

OUTPUT:
enter value of n 7(odd)

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

Write a program to display RIGHT ANGLE triangle with stars?

```
class Triangle
{
	public static void main(String[] args)
	{
		java.util.Scanner sc=new java.util.Scanner(System.in);
		System.out.println("enter value of n");
		int n=sc.nextInt();
		for(int i=1;i<n ;i++)
		{
			for (int j=0;j<i ;j++ )
```

```
                        {
                                System.out.print("*");
                        }
                        System.out.println();
                }
        }
}
```

OUTPUT:

```
enter value of n :7
*
**
***
****
*****
******
```

Write a program to display Reverse Triangle with stars?

```
class Triangle1
{
        public static void main (String [] args)
        {
                java.util.Scanner sc=new java.util.Scanner (System.in);
                System.out.println ("enter value of n");
                int n=sc.nextInt ();
                for (int i=1; i<n; i++)
                {
                        for (int j=0; j<n; j++)
                        {
                                if (i<=j)
                                        System.out.print ("*");
                                else
                                        System.out.print (" ");
                        }
                        System.out.println ();
                }
        }
}
```

OUTPUT:

```
enter value of n 7
 ******
  *****
   ****
    ***
     **
      *
```

Write a program to display MIRROR of RIGHT ANGLE triangle with stars?

```
class Triangle2
{
        public static void main(String[] args)
        {
                java.util.Scanner sc=new java.util.Scanner(System.in);
        System.out.println("enter value of n");
        int n=sc.nextInt();
        for(int i=1;i<n ;i++)
        {
            for (int j=0;j<n ;j++ )
            {
                if(i + j>n-1)
                    System.out.print("*");
                else
                    System.out.print(" ");
            }
            System.out.println();
        }
    }
}
```

OUTPUT:

```
enter value of n: 7
      *
     **
    ***
   ****
  *****
 ******
```

Write a program to display DOWNWARD MIRROR of RIGHT ANGLE triangle with stars?

```
class Triangle2
{
    public static void main(String[] args)
    {
        java.util.Scanner sc=new java.util.Scanner(System.in);
        System.out.println("enter value of n");
        int n=sc.nextInt();
        for(int i=1;i<n ;i++)
        {
            for (int j=0;j<n ;j++ )
            {
                if(i + j<=n-1)
                    System.out.print("*");
                else
                    System.out.print(" ");
            }
            System.out.println();
        }
    }
}
```

OUTPUT:

```
enter value of n: 7
******
*****
****
***
**
*
```

Write a program to display DIAMOND with stars?

```
class Diamond
{
public static void main(String[] args)
{
        java.util.Scanner scn=new  java.util.Scanner (System.in);
System.out.println ("enter  odd number");
        int n=scn.nextInt();
        int spaces=n/2;
        int stars=1;
for(int i=1;i<n ;i++)
{
for( int j=1;j<=spaces ;j++)
{
System.out.print(" ");
}
for ( int k=1;k<=stars ;k++)
{
System.out.print("*");
}
System.out.println();
if (i<=n/2)
{
        spaces--;
        stars+=2;
}
else
        {
        spaces++;
   stars-=2;
        }
}
}
}
```

OUTPUT:

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

Write a program to display HALLOWDIAMOND with stars?

```
import java.util.Scanner;
class HallowDiamond
{
        public static void main(String[] args)
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("enter   the value of n");
        int n = sc.nextInt();
        n = (n+1)/2;
        for (int i=0;i<n ;i++ )
        {
            for (int j=0;j<n-i-1 ;j++ )
            {
                System.out.print(" ");
            }
            for (int j=0;j<2*i+1 ;j++ )
            {
                if (j==0||j==2*i)
                {
                    System.out.print("*");
                }
                else
                    System.out.print(" ");
            }
            System.out.println();
        }
        n = n-1;
        for (int i=0;i<n ;i++ )
        {
            for (int j=0;j<=i ;j++ )
            {
                System.out.print(" ");
            }
            for (int j=0;j<2*(n-i)-1 ;j++ )
            {
                if (j==0||j==2*(n-i)-2)
                {
                    System.out.print("*");
                }
                else
                    System.out.print(" ");
            }
            System.out.println();
        }
    }
}
```

OUTPUT:

```
enter   the value of n  ; 13
      *
     * *
    *   *
   *     *
  *       *
 *         *
*           *
 *         *
  *       *
   *     *
    *   *
     * *
      *
```

Write a program to display NUMBERS in DIAMOND shape?

```
import java.util.Scanner;
class NumDiamond
{
    public static void main(String[] args)
    {
Scanner sc = new Scanner(System.in);
        System.out.println("enter   the value of n");
        int n = sc.nextInt();
        n = (n+1)/2;
        for (int i=0;i<n ;i++ )
        {
            for (int j=0;j<n-1-i ;j++ )
            {
                System.out.print(" ");
            }
            int k=1;
            for (int j=0;j<2*i+1 ;j++ )
            {
                System.out.print(""+k);
                if (j<(2*i+1)/2)
                k++;
                else
                k--;
            }
            System.out.println();
        }
        n = n-1;
        for (int i=0;i<n ;i++ )
        {
            for (int j=0;j<=i ;j++ )
            {
                System.out.print(" ");
            }
            int k=1;
            for (int j=0;j<2*(n-i)-1 ;j++ )
            {
                System.out.print(""+k);
                if (j<(2*(n-i)-1)/2)
                k++;
                else
                    k--;
            }

            System.out.println();
        }
    }
}
```

OUTPUT:

```
enter   the value of n: 7
   1
  121
 12321
```

```
1234321
 12321
  121
   1
```

Write a program to display CHARACTERS in DIAMOND shape?

```
import java.util.Scanner;
class CharDiamond
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter   the value of n");
        int n = sc.nextInt();
        n = (n+1)/2;
        char ch='A';
        for (int i=0;i<n ;i++ )
        {
            for (int j=0;j<n-1-i ;j++ )
            {
                System.out.print(" ");
            }
            int k=0;
            for (int j=0;j<2*i+1 ;j++ )
            {
                System.out.print(""+(char)(ch + k));
                if (j<(2*i+1)/2)
                k++;
                else
                    k--;
            }
            System.out.println();
        }
        n = n-1;
        for (int i=0;i<n ;i++ )
        {
            for (int j=0;j<=i ;j++ )
            {
                System.out.print(" ");
            }
            int k=0;
            for (int j=0;j<2*(n-i)-1 ;j++ )
            {
                System.out.print(""+(char)(ch + k));
                if (j<(2*(n-i)-1)/2)
                k++;
                else
                    k--;
            }
            System.out.println();
        }
    }
}
```

OUTPUT:

```
enter   the value of n: 7
     A
   ABA
  ABCBA
ABCDCBA
  ABCBA
    ABA
      A
```

Write a program to display M pattern with stars?

```
class DisplayM
{
        public static void main(String[] args)
        {
                int spaces=8;
                for (int i=1;i<=5 ;i++ )
                {
                        for ( int j=1;j<=i ;j++ )
                        {
                       System.out.print("*");
                        }
                        for (  int k=1;k<=spaces ; k++)
                        {

                                System.out.print(" ");
                        }
                        for(int l=1;l<=i ;l++)
                                {
                          System.out.print("*");
                        }

                System.out.println();
                spaces -=2;
                }

        }
        }
```

OUTPUT:

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

Write a program to display sequence of numbers in TRIANGLE format?

```
import java.util.Scanner;
class Series
{
        public static void main(String[] args)
```

```
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the rows");
        int n = sc.nextInt();
        int k =0;
        for ( int i=1;i<=n ;i++ )
        {
            for ( int j=1;j<=i ; j++)
            {
                k++;
                System.out.print(k+" ");

            }
        System.out.println(" ");
        }

    }
}
```

OUTPUT:

```
enter the rows: 5
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

Programs on Strings

Ashirbad Swain

Write a program to find weather a string is ANAGRAM or not?

Def: a word, phrase, or name formed by rearranging the letters of another, such as *silent* formed from *listen*.

```
class Anagram
{
    static String removeSpaces(String str)
    {
        char [] ch=str.toCharArray ();

        //convert the string into array

        String nstr=" ";

        //create a new empty string

        for(int i=0;i<ch.length;i++)
        {
            if(ch[i]!=' ')
                nstr=nstr + ch[i];

                /* if the character at ith index is not equal to space
                then add that character to new empty string*/
        }
        return nstr;
    }

    static String toLowerCase(String str)
    {
        char[] ch=str.toCharArray();

            //convert the string into array

        String nstr=" ";

            //create a new empty string

        for(int i=0;i<ch.length;i++)
        {
            if(ch[i]>=65 && ch[i]<=90)
            {
                nstr=nstr+((char)ch[i]+32);
            }
                    /*if any alphabet is in upper case convert it
                        into lower case*/
            else
            {
                nstr=nstr + ch[i];
                    //if it is in lower case no need to convert
            }
        }
        return nstr;
    }
    static String sort(String str)
    {
        char[] ch=str.toCharArray();

            //sort string in alphabetical order

        for(int i=0;i<ch.length-1;i++)
        {
            for(int j=i+1;j<ch.length;j++)
            {
                if(ch[i]>ch[j])
                {
                    char t=ch[i];
                    ch[i]=ch[j];
                    ch[j]=t;
                }
            }
        }
        String st=new String(ch);
        return st;
    }

    static boolean compare(String s1, String s2)
    {
        if(s1.length()!=s2.length())
            return false;
        else
        {
            s1=toLowerCase(s1);
            s2=toLowerCase(s2);
            s1=sort(s1);
            s2=sort(s2);
            char ch1[]=s1.toCharArray();
            char ch2[]=s2.toCharArray();

            for(int i=0;i<ch1.length;i++)
            {
                if (ch1[i]!=ch2[i])
                {
                    return false;
                }
            }
            return true;
        }
    }

    public static void main(String[] args)
    {
        java.util.Scanner sc=new java.util.Scanner(System.in);
        System.out.println ("Enter the first string");
        String s1=sc.nextLine();
        System.out.println ("Enter the second string");
        String s2=sc.nextLine();
        s1=removeSpaces (s1);
        s2=removeSpaces (s2);
        boolean b= compare(s1,s2);

        if(b)
            System.out.println("string is anagram");
        else
            System.out.println("not an anagram");

    }
}
```

Output:

```
    Enter the first string
    Mother in law
    Enter the second string
    Hitler woman
    string is anagram
```

Write program weather the string is PANAGRAM or not?

Def: a sentence containing every letter of the alphabet.

```
import java.util.Scanner;
public class Panagram
{
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("enter the string ");
        String s = sc.nextLine();
        System.out.println("given string is :"+"\n" +s);
        String st=removeSpace(s);
```

```
       int d = check(st);
        if(d == -1)
            System.out.print(s+"\n" + "is not pangram");
        else
             System.out.print(s+"\n" +"is a pangram");

    }
    public static String removeSpace(String s)
    {
            char ch[]=s.toCharArray();
            String nstr="";
            for (int i = 0; i < s.length(); i++)
            {
                                     if (ch[i]!=' ')
                                     {
                                                 nstr=nstr + ch[i];
                                     }
            }

   return nstr;
            }

            public  static int check(String st)
    {

             int n = 26;

        /*if(s.length() < n){
           return -1;                                   use these lines only for perfect  Panagram  i.e., it must
contain only
        }*/                                             26 letters (alphabets) without any repetition.

        for(char i = 'A'; i <= 'Z' ; i++){
            if((st.indexOf(i) < 0) && (st.indexOf((char)(i + 32)) < 0))
            {
               return -1;
            }
        }
        return 1;
    }
}
```

OUTPUT:

```
enter the string:
the quick brown fox jumps over a lazy dog
given  string is :
the quick brown fox jumps over a lazy dog
the quick brown fox jumps over a lazy dog
is a pangram
```

Write a program check the given string is PALINDROME or not?

```
import java.util.Scanner;
```

```
public class PalindromeStr
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the string");
        String st=sc.nextLine();
        String nstr="";
        char ch[]=st .toCharArray();
        for (int i=0 ;i<ch.length/2;i++ )
        {
            char t=ch[i];
            ch[i]=ch[ch.length-1-i];
            ch[ch.length-1-i]=t;
        }
        nstr=new String (ch);

        if(nstr.equalsIgnoreCase(st))
        System.out.println( st+"  string is palindrome ");
        else
        System.out.println(st+" string is not palindrome");

    }
}
```

OUTPUT:

```
Enter the string: Malayalam
Malayalam  string is palindrome
```

Write a program to display REVERSE of a STRING?

```
import java.util.Scanner;
class  Revstring
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the string");
        String st=sc.nextLine();
        char ch[]=st.toCharArray();
        for (int i=0 ;i<ch.length/2;i++ )
        {
         char t=ch[i];
        ch[i]=ch[ch.length-1-i];
        ch[ch.length-1-i]=t;
        }
  st=new String (ch);

 System.out.println("Reserved string is :"+st);
    }
}
```

OUTPUT:

```
enter the string
rama and laxmana
Reserved string is : anamxal dna amar
```

Write a program to COUNT number of CHARACTERS in a String?

```
import java.util.Scanner;
public class NoOfCharactersInaString
{

        public static void main(String[] args)
        {
                int count=0;
                Scanner scn=new Scanner(System.in);
                System.out.println("Enter a string:..........");
                String st=scn.nextLine();
                char ch[]=st.toCharArray();
                for (int i = 0; i < ch.length; i++)
                {
                        if(ch[i]>=65&&ch[i]<=90  ||ch[i]>=97 && ch[i]<=122||ch[i]>=48&&ch[i]<=57  &&
                                                                    ch[i]!=32  && ch[i]!=','  &&ch[i]!='.')
                         count++;
                }
                System.out.println("No of Characters="+count);
        }
}
```

OUTPUT:

```
Enter a string:..........
adkvdh  dodksk
No of Characters=12
```

Write a program to find the sum of numbers in an ALPHA NUMERIC STRING?

```
import java.util.Scanner;
public class SumOfDigits
{
        public static void main(String[] args)
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("enter the alpha numeric string");
        String str=sc.nextLine();
        char[] ch=str.toCharArray();
        int j=0;
        for(int i=0;i<ch.length;i++)
        {
                if(ch[i]>=48 && ch[i]<=57)
                {

                        j+=ch[i]-48;
                }
        }
                System.out.println(j);
        }
}
```

OUTPUT:

```
enter the alpha numeric string
139y1d5801
28
```

Write a Program for number of characters in each WORD and count them?

```
import java.util.Scanner;
class  Countword
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the string");
        String s=sc.nextLine();
        String nst=" ";
        int nc=0;
        for (int i=0; i<s.length();i++ )
        {
            if (s.charAt(i)==' ')
            {
                nst=nst + nc;
                nc=0;
            }
            else
            {
                nc++;
                nst=nst + s.charAt(i);
            }
        }
        nst=nst + nc;
        System.out.println (" no of character in each word in a string is "+  nst);
    }
}
```

OUTPUT:

```
enter the string
rama and laxmana
 no of character in each word in a string is rama 4 and 3 laxmana 7
```

Write a Program to display OCCURENCES of each character in a STRING?

```
import java.util.Scanner;
class NumOfOcc
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the String");
        String st = sc.nextLine();
        int n=st.length();
        char ch[]=st.toCharArray();
        for (int i=0;i<n ;i++ )
        {
            int count=1;
            for (int j=i+1;j<n ;j++ )
            {
                if(ch[i]==ch[j])
                {
                count++;
                int k=j;
                while (k<n-1)
                {
                    ch[k]=ch[k+1];
                    k++;
                }
                n--;
                j--;
                }
            }
             System.out.println(ch[i]+" occurred "+count+" times");
        }
        String nst=" ";
        for (int i=0;i<n ;i++ )
        {
            nst=nst + ch[i];

        }
         System.out.println(nst);

    }

}
```

OUTPUT:

```
Enter the String Malayalam
m occurred 2 times
a occurred 4 times
l occurred 2 times
y occurred 1 times
 maly
```

Write a program to display number of LOWERCASE, UPPERCASE, SPECIAL SYMBOLS, SPACES and DIGITS in a STRING?

```
import java.util.Scanner;
class  DiffTypeCharsSymbols
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the string");
        String st=sc.nextLine();
        char ch[]=st.toCharArray();
        int uc=0,lc=0,spc=0,dc=0,sp=0;
        for (int i=0;i<ch.length ;i++ )
        {
            if (ch[i]>=65&&ch[i]<=90)
```

```
                    { uc++;
                    }
                    else if (ch[i]>=97&&ch[i]<=122)
                    {
                        lc++;
                    }
                    else if (ch[i]>=48&&ch[i]<=57)
                        dc++;
                    else
                        if(ch[i]==' ')
                        sp++;
                    else spc++;

                }
                System.out.println("no :of upper case letter "+uc);
                System.out.println("no: of lower case letter" +lc);
                System.out.println("no: of decimal number" +dc);
                System.out.println("no: of spaces "+sp);
                System.out.println("no: of special characters" +spc);
        }
}
```

OUTPUT:

```
enter the string: PramoD123$@gmail.com
no :of upper case letter 2
no : of lower case letter12
no : of decimal number3
no : of spaces 0
no : of special characters3
```

Write a program to convert NUMBER into WORDS?

```
import java.util.*;
public class Numtoword
{
        static String one[]={"","one","two","three","four","five","six","seven","eight","nine","ten",

"eleven","tweleve","thirteen","fourteen","fifteeen","sixteeen","seventeen","eighteen","nineteen"};
        static String two[]={"","","twenty","thirty","fourty","fifty","sixty","seventy","eigty","ninety"};

        static void pw(int n, String st)
{
        if(n<=19)
                System.out.print(one[n]+" ");
        else
                System.out.print(two[n/10]+one[n%10]+" ");
        if(n!=0)
                System.out.print(st+" ");
}
public static void main(String[] args)
{
        System.out.println("enter the number");
        Scanner sc=new Scanner(System.in);
        int num=sc.nextInt();
        pw(num/10000000,"crores");
```

```
    pw((num/100000)%100,"Lakhs");
    pw((num/1000)%100,"Thousand");
    pw((num/100)%10,"Hundered");
    pw(num%100," ");
}
}
```

OUTPUT:

```
enter the number : 999999
 nine Lakhs ninety nine Thousand nine Hundered and  ninety nine
```

Write a program to REVERSE the SENTENCE?

```
import java.util.Scanner;
class  Revsentence
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the sentence");
        String st=sc.nextLine();
        char ch[]=st.toCharArray();
        String rst=" ";
        for (int i=ch.length-1;i>=0 ;i-- )
        {
            int k=i;
            while (i>=0&&ch [i]!=' ')
            {
                i--;
            }
            int j=i+1;
            while ( j<=k)
            {
                rst =rst +ch[j];
                j++;
            }
            rst=rst+' ';
        }
        System.out.println("The reserve sentence is:"+rst);
    }
}
```

OUTPUT:

```
enter the sentence: rama and laxmana
The reserve sentence is: laxmana and rama
```

Write a program to REVERSE THE WORDS in a SENTENCE?

```
import java.util.Scanner;

class  Revwords
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the sentence");
        String st=sc.nextLine();
        char ch[]=st.toCharArray();
        String rst=" ";
        for (int i=0 ;i<ch.length;i++ )
        {
            int k=i;
            while (i<ch.length &&ch [i]!=' ')
            {
                i++;
            }
            int j=i-1;
            while ( k<=j)
            {
                rst=rst + ch[j];
                j--;
            }
            rst=rst+' ';
        }
System.out.println("The reserved  words of sentence  is:"+rst);
    }
}
```

OUTPUT:

```
enter the sentence: rama and laxmana
The reserved  words of sentence  is: amar dna anamxal
```

Write a program to display STRING INITCAP of Words?

```
import java.util.Scanner;
class  Stringinitcap
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the string");
        String st=sc.nextLine();
        char ch[]=st.toCharArray();
        for (int i=0 ;i<ch.length;i++ )
        {
            if (i==0||(ch[i]!=' '&&ch[i-1]==' '))
            {
                if (ch[i]>=97&&ch[i]<=122)
                {
                    ch[i]=(char)(ch[i]-32);
                }
                else if (ch[i]>=65&&ch[i]<=90)
                {
                    ch[i]=(char)(ch[i]-32);

                }
            }
        }
            st=new String(ch);
System.out.println("enter the string in it cap : "+st);
    }
}
```

OUTPUT:

```
enter the string: pramod reddy pavan chandu
enter the string in it cap : Pramod Reddy Pavan Chandu
```

Write a program to convert UPPER CASE TO LOWER CASE & VICE VERSA?

```
import java.util.Scanner;

class  Stringuptolow
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the string");
        String st=sc.nextLine();
        char ch[]=st.toCharArray();
        for (int i=0 ;i<ch.length;i++ )
        {
                if (ch[i]>=65&&ch[i]<=90)
                {
                    ch[i]=(char)(ch[i]+32);
                }
                else if (ch[i]>=97&&ch[i]<=122)
                {
                    ch[i]=(char)(ch[i]-32);

                }

        }
            st=new String(ch);
System.out.println("converted String in Case : "+st);
    }
}
```

OUTPUT:

```
enter the string : PraMoD ReddY GoPi RedDY
converted String in Case : pRAmOd rEDDy gOpI rEDdy
```

Write a program to find a SUB-STRING without using INBUILT functions?

```
import java.util.Scanner;
class Substring
{
    public static void main(String[] args)
    {
        System.out.println("enter the main string");
        Scanner sc=new Scanner(System.in);
        String st1=sc.next();
        char ch1[]=st1.toCharArray();
        System.out.println("enter the sub string");
        String st2=sc.next();
        char ch2[]=st2.toCharArray();
        int find=0;
        for (int i=0;i<ch1.length ;i++ )
        {
            int k=i, j=0;
            while (k<ch1.length && j<ch2.length && ch1[k]==ch2[j])
            {
                j++;
                k++;

            }
            if(j==ch2.length)
            {
                find++;
                System.out.println( find+" times   "+st2+"  present between "+i+" to 
"+k+" indexs");

            }

        }
        if(find==0)
            System.out.println("not found");

    }
}
```

OUTPUT:

```
enter the main string : PramodReddy
enter the sub string : Reddy
1 times   Reddy  present between 6 to  11 indexs
```

Write a program to convert Integer of String type to INTEGER type without using parse int?

```
import java.util.Scanner;

public class StringToInt
{
public static void main (String [] args)
{
    Scanner sc=new Scanner (System.in);
    System.out.println ("enter the String");
    String s=sc.next ();
    System.out.println (" After converting string to integer");
    int d = check(s);
    if (d==0)
        System.out.println ("not valid string ");
    else
    System.out.println (d + “is in integer type");
}

public static int check (String s)
{
    int i=0, number=0;

    for (int j = 0; j < s.length (); j++)
    {
        char ch [] =s.toCharArray ();
        if (ch[j]>'a'&&ch[j] <='z'||ch[j]>'A'&&ch[j]<='Z')
        {
                return 0;
        }
    }

    while (i<s.length ())
    {
    number= number*10;
    number=number+ (s.charAt (i++)-'0');

    }
    return number;
}
}
```

OUTPUT:

```
enter the String
3306
 After converting string to integer
3306 is in integer type
```

SEARCHING & SORTING PROGRAMS

Ashirbad Swain

Write a program for LINEAR SEARCH?

```
public class SearchLinear
{
    public static int linearSearch(int[] arr, int x)
    {
        for(int i=0;i<arr.length;i++)
        {
            if(x==arr[i])
            {
                return i;
            }
        }
        return -1;
    }
    public static void main(String[] args)
    {
     int[] ar ={3,46,76,4,89,7,27};
     System.out.println(linearSearch(ar,4));
     System.out.println(linearSearch(ar,78));
    }

}
```

OUTPUT:

```
 3
-1
```

Write a program for BINARY SEARCH?

```
public class SearchBinary
{
    public static int binarySearch(int[] arr, int x)
    {
        int first=0;
        int last=arr.length-1;
        while(first<=l)
        {
            int middle=(first + last)/2;
            if(x==arr[middle])
            {
                return middle;
            }
            else if(x>arr[middle])
            {
                first=middle+1;
            }
            else
            {
                last=middle-1;
            }
```

```
                }return -1;
        }

        public static void main(String[] args)
        {
                int[] i={10,49,67,90,40,86};
                                System.out.println(binarySearch(i,49));
        }


}
```

OUTPUT:

```
1
```

Write a program for BUBBLE SORT?

```
class Bubbledown
{
public static void sortdown(int[]a)
        {
                int n=a.length;
                for (int i=0;i<n-1 ;i++ )
                {
                        for (int j=i+1;j<n ;j++ )
                        {
                                if(a[i]>a[j])
                                {
                                        int temp=a[i];
                                        a[i]=a[j];
                                        a[j]=temp;
                                }
                        }
                }
        }
        public static void main(String[] args)
        {
                int []a={5,8,1,6,9,2};
                sortdown(a);
                for (int x: a )
                {
                System.out.println(x);
                }
        }
}
```

OUTPUT:

```
1
2
5
6
8
```

PROGRAMS on ARRAYS

Write a program to INSERT the ELEMENTS in an Array?

```
import java.util.Scanner;
public class InstSingArray
{
public static void main (String [ ] args)
{
        Scanner sc= new Scanner (System.in);
        System.out.println ("enter the size");
        int length= sc.nextInt ();

        int arr [ ] =new int [length];
        System.out.println ("enter the   "+length+" elements");
        for (int i = 0; i < arr.length; i++)
        {
                arr[i] =sc.nextInt ();
        }
        for (int i = 0; i < arr.length; i++)
        {
                System.out.println ("arr ["+i+"] ---->"+arr[i]);
        }
}

}
```

Output: enter the size

```
5
Enter the   5 elements
2
3
5
8
64
arr [0] ---->2
arr [1] ---->3
arr [2] ---->5
arr [3] ---->8
arr [4] ---->64
```

Write a Program to REVERSE THE ELEMENTS of an array?

```
import java.util.Scanner;
public class InstSingArray
{
public static void main (String [ ] args)
{
        Scanner sc= new Scanner (System.in);
        System.out.println ("enter the size");
        int length= sc.nextInt ();
        int arr [ ] =new int [length];
        System.out.println ("enter the   "+length+" elements");
        for (int i = 0; i < arr.length; i++)
        {
                arr[i] =sc.nextInt ();
        }
System.out.println (“Before Reverse of an Array”);
for (int i = 0; i < arr.length; i++)
        {

                System.out.println ("arr ["+i+"] ---->"+arr[i]);
        }
for (int i = 0; i < arr.length/2; i++)
{
int t=arr[i];
arr[i] =arr [arr.length-1-i];
arr [arr.length-1-i) =t;
}
System.out.println (“After Reverse of an Array”);
for (int i = 0; i < arr.length; i++)
        {

                System.out.println ("arr ["+i+"] ---->"+arr[i]);
        }
}
}
```

Output:

```
Enter the size
5
Enter the   5 elements
1
5
6
8
9
Before Reverse of an Array
arr [0] ---->1
arr [1] ---->5
arr [2] ---->6
arr [3] ---->8
arr [4] ---->9

After Reverse of an Array
arr [0] --→9
arr [1] --→8
arr [2] --→6
arr [3] --→5
arr [4] --→1
```

Write a program to INSERT A ELEMENT INTO EXISTING ARRAY in a specified position?

```
import java.util.Scanner;

class Insertingelement
{
    public static void main (String [] args)
    {
        Scanner sc= new Scanner (System.in);
        System.out.println ("enter the length");
        int length= sc.nextInt ();

        int arr [] =new int [length];
        System.out.println ("enter the   "+length+" elements");
        for (int i = 0; i < arr.length; i++)
        {
            arr[i]=sc.nextInt();
        }
        System.out.println ("length of array before inserting"+"--->"+arr.length);
        for (int i=0; i<arr.length; i++)
        {
       System.out.println (i+"------->"+arr[i]);
         }
   System.out.println ("enter the index of specified position or index");
        int in=sc.nextInt ();
        System.out.println ("enter the element to replace to specified index");
        int ele=sc.nextInt ();
         arr=insert (arr ,in ,ele);

         for (int i=0; i<arr.length; i++)
            {
           System.out.println (i+"------->"+arr[i]);
              }

    }
    static int [] insert (int a[],int in, int ele)
    {
        if (in>a.length||in<0)
        {
            System.out.println ("invalid index");
        return a;
        }
        else
        {
            int na [] = new int [a.length+1];
            for (int i= 0 ; i<in  ;i++ )
            {
                na[i] = a[i];
            }
            na [in] =ele;
            for (int i= in; i<a.length; i++)
            {
                na [i+1] = a[i];
            }
System.out.println ("length of array after inserting"+"--->"+na.length);
            return na;
        }
    }
}
```

Output:

```
enter the length
5
enter the   5 elements
2
8
6
7
88


length of array before inserting--->5
0------->2
1------->8
2------->6
3------->7
4------->88
enter the index of specified position or index
3
enter the element to replace to specified index
62
length of array after inserting--->6
0------->2
1------->8
2------->6
3------->62
4------->7
5------->88
```

Write a program to DELETE AN ELEMENT OF A SPECIFIED INDEX IN THE EXISTING ARRAY?

```
import java.util.Scanner;

class DeletingArray

{
    public static void main (String [] args)
    {
        Scanner sc= new Scanner (System.in);
        System.out.println ("enter the length");
```

```
                int length= sc.nextInt ();

                int ar [] = new int [length];
                System.out.println ("enter the   "+length+" elements");
                for (int i = 0; i < ar.length; i++)
                {
                        ar[i] = sc.nextInt ();
                }
                System.out.println ("length of array before deleting"+"--->"+ar.length);
      display (ar);
      System.out.println ("enter specified position for deleting that element");
                 int in=sc.nextInt ();
      ar=delete (ar , in);
                display (ar);

        }
        static void display (int a[])
          {
                for (int i=0; i<a.length; i++)
       System.out.println (i+"------->"+a[i]);
          }

        static int [] delete (int a[] , int in)
        {
                If (in>a.length||in<0)
                {
                        System.out.println ("invalid index");
                return a;
                }
                else
                {
                        int na [] = new int [a.length-1];
                        for (int i=0; i<in; i++)
                        {
                                na[i] = a[i];
                        }
                        for (int i=in; i<a.length; i++)
                        {
                                na [i-1] = a[i];
                        }
                        System.out.println ("length of array after deleting"+"---->"+na.length);
                        return na;
                }
         }
}
```

OUTPUT:

```
enter the length
6
enter the   6 elements
5
5
```

```
9
8
6
2
length of array before deleting--->6
0------->5
1------->5
2------->9
3------->8
4------->6
5------->2
enter specified position for deleting that element
5
length of array after deleting---->5
0------->5
1------->5
2------->9
3------->8
4------->2
```

Write a program to SEARCH AN ELEMENT IN THE EXISTING ARRAY?

```
public class Search element
{
    public static void main (String [] args)
    {
        int ar [] = {22, 11, 23, 11, 15, 19};
        int inx=search (ar, 15);
        display (ar);
        if (inx>=0)
            System.out.println ("your element found at index   "+inx);
        else
        System.out.println ("not valid");
    }
    static void display (int a [])
     {
        for (int i=0; i<a.length; i++)
   System.out.println (i+"------->"+a[i]);
     }
    static int search (int a [], int ele)
    {
        for (int i=0; i<a.length; i++)
        {
            If (ele==a[i])
                return i;
        }
        return -1;
    }
}
```

OUTPUT:

```
0------->22
```

```
1------->11
2------->23
3------->11
4------->15
5------->19
your element found at index   4
```

Write a program to find BIGGEST AND SMALLEST ELEMENT in the given array?

```
import java.util.Scanner;

public class BigeleArray
{
        public static void main (String [] args)
        {
                Scanner sc= new Scanner (System.in);
                System.out.println ("enter the length");
                int length= sc.nextInt ();
                int arr [] =new int [length];
        int bigger=0;
         int smaller = 0;
                System.out.println ("enter the   "+length+ “elements");
                for (int i = 0; i < arr.length; i++)
                {
                        arr[i] = sc.nextInt ();
                }
                for (int i = 0; i < arr.length; i++)
                {
                        System.out.println ("arr ["+i+"] ----> "+arr[i]);
                }
                for (int i = 0; i < arr.length; i++)
                {
                        int big=arr [0];
                        int small=arr [0];
                        if (big<arr[i])
                        {
                                big=arr[i];
                        }
                        if (small>arr[i])
                        {
                                small=arr[i];
                        }
                smaller=small;
        bigger=big;

                }
                System.out.println ("biggest element is ---->"+bigger);
                System.out.println ("Smallest element is ---->"+smaller);
   }

}
```

OUTPUT:

```
enter the length
5
enter the   5elements
1
8
99
66
75
arr [0] ---->1
arr [1] ---->8
arr [2] ---->99
arr [3] ---->66
arr [4] ---->75
biggest element is ---->75
Smallest element is ---->1
```

Write a program to find FIRST BIGGEST AND SECOND BIGGEST ELEMENT in given array?

```
class Fbiggest
{
        public static void main (String [] args)
        {
                int ar[]={23,12,14,56,22,28,13};
                int fbig=ar [0];
                int sbig=ar [1];
                for (int i=1; i<ar.length; i++)
                {
                        if (fbig<ar[i])
                        {
                                sbig=fbig;
                                fbig=ar[i];
                        }
                        else if (sbig<ar[i])
                        {
                                sbig=ar[i];
                        }
                }
                System.out.println ("first biggest element is “+fbig);
                System.out.println ("second biggest element is “+sbig);
        }
}
```

OUTPUT:

```
First biggest element    is  56
Second biggest element is 28
```

Write a program to FIND THE SECOND OCCURRENCE ELEMENT in a given array?

```
class Secondoccuranceelement
{
        public static void main (String [] args)
        {
                int ar[]={22,11,23,11,15,19,11};
                int inx=secondoccurance (ar, 11);
                display (ar);
                if (inx>=0)
                        System.out.println ("Second time occurred element found at the  index
"+inx);
                else
                System.out.println ("not valid");
        }
        static void display (int a [])
          {
                for (int i=0; i<a.length; i++)
      System.out.println ("arr ["+i+"]"+"------->"+a[i]);
          }
        static int secondoccurance (int a [], int ele)
        {
                int count=0;
                for (int i=0; i<a.length; i++)
                {
                        If (ele==a[i])
                                count++;
                        if (count==2)
                                return i;
                }
                return -1;
        }
}
```

OUTPUT:

```
arr [0] ------->22
arr [1] ------->11
arr [2] ------->23
arr [3] ------->11
arr [4] ------->15
arr [5] ------->19
arr [6] ------->11

Second time occurred element found at the index 3
```

Write a program to FIND THE OCCURRENCE ELEMENT IN which position in a given array?

```
class Occuranceelement
{
        public static void main (String [] args)
        {
                int ar[]={22,11,23,11,15,19,11};
                int inx=occurrence (ar, 11, 2);
                display (ar);
                if (inx>=0)
                    System.out.println ("your element found at index   "+inx);
                else
                System.out.println ("not valid");
        }
        static void display (int a [])
         {
                for (int i=0; i<a.length; i++)
           System.out.println (i+"------->"+a[i]);
         }
        static int occurrence (int a [], int ele, int oc)
        {
                int count=0;
                for (int i=0; i<a.length; i++)
                {
                    if (ele==a[i])
                        count++;
                    if (count==oc)
                        return i;
                }
                return -1;
        }
}
```

OUTPUT:

```
0------->22
1------->11
2------->23
3------->11
4------->15
5------->19
6------->11
Your element found at index   3
```

Write a program to FIND HOW MANY TIMES ELEMENT IS OCCURED in a given array?

```
class Elementoccured
{
    public static void main (String [] args)
    {
        int ar[]={22,11,23,11,15,19,11};
        int in=occurred (ar, 11);
        display (ar);
        if (in>=0)
            System.out.println ("your element occurred “+in);
        else
        System.out.println ("not valid");
    }
    static void display (int a [])
     {
        for (int i=0; i<a.length; i++)
```

```
      System.out.println (i+"------->"+a[i]);
          }
        static int occurred (int a [], int ele)
        {
                int count=0;
                for (int i=0; i<a.length; i++)
                {
                        if (ele==a[i])
                                count++;
                }
                return count;
        }
}
```

OUTPUT:

```
0------->22
1------->11
2------->23
3------->11
4------->15
5------->19
6------->11
Your element occurred 3
```

Write a program to DISPLAY MISSING ELEMENT in a given sorted array?

```
class Missingelement
{
        public static void main (String [] args)
        {
                int ar [] = {8, 15, 21, 24, 30, 37};
                System.out.println ("Missing elements in given array are :");
                for (int i=0;i<ar.length-1 ;i++ )
                {
                        for (int j=ar[i]+1;j<ar[i+1]; j++ )
                        {

                                System.out.println (j);
                        }
                }
        }
}
```

OUTPUT:

```
Missing elements in given array are:
9 ,10 ,11 ,12 ,13 ,14 ,16 ,17 ,18 ,19 ,20 ,22 ,23 ,25 ,26 ,27 ,28 ,29 ,31 ,32 ,33 ,34 ,35 ,36
```

Write a program to FIND HIGHEST CONTIGUOUS SUM OF TWO ELEMENT in a given array?

```
public class Sumoftwoelemnts
{
```

```
                public static void main (String [] args)
                {
                        int ar[]={21,12,15,32,16,17,22};
                        int inx=0;
                        int big=ar [0] +ar [1];
                        for (int i=1; i<ar.length-1; i++)
                        {
                                if (big<ar[i] + ar [i+1])
                                {
                                        big=ar[i] +ar [i+1];
                                        inx=i;
                                }
                        }
                        System.out.println (“sum of two element"+"----->"+big);
                        System.out.println (“the first element"+"--->"+ar [inx]);
                        System.out.println (" the second element"+"--->"+ar [inx+1]);
                }
        }
```

OUTPUT:

```
Sum of two element----->48
The first element--->32
The second element--->16
```

Write a program to DISPLAY THE COMMON ELEMENTS between two arrays?

```
public class Commonelement
{
        public static void main (String [] args)
                {
                        int ar1 [] = {12, 13, 23, 15, 11, 16};
                        int ar2 [] = {53, 26, 23, 15, 18, 13};
                        System.out.println ("common elements are: ");
                        for (int i=0; i<ar1.length; i++)
                        {
                             for (int j=0; j<ar2.length;j++ )
                             {
                                if (ar1 [i] ==ar2 [j])
                                {
                                      System.out.println (ar1 [i]);
                                  break;
                                }
                              }
                        }
                }
        }
```

OUTPUT:

```
common elements are:
```

```
13
23
15
```

Write a program to EXCHANGE OF FIRST PART ELEMENT TO SECOND PART Element between two arrays?

```
public class Exchangeofelements
{
        public static void main (String [] args)
        {
            int ar[]={21,12,15,32,16,17,22};
            System.out.println ("BEFORE EXCHANGE OF ARRAY");
            for (int i = 0; i < ar.length; i++)
            {
                System.out.println (ar[i]);
            }
            int n;
            if (ar.length%2==0)
                n=ar.length/2;
            else
                n= (ar.length/2) +1;
            for (int i=0; i<ar.length/2; i++)
            {
                int t=ar[i];
                ar[i] =ar [n+i];
                ar [n+i]=t;

            }
            System.out.println ("AFTER EXCHANGE OF ARRAY");
            for (int i = 0; i < ar.length; i++)
            {
                System.out.println (ar[i]);
            }
        }
    }
```

OUTPUT:

```
BEFORE EXCHANGE OF ARRAY
21
12
15
32
16
17
22
AFTER EXCHANGE OF ARRAY
16
17
22
32
21
12
```

15

Write program TO DISPLAY DISTINCT ELEMENTS from given two array?

```
public class Disctinctelements
{
    public static void main (String [] args)
    {
        int ar1 [] = {12, 13, 23, 15, 11, 16};
        int ar2 [] = {53, 26, 23, 15, 18, 13};
        System.out.println ("Distinct elements from given two arrays");
        for (int i=0; i<ar1.length; i++)
        {
            int find=0;
        for (int j=0; j<ar2.length; j++)
        {
            if (ar1 [i] ==ar2 [j])
            {
                find=1;
              break;
            }

        }
        if (find==0)
                System.out.println (ar1 [i]);
        }

        for (int i=0; i<ar2.length; i++)
        {int find=0;
        for (int j=0; j<ar1.length; j++)
        {
            if (ar2 [i] ==ar1 [j])
            {
                find=1;
              break;
            }
        }
        if (find==0)
                System.out.println (ar2 [i]);
        }
    }
}
```

OUTPUT:

```
Distinct elements from given two arrays
12
11
16
53
26
18
13
```

```
13
```

Write a program to MERGE TWO ARRAYS?

```
public class Merge
{
public static void main (String [] args)
{
        int ar1 [] = {12, 13, 23, 15, 11, 16};
        int ar2 [] = {53, 26, 23, 15, 18, 13};
        int res [] =new int [ar1.length+ar2.length];
        int j=0;
        for (int i = 0; i < ar1.length; i++, j++)
        {
                res[j] =ar1 [i];
        }
        for (int i = 0; i < ar2.length; i++, j++)
        {
                res[j] =ar2 [i];
        }
        System.out.println ("MERGED ARRAY ");
        for (int i = 0; i < res.length; i++)
        {
                System.out.println (res[i]);
        }
}
}
```

OUTPUT:

```
MERGED ARRAY
12
13
23
15
11
16
53
26
23
15
18
13
```

Write a program to COMBINE TWO ARRAYS IN ZIGZAG manner?

```
public class Zigzag
{
        public static void main (String [] args)
        {
                int ar1 [] = {12, 13, 23, 15, 11, 16};
                int ar2[]={53,26,23,15,18,13,23,45};
                int res [] =new int [ar1.length+ar2.length];
```

```
            int i=0, j=0;

            for (int k = 0; k < res.length; )
            {
            if (i<ar1.length)
             {
                  res[k] =ar1 [i];
                  i++;
                  k++;
             }
            if (j<ar2.length)
             {
                  res[k] =ar2 [j];
                  j++;
                  k++;
              }
        }
            System.out.println ("ZIGZAG ARRAY IS");
            for (int l = 0; l < res.length; l++)
            {
                  System.out.println (res[l]);
            }
      }
}
```

OUTPUT:

```
ZIGZAG ARRAY IS
12
53
13
26
23
23
15
15
11
18
16
13
23
45
```

Write a program to find the PALINDROME numbers in the given ARRAY?

```
class Main3
{
      static void display (int a [])
      {
            for (int i=0; i<a.length; i++)
            {
                  System.out.print (a[i] +",");
            }
            System.out.println ();
      }
       static int revdig (int n)
            {
             int rev=0;
             while (n>0)
                   {
                    int r=n%10;
                    rev=rev*10+r;
                     n=n/10;
                   }
                   return rev;
            }
      public static void main (String [] args)
      {
       int ar [] = {232, 12, 78, 898, 34543, 45};

       display (ar);
      int count=0;
       for (int i=0; i<arr.length;i++ )
       {
             if (ar [i] ==revdig (ar[i]))                        count++;
       }
       System.out.println ("---------------------");
       System.out.println (" number of palindrome:"+count);
      }
}
```

OUTPUT:

```
232, 12,78,898,34543,45,
---------------------
 number of palindrome: 3
```

Write a program to read elements into the MATRIX from SCANNER?

```
import java.util.*;
class Main2
{
      static int [] [] readMat ()
      {
            Scanner sc= new Scanner (System.in);
            System.out.println ("Enter the Order");
   int m=sc.nextInt ();
        int n=sc.nextInt ();
        int ar [][] =new int[m][n];
        System.out.println ("enter "+m*n+" Elements");
        for (int i=0; i<ar.length; i++)
        {
              for (int j=0; j<ar[i].length; j++)
              {
                    ar[i] [j] =sc.nextInt ();
              }
        }
        return ar;
      }
      static void display (int a [] [])
      {
   for (int i=0; i<a.length; i++)
        {
         for (int j=0; j<a[i].length; j++)
         {
             System.out.print (a[i][j]+" ");
         }
         System.out.println ();
     }
    }
    public static void main (String [] args)
    {
        int ar [] []=readMat();
        System.out.println ("Entered Matrix :");
        display (ar);
    }
}
```

OUTPUT:

```
Enter the Order
2
2
enter 4 Elements
9
6
5
1
Entered Matrix:
9 6
5 1
```

Write a program to read inputs from SCANNER and find the BIGGEST ELEMENT in EACH ROW and EACH COLUMN?

```
import java .util.*;
class Readmatrix
{
    public static void main (String [] args)
    {
        Scanner sc=new Scanner (System.in);

        System.out.println ("enter the order");
    int m=sc.nextInt ();
    int n=sc.nextInt ();
    int ar[][]=new int [m][n];
    System.out.println ("enter" + m*n + " elements");
    for (int i=0;i<ar.length ;i++ )
    {
        for (int j=0;j<ar[i].length ;j++ )
        {
            ar[i][j]=sc.nextInt();
        }
    }
    System.out.println (" entered matrix:");
 for (int i=0;i<ar.length ;i++ )
    {
        for (int j=0; j<ar[i].length ; j++ )
        {
            System.out.print (ar[i][j]+"("+i+","+j+")");
        }
        System.out.println ();
    }

System.out.println ();

    for (int i=0;i<ar.length ;i++ )
    {
        int big=ar[i][0];
        for (int j=i ; j<ar[i].length ;j++)
        {
            if(big<ar[i][j])
             big = ar[i][j];
            break;
        }
        System.out.println (i+1+"row biggest element "+big);
    }
for (int i=0; i<ar[0].length ;i++ )
    {
      int big=ar[0][i];
        for (int j=0;j<ar.length ;j++ )
        {
            if (big<ar[j][i])
             big =ar[j][i];
        }
        System.out.println(i+1+"column biggest element "+big);
    }
    }
}
```

OUTPUT:

```
enter the order
2
2
enter4elements
5
6
8
9
entered matrix:
5(0,0)6(0,1)
8(1,0)9(1,1)

1row biggest element5
2row biggest element9
1column biggest element8
2column biggest element9
```

Write a program to read inputs from SCANNER and find the SUM of ELEMENTS in EACH ROW and EACH COLUMN?

```
import java.util.*;
class Rowwiseandcolwisesum
{
        static int [][] readMat()
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("enter the order");
           int m=sc.nextInt();
           int n=sc.nextInt();
           int ar[][]=new int [m][n];
        System.out.println("enter"+ m*n+ "elements");

        for (int i=0;i<ar.length ;i++ )
           {
                for (int j=0;j<ar[i].length ;j++ )
                {
                        ar[i][j]=sc.nextInt();
                }
           }
        return ar;
         }

        static void display(int a[][])
        {
                for (int i=0;i<a.length ;i++ )
           {
                for (int j=0;j<a[i].length ;j++ )
                {
                        System.out.print(a[i][j]+"("+i+","+j+")");
                }
                System.out.println();
        }
        }

        public static void main(String[] args)
        {
                int ar[][]=readMat();
                System.out.println("entered matrix");
                display(ar);
                for (int i=0;i<ar.length ;i++)
                   {
                        int rsum=0;
                        int csum=0;
                        for (int j=0;j<ar.length ;j++)
                        {
                                rsum=rsum + ar[i][j];
                                csum=csum + ar[j][i];
                        }

       System.out.println(i+1+"row sum is :"+rsum);
            System.out.println(i+1+"column sum is:"+csum);
                }
        }
}
```

OUTPUT:

```
enter the order
2
2
enter4elements
6
5
7
9
entered matrix
6(0,0)5(0,1)
7(1,0)9(1,1)
1row sum is :11
1column sum is:13
2row sum is :16
2column sum is: 14
```

SPECIAL PROGRAMS

Write a program to find the given YEAR is LEAP-YEAR or not?

```
import java.util.*;
public class Leapyear
{
        public static void main (String [] args)
        {
                Scanner sc=new Scanner (System.in);
                System.out.println ("Enter the year") ;
                int m=sc.nextInt ();
                if (m%4==0&&m%100! =0||m%400==0)
                        System.out.println ("it is a leap year");
                else
                        System.out.println ("not a leap year");
        }
}
```

OUTPUT:

```
Enter the year
1990
not a leap year

Enter the year
2016
it is a leap year
```

Write a program to find days between DATE to DATE?

```
import java.util.Scanner;
class Date
{
        final int m[]={31,28,31,30,31,30,31,31,30,31,30,31};
        int dd, mm, yyyy;
        Date (int dd, int mm, int yyyy)
        {
                this.dd=dd;
                this.mm=mm;
```

```
            this.yyyy=yyyy;
      }
      int getNumberOfLeapYear ()
       {
            if (mm>2)
                  return yyyy/4-yyyy/100+yyyy/400;
            else
                  return (yyyy-1)/4-(yyyy-1)/100+ (yyyy-1)/400;
       }
      int getNumberOfDays ()
      {
   int dCount= yyyy*365+getNumberOfLeapYear () +dd;
        for (int i=0; i<mm-1; i++)
        {
            dCount+=m[i];
        }
        return dCount;
      }
      int difference (Date d1, Date d2)
      {
       int dy1=d1. getNumberOfDays ();
       int dy2=d2. getNumberOfDays ();
       if (dy1>dy2)
             return dy1-dy2;
       else
             return dy2-dy1;
  }
  public String toString ()
      {
            return dd+":"+mm+":"+yyyy+" ";
      }
      static Date readDate ()
      {
            Scanner sc= new Scanner (System.in);
            System.out.println ("Enter dd: ");
            int dd=sc.nextInt ();
            System.out.println ("Enter mm: ");
            int mm=sc.nextInt ();
            System.out.println ("Enter yyyy: ");
            int yy=sc.nextInt ();
            return new Date (dd, mm, yyyy);
      }
      public static void main (String [] args)
      {
        Date date1=readDate ();
        Date date2=readDate ();
        System.out.println ("Number of Days between"+date1+
              "And"+date2+" is: "+date1.difference (date1, date2));
      }
}
```

OUTPUT:

```
Enter dd: 31
Enter mm: 08
Enter yyyy: 2016
Enter dd: 5
Enter mm: 09
Enter yyyy: 2016
Number of Days between31:8:2016 And5:9:2016 is: 5
```