

Computer Assignment Questions

1. Write a program to input a sentence and find the number of palindromes present in it, also count the number of palindromes present in it.

```
import java.util.Scanner;
class Palindrome_Words
{
    public static void main(String Args[])
    {
        Scanner input=new Scanner(System.in);
        System.out.println("Enter a sentence.");
        String sentence=input.nextLine().toLowerCase();
        String words[] = sentence.split(" ");
        int count = 0;
        for (String word : words)
        {
            boolean isPalindrome = true;
            for (int i = 0; i < word.length() / 2; i++)
            {
                if (word.charAt(i) != word.charAt(word.length() - i - 1))
                {
                    isPalindrome = false;
                    break;
                }
            }
            if (isPalindrome)
            {
                System.out.println(word);
                count++;
            }
        }
        System.out.println(count + " words in the given sentence are palindrome words.");
    }
}
```

2. Write a program to input the sentence and reprint it after exchanging the first and last words of the sentence.

```
import java.util.*;
class First_Last_Word
{
    public static void main(String Args[])
    {
        Scanner input=new Scanner(System.in);
        System.out.println("Enter a sentence.");
        String sentence=input.nextLine();
        String words[]=sentence.split(" ",0);
        int array_length=words.length;
        String temporary;
        temporary=words[0];
        words[0]=words[array_length-1];
        words[array_length-1]=temporary;
        for (String word:words)
        {
            System.out.print(word+" ");
        }
    }
}
```

3. Write a program to input a sentence and find the number of words which begin and end with a vowel; also print those words.

```

import java.util.Scanner;
class Vowel_Start_End
{
    public static void main(String Args[])
    {
        Scanner input=new Scanner(System.in);
        System.out.println("Enter a sentence.");
        String sentence=input.nextLine();
        String words[]=sentence.split(" ",0);
        String vowels = "aeiouAEIOU";
        int length = 0, word_count=0;
        for (String word : words)
        {
            length = word.length();
            if (vowels.indexOf(word.charAt(0)) != -1 && vowels.indexOf(word.charAt(length - 1))
            {
                System.out.println(word);
            }
        }
        System.out.println("There are "+word_count+" word(s) present in the sentence which start
    }
}

```

4. Write a program to input a sentence and print those words which do not contain a vowel, also count those words.

```

import java.util.Scanner;
class Vowel_Count
{
    public static void main(String Args[])
    {
        Scanner input=new Scanner(System.in);
        System.out.println("Enter a sentence.");
        String sentence=input.nextLine();
        String words[]=sentence.split(" ",0);
        String vowels = "aeiouAEIOU";
        int word_count=0;
        for (String word : words)
        {
            boolean has_Vowel = false;
            for(int i=0;i<word.length();i++)
            {
                if(vowels.indexOf(word.charAt(i)) != -1)
                {
                    has_Vowel=true;
                    break;
                }
            }
            if(!has_Vowel)
            {
                System.out.println(word);
                word_count++;
            }
        }
        System.out.println("There are "+word_count+" word(s) present in the sentence which do not contain vowels");
    }
}

```

5. Write a program to input a sentence, develop a menu based upon which print

(i) Word formed by combining first letters of each word.

(ii) Word formed by combining the last letters of each word.

```

import java.util.*;
class First_Last_Letter
{
    public static void main(String Args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter a sentence.");
        String sentence = input.nextLine();
        String words[] = sentence.split(" ");
        String new_word = "";
        System.out.println(
            "Please choose from the following: \n(i)Word formed by combining first letters of e
int choice = input.nextInt();
switch (choice)
{
    case 1:
        for (String word : words)
        {
            new_word += word.charAt(0);
        }
        System.out.println(
            "The new word formed after taking only the first letters of the words preser
        break;
    case 2:
        for (String word : words)
        {
            new_word += word.charAt(word.length() - 1);
        }
        System.out.println(
            "The new word formed after taking only the last letters of the words present
            + new_word);
        break;
    default:
        System.out.println("Please enter either '1' or '2' to continue.");
}
}
}

```

**6. Write a program to input a name and check whether it is a unique name or not.
(A name is said to be unique in which the letters do not repeat)**

```

import java.util.*;
class Unique_Name
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a string : ");
        String name = in.nextLine().toUpperCase();
        boolean isUnique = true;
        for (int i = 0; i < name.length(); i++)
        {
            char ch = name.charAt(i);
            for (int j = i + 1; j < name.length(); j++)
            {
                if (ch == name.charAt(j))
                {
                    isUnique = false;
                    break;
                }
            }
            if (!isUnique)
            {
                break;
            }
        }
        if (isUnique)
        {
            System.out.println(name+" is a Unique Name.");
        }
        else
        {
            System.out.println(name+" is not a Unique Name.");
        }
    }
}

```

7. Write a program which will input a number and check whether it is a special number or not. (A number is said to be special if the sum of the factorials of the digits is equal to the original number. Example:145)

```

import java.util.*;
class Special_Number
{
    public static void main(String Args[])
    {
        try (Scanner input = new Scanner(System.in));)
        {
            System.out.println("Enter a number.");
            int input_number = input.nextInt();
            int store_input_number = input_number;
            int factorial = 1, check_number = 0, temporary = 0;
            int number_digits = (String.valueOf(input_number).length());
            for (int i = 0; i <= number_digits; i++)
            {
                temporary = (store_input_number % 10);
                for (int j = 1; j <= temporary; j++)
                {
                    factorial *= j;
                }
                check_number += factorial;
                store_input_number = (store_input_number / 10);
            }
            if (check_number == input_number)
            {
                System.out.println("The number given " + input_number + " is a Special number.");
            }
            else
            {
                System.out.println("The number given " + input_number + " is not a Special number.");
            }
        }
    }
}

```

8. Using the switch statement, write a menu driven program

- (i) To input a sentence and count the number of vowels present in it.
- (ii) To input a word and check if it is palindrome or not.

```

import java.util.*;
class Vowels_Palindrome
{
    public static void main(String Args[])
    {
        Scanner input=new Scanner(System.in);
        System.out.println("What do you want to do? \n(i) To find the number of vowels present :
        int choice=input.nextInt();
        switch(choice)
        {
            case 1:
                System.out.print("Enter a sentence : ");
                String sentence=input.nextLine();
                String words[]=sentence.split(" ",0);
                String vowels="aeiouAEIOU";
                int vowel_count=0;
                for(String word:words)
                {
                    for(int i=0;i<word.length();i++)
                    {
                        if(vowels.indexOf(word.charAt(i))!=-1)
                        {
                            vowel_count++;
                        }
                    }
                }
                System.out.println(vowel_count+" vowels are present in the given sentence.");
                break;
            case 2:
                System.out.print("Enter a word : ");
                String word=input.next();
                String new_word="";
                for (int i = word.length()-1; i >= 0; i--)
                {
                    new_word+=word.charAt(i);
                }
                if (new_word.equalsIgnoreCase(word))
                {
                    System.out.println(word+" is a palindrome word.");
                }
                else
                {
                    System.out.println(word+" is not a palindrome word.");
                }
            }
        }
    }
}

```



```

        }
        break;
    default:
        System.out.println("Please enter a valid choice.");
    }
}
}

```

9. Write a program to input 10 numbers in 2 different arrays and find the sum of the 2 arrays and store it in the third array.

```

import java.util.*;
class Sum_Array
{
    public static void main(String Args[])
    {
        Scanner input=new Scanner(System.in);
        int i;
        int array_1[]=new int[10];
        int array_2[]=new int[10];
        int sum[]=new int[10];
        for(i=0;i<10;i++)
        {
            System.out.print("Enter the "+(i+1)+" element for the first array : ");
            array_1[i]=input.nextInt();
            System.out.print("Enter the "+(i+1)+" element for the second array : ");
            array_2[i] = input.nextInt();
            sum[i]=array_1[i]+array_2[i];
        }
        System.out.println("The sum of the elements.");
        for(int value:sum)
        {
            System.out.println(value);
        }
    }
}

```

10. Write a program that asks the user to enter an integer (n). Create an arrays with 'n' elements.

Ask the user to enter the 'n' double value to populate the array.

Create another array which stores the square root of the number. Display both the arrays.

```

import java.util.*;
class Square_Root
{
    public static void main(String Args[])
    {
        Scanner input=new Scanner(System.in);
        System.out.print("Enter the value for 'n' : ");
        int n=input.nextInt();
        double array[]=new double[n];
        double sqrt[]=new double[n];
        int i;
        for(i=0;i<n;i++)
        {
            System.out.print("Enter the "+(i+1)+" element : ");
            array[i]=input.nextDouble();
            sqrt[i]=Math.sqrt(array[i]);
        }
        for(i=0;i<n;i++)
        {
            System.out.println(array[i]+"\\t"+sqrt[i]);
        }
    }
}

```

11. The annual examination result of 50 students in a class is tabulated as follows:

Roll Number | Subject A | Subject B | Subject C

Write a program to read the data, calculate and display the following:

(a) Average marks obtained by all students

(b) Print the roll number and average marks of the students whose average marks is above 80.

(c) Print the roll number and average marks of the students whose average marks if below 40.

```

import java.util.*;
class Marks_Average
{
    public static void main(String Args[])
    {
        Scanner input=new Scanner(System.in);
        int i;
        int subject_a[] = new int[50];
        int subject_b[] = new int[50];
        int subject_c[] = new int[50];
        double average[] = new double[50];
        for(i=0;i<50;i++)
        {
            System.out.print("Enter the marks scored by roll number "+(i+1)+" in Subject A : ");
            subject_a[i]=input.nextInt();
            System.out.print("Enter the marks scored by roll number " + (i + 1) + " in Subject B : ");
            subject_b[i] = input.nextInt();
            System.out.print("Enter the marks scored by roll number " + (i + 1) + " in Subject C : ");
            subject_c[i] = input.nextInt();
            average[i]=(subject_a[i]+subject_b[i]+subject_c[i])/3;
        }
        for(i=0;i<50;i++)
        {
            System.out.println("The average marks of roll number "+(i+1)+" is "+(average[i]));
        }
        System.out.println("The roll numbers which have a average marks of more than 80 are");
        for(i=0;i<50;i++)
        {
            if(average[i]>80)
            {
                System.out.println(i+1);
            }
        }
        System.out.println("The roll numbers which have a average marks of less than 40 are");
        for (i = 0; i < 50; i++)
        {
            if (average[i] < 40)
            {
                System.out.println(i+1);
            }
        }
    }
}

```

```
}  
}
```

12. Write a program to input 10 numbers in a single dimensional array. Input a number and using Binary Search Technique check whether the number is present in the array or not.

```
import java.util.*;

class Binary_Search
{
    public static void main(String Args[])
    {
        Scanner input = new Scanner(System.in);
        int array[] = new int[10];
        int i;
        for (i = 0; i < 10; i++)
        {
            System.out.print((i + 1) + " element : ");
            array[i] = input.nextInt();
        }
        Arrays.sort(array);
        System.out.print("Enter the number you want to search for : ");
        int search=input.nextInt();
        boolean isPresent=false;
        int low = 0, high = 9;
        while (low <= high)
        {
            int mid = (low+high) / 2;
            if (array[mid] == search)
            {
                System.out.println("The element is present at index " + mid);
                isPresent=true;
                break;
            }
            else if (array[mid] < search)
            {
                low = mid + 1;
            }
            else
            {
                high = mid - 1;
            }
        }
        if(!isPresent)
        {
            System.out.println("The element is not present in the array.");
        }
    }
}
```

```
}  
}
```

13. Write a program to input 10 numbers in a Single Dimensional Array.

Using Bubble Sort technique arrange the array in ascending order and display the sorted array.

Input a number and using linear search check whether the element is present in the array or not.

Use the following methods:

(i) void input()

(ii) void sort()

(iii) void search()

(iv) void display()

Create a main method and create the object of the class and call other functions.

```

import java.util.*;
class Bubble_Linear
{
    int number[]=new int[10];
    int search;
    boolean isPresent=false;

    void input()
    {
        Scanner input=new Scanner(System.in);
        int i;
        for(i=0;i<10;i++)
        {
            System.out.print("Enter the "+(i+1)+" number : ");
            number[i]=input.nextInt();
        }
        System.out.print("Enter the number you want to search for : ");
        search=input.nextInt();
    }

    void sort()
    {
        int i,j;
        for (i=0;i<10;i++)
        {
            for (j=0;j<9-i;j++)
            {
                if (number[j] < number[j + 1])
                {
                    int temporary = number[j];
                    number[j] = number[j + 1];
                    number[j + 1] = temporary;
                }
            }
        }
    }

    void search()
    {
        for(int i=0;i<10;i++)
        {
            if(number[i]==search)
            {

```

```

        isPresent=true;
    }
}

void display()
{
    System.out.println("The sorted array : ");
    for(int value:number)
    {
        System.out.print(value+" ");
    }
    System.out.println();
    if(isPresent)
    {
        System.out.println("The number you searched for is present in the array.");
    }
    else
    {
        System.out.println("The number you searched for is not present in the array.");
    }
}

public static void main(String Args[])
{
    Bubble_Linear object = new Bubble_Linear();
    object.input();
    object.sort();
    object.search();
    object.display();
}
}

```

14. Write a program to input 10 names in a Single Dimensional Array.

Input a name and using binary search technique check whether the element is present in the array or not.


```

import java.util.*;
class Linear_Names
{
    public static void main(String Args[])
    {
        Scanner input=new Scanner(System.in);
        String names[]=new String[10];
        int i;
        for(i=0;i<10;i++)
        {
            System.out.print("Enter the "+(i+1)+" name: ");
            names[i]=input.next();
        }
        System.out.print("Write the name you want to search for: ");
        String search=input.next();
        boolean isPresent=false;
        for(i=0;i<10;i++)
        {
            if(search.equalsIgnoreCase(names[i]))
            {
                isPresent=true;
            }
        }
        if(isPresent)
        {
            System.out.println("The name you searched for "+search+" is present in the array.");
        }
        else
        {
            System.out.println("The name you searched for "+search+" is not present in the array");
        }
    }
}

```

15. Write a program to input the numbers in a 4 x 5 array and find the biggest element in each row.

```

import java.util.*;
class Biggest_Row_Element
{
    public static void main(String Args [])
    {
        int[][] array = new int[4][5];
        Scanner input = new Scanner(System.in)
        System.out.println("Enter 4 x 5 array elements:");
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 5; j++)
            {
                System.out.print("Enter element at position [" + i + "][" + j + "]: ");
                array[i][j] = input.nextInt();
            }
        }
        int[] maxElements = new int[4];
        for (int i = 0; i < 4; i++)
        {
            int max = array[i][0];
            for (int j = 1; j < 5; j++)
            {
                if (array[i][j] > max)
                {
                    max = array[i][j];
                }
            }
            maxElements[i] = max;
        }
        System.out.println("The biggest element in each row:");
        for (int i = 0; i < 4; i++)
        {
            System.out.println("Row " + i + ": " + maxElements[i]);
        }
    }
}

```

16. Write a program to input the numbers in a 4 x 4 array and display the sum of its left and right diagonals.

```

import java.util.*;
class Diagonal_Sum
{
    public static void main(String[] args)
    {
        int[][] array = new int[4][4];
        Scanner input = new Scanner(System.in)
        System.out.println("Enter 4 x 4 array elements:");
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 4; j++)
            {
                System.out.print("Enter element at position [" + i + "][" + j + "]: ");
                array[i][j] = input.nextInt();
            }
        }
        int Left_Sum = 0;
        for (int i = 0; i < 4; i++)
        {
            Left_Sum += array[i][i];
        }
        int Right_Sum = 0;
        for (int i = 0; i < 4; i++)
        {
            Right_Sum += array[i][3 - i];
        }
        System.out.println("Sum of left diagonal: " + Left_Sum);
        System.out.println("Sum of right diagonal: " + Right_Sum);
    }
}

```

17. Write a program to input numbers in a 4 x 4 array, display the array. Replace all the boundary elements with '0' and print the new array.

```

import java.util.*;
class Boundary_Replace
{
    public static void main(String Args[])
    {
        Scanner input=new Scanner(System.in);
        int multi_array[][]=new int[4][4];
        int i,j;
        for(i=0;i<4;i++)
        {
            for(j=0;j<4;j++)
            {
                System.out.print("Enter ["+(i)+"] ["+(j)+"] element : ");
                multi_array[i][j]=input.nextInt();
            }
        }
        System.out.println("The array");
        for (i = 0; i < 4; i++)
        {
            for (j = 0; j < 4; j++)
            {
                System.out.print(multi_array[i][j]+" ");
            }
            System.out.println();
        }
        for(i=0;i<4;i++)
        {
            for(j=0;j<4;j++)
            {
                if(i==0||j==0||i==3||j==3)
                {
                    multi_array[i][j]=0;
                }
            }
        }
        System.out.println("The boundary elements replaced.");
        for (i = 0; i < 4; i++)
        {
            for (j = 0; j < 4; j++)
            {
                System.out.print(multi_array[i][j]+" ");
            }
            System.out.println();
        }
    }
}

```

```
    }  
  
    }  
}
```

18. The co-ordinates of a point P on a two dimensional place can be represented by P(x, y) with 'x' as the x co-ordinate and y as the y co-ordinate. The co-ordinates of midpoint of two points P1(x1, y1) and P2(x2, y2) can be calculated as P(x, y) where

- $x = (x1 + x2)/2$
- $y = (y1 + y2)/2$

Design a class with the following details

Class Name: Point

Data members/ Instance Variables:

- **x** : Stores the x co-ordinates
- **y** : Stores the y co-ordinates

Member functions

- **Point()** : Constructor to initialize x = 0, y = 0;
- **void readPoint()** : Accept the co-ordinates x and y of points
- **Point midPoint(Point A, Point B)** : Calculates and returns the midpoint of the two points A and B.
- **void displayPoint()** : Displays the co-ordinates of a point.

Specify the class **Point** giving details of the constructor Point(), member functions void readPoint(), Point midPoint(Point, Point) and void displayPoint() along with the main() function to create an object and call the functions accordingly to calculate the midpoint between any two given points.

```

import java.util.*;
class Point
{
    int x,y;
    Point()
    {
        x = 0;
        y = 0;
    }
    void readpoint()
    {
        Scanner input = new Scanner (System.in);
        System.out.print("x : ");
        x = input.nextInt();
        System.out.print("y : ");
        y = input.nextInt();
    }
    Point midpoint(Point A, Point B)
    {
        Point answer=new Point();
        answer.x = (A.x + B.x)/2;
        answer.y = (A.y + B.y)/2;
        return(answer);
    }
    void displaypoint()
    {
        System.out.print("x : "+ x);
        System.out.print("y : "+ y);
    }
    public static void main(String Args[])
    {
        Point Point_A = new Point();
        Point Point_B = new Point();
        Point Midpoint = new Point();
        System.out.println("For point A");
        Point_A.readpoint();
        System.out.println("For point B");
        Point_B.readpoint();
        System.out.println("Midpoint of the two points.");
        Midpoint.midpoint(Point_A, Point_B);
        Midpoint.displaypoint();
    }
}

```

```
}  
}
```

19. An angle can be measured in degrees and minutes.

Example: Angle A = 70 degrees and 35 minutes

Angle B = 50 degrees and 40 minutes

Now find the sum of these two angles

Angle C = 70 degrees and 35 minutes + 50 degrees and 40 minutes

= 121 degrees and 14 minutes (As 1 degree = 60 minutes)

A class called **Angle** has been defined to calculate the angle related functions. Some of the functions/ methods are shown below:

Class Name: Angle

Data members/ Instance Variables:

- dd : Degrees
- mm : Minutes

Member functions

- Angle() : Constructor to initialize dd = 0, mm = 0;
- void readAngle() : To read angle as dd mm;
- void displayAngle() : To display angle as dd mm;
- void sumAngle(Angle A, Angle B) : To find sum of the two angles in degrees and
- void diffAngle(Angle A, Angle B) : To find the difference between the two angle

Specify the class **Angle** giving details of the constructor, **void sumAngle(Angle A, Angle B)**, **void diffAngle(Angle A, Angle B)** only. You may assume the other functions are written for you.

```
import java.util.*;
class Angle
{
    int dd, mm;

    Angle()
    {
        dd = 0;
        mm = 0;
    }

    void readAngle()
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the angle in degree : ");
        dd = input.nextInt();
        System.out.print("Enter the angle in minutes : ");
        mm = input.nextInt();
    }

    void displayAngle()
    {
        System.out.println("Degree : "+dd);
        System.out.println("Minutes : "+mm);
    }

    void sumAngle(Angle A, Angle B)
    {
        this.dd = A.dd + B.dd;
        this.mm = A.mm + B.mm;
        if (this.mm >= 60)
        {
            this.mm = this.mm - 60;
            this.dd = this.dd + 1;
        }
    }

    void diffAngle(Angle A, Angle B)
    {
        if (A.mm < B.mm)
        {
            A.dd = A.dd - 1;
            A.mm = A.mm + 60;
        }
    }
}
```



```

    }
    this.dd = A.dd - B.dd;
    this.mm = A.mm - B.mm;
}

public static void main(String Args[])
{
    Angle A = new Angle();
    A.readAngle();
    Angle B = new Angle();
    B.readAngle();

    Angle C = new Angle();
    C.sumAngle(A,B);
    System.out.println("The sum of the two angles is :");
    C.displayAngle();

    Angle D = new Angle();
    D.diffAngle(A,B);
    System.out.println("The difference between the two angles is: ");
    D.displayAngle();
}
}

```

20. Specify the class **WordPile** giving the details of the constructor, **void pushChar(char c)** and **char popChar()**. The main function and the algorithm need not be written. Write a program in Java for conversion of WordPile into stack.

```

import java.util.*;
class WordPile
{
    int top, size;
    char ST[];

    WordPile(int n)
    {
        top = -1;
        size = n;
        ST = new char [size];
    }

    void pushChar(char c)
    {
        if (top == (size-1))
        {
            System.out.println("Overflow.");
        }
        else
        {
            top = top+1;
            ST [top] = c;
            System.out.println(c + " pushed into WordPile.");
        }
    }

    void popChar()
    {
        if (top == -1)
        {
            System.out.println("The stack is empty.");
        }
        else
        {
            char v = ST [top];
            top = top-1;
            System.out.println(v + " character popped from WordPile");
        }
    }
}

```

21. An array containing the marks of 50 students in ascending order is to be pushed into the stack.

Define a class **Array_to_Stack** with the following details:

Class Name: Array_to_Stack

Data members/ Instance Variables

- int m : To store the marks
- int st[] : To store stack elements
- int cap : Maximum capacity of array and stack
- int top : To point the index of the topmost element of the stack

Member functions

- Array_to_Stack (int n) : Parametrized constructor to initialize cap = n and top
- void inputMarks() : To input marks from the user and store it in the array n[]
- void pushMarks(int v) : To push the marks into the stack at top location if pos
- int popMarks() : To return marks from the stack if possible otherwise return 99
- void display() : To display stack element

Specify the class **Array_to_Stack** giving details of the constructor(int n), **void inputMarks()**, **void pushMarks()**, **void popMarks()** and **void display()**.

```

import java.util.*;
class Array_to_Stack
{
    int m;
    int n[];
    int st[];
    int cap;
    int top;

    Array_to_Stack(int n)
    {
        cap = n;
        top = -1;
    }

    void inputMarks()
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the marks of the students.");
        for (int i =0; i<50; i++)
        {
            System.out.print("Marks for "+ (i+1)+" : ");
            n[i] = input.nextInt();
            pushMarks(n[i]);
        }
        Arrays.sort(st);
    }

    void pushMarks( int v)
    {
        if (top == (cap-1))
        {
            System.out.println("Not Possible.");
        }
        else
        {
            top = top+1;
            st[top] = v;
        }
    }

    void popMarks()
    {

```

```

    if (top == -1)
    {
        System.out.println("9999");
    }
    else
    {
        int b = st[top];
        top = top-1;
        System.out.println(b+" popped out of the stack.");
    }
}

void display()
{
    System.out.println("The marks scored by 50 students are : ");
    for(int i=0; i<=top; i++)
    {
        System.out.print((i+1)+ " : "+st[i]+"\\n");
    }
}

public static void main(String Args[])
{
    Scanner input = new Scanner(System.in);
    Array_to_Stack object = new Array_to_Stack(50);
    object.inputMarks();
    while(true)
    {
        System.out.println("\\nMenu for Array to Stack");
        System.out.println("1.Pop items");
        System.out.println("2.Display Stack");
        System.out.println("3.Exit");
        System.out.print("Enter your choice: ");
        int choice = input.nextInt();
        switch(choice)
        {
            case 1:
                object.popMarks();
                break;
            case 2:
                object.display();
                break;
            case 3:

```

```
        System.out.println("Exiting the program.");
        System.exit(0);
        break;
    default:
        System.out.println("Invalid choice.");
    }
}
}
```

22. Write a program in java to make various operations in a queue.

```

import java.util.*;
class Queue_Operations
{
    int rear;
    int front;
    int size;
    int q[];

    Queue_Operations(int n)
    {
        rear = -1;
        front = -1;
        q = new int[n];
        size = n;
    }

    void enqueue(int item)
    {
        if (rear == (size-1))
        {
            System.out.println("Queue Overflow.");
        }
        else
        {
            if(front == -1)
            {
                front = 0;
            }
            rear = rear+1;
            q[rear] = item;
        }
    }

    void dequeue()
    {
        if(front == -1)
        {
            System.out.println("The queue is empty.");
        }
        else
        {
            if(front>rear)
            {

```

```

        front = -1;
        rear = -1;
    }
    int dq = q[front];
    front++;
    System.out.println(dq+" dequeued from the queue.");
}
}

```

```

void display()
{
    if (front == -1)
    {
        System.out.println("The queue is empty.");
    }
    else
    {
        System.out.println("Elements of the queue are : ");
        for (int i=front ; i<=rear ; i++ )
        {
            System.out.print(q[i]+" ");
        }
        System.out.println();
    }
}
}

```

```

public static void main(String Args[])
{
    Scanner input = new Scanner(System.in);
    System.out.print("Enter the size of the array : ");
    int s = input.nextInt();
    Queue_Operations object = new Queue_Operations(s);
    while (true)
    {
        System.out.println("\nQueue Operations:");
        System.out.println("1. Enqueue");
        System.out.println("2. Dequeue");
        System.out.println("3. Display Queue");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        int choice = input.nextInt();
        switch (choice)
        {

```



```

        case 1:
            System.out.print("Enter the element to enqueue: ");
            int element = input.nextInt();
            object.enqueue(element);
            break;
        case 2:
            object.dequeue();
            break;
        case 3:
            object.display();
            break;
        case 4:
            System.out.println("Exiting the program.");
            System.exit(0);
            break;
        default:
            System.out.println("Invalid choice.");
    }
}
}
}
}

```

23. A class **Student** defines student related information such as name, roll number and date of birth while another class **Marks** defines marks in various subjects, total percentage and grade of the student. The details of the classes are given below:

Class name : Student

Data members/ Instance Variables

- String name, DOB : String variables to store name and date of birth of student.
- int roll_num : Integer variable to store roll number

Member functions

- void inputData() : To input values of all the data members.
- void printData() : To display the values of all data members.

Class name : Marks

Data members/ Instance Variables

- int phy, chem, maths, cs, eng, total, per

Member functions

- void readData() : To read marks in all subjects out of 100
- void compute() : To find and store total marks, percentage and grade
- void printData() : To display the details

Specify the class **Student** with its functions. Using concept of inheritance specify the class marks with its functions.

Class Marks is derived from class Student.

```
import java.util.*;
class Student
{
    String name, DOB;
    int roll_num;

    void inputData()
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the name of the student : ");
        name = input.next();
        System.out.print("Enter the date of birth of the student : ");
        DOB = input.next();
        System.out.print("Enter the roll number of the student");
        roll_num = input.nextInt();
    }

    void printData()
    {
        System.out.println("Name : "+name);
        System.out.println("Date of Birth : "+DOB);
        System.out.println("Roll no. : "+roll_num);
    }
}
```

```

import java.util.*;
class Marks extends Student
{
    int phy, chem, maths, cs, eng, total;
    double per;

    void readData()
    {
        super.inputData();
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the marks of the given subjects : ");
        System.out.print("Physics : ");
        phy = input.nextInt();
        System.out.print("Chemistry : ");
        chem = input.nextInt();
        System.out.print("Mathematics : ");
        maths = input.nextInt();
        System.out.print("Computer Science : ");
        cs = input.nextInt();
        System.out.print("English : ");
        eng = input.nextInt();
    }

    void compute()
    {
        total = (phy + chem + maths + cs + eng);
        per = (total/500.0)*100;
    }

    void printData()
    {
        super.printData();
        System.out.print("Physics : " +phy);
        System.out.print("Chemistry : " +chem);
        System.out.print("Mathematics : " +maths);
        System.out.print("Computer Science : " +cs);
        System.out.print("English : " +eng);
        System.out.print("Total : " +total);
        System.out.print("Percentage : " +per);
    }

    public static void main(String Args[])
    {

```

```

        Marks object = new Marks();
        object.readData();
        object.compute();
        object.printData();
    }
}

```

24. A super class **Detail** has been defined to store the details of the customer. Define a subclass **Bill** to compute the monthly telephone charge of the customer as per the chart given below:

Number of Calls	Rate
1-100	Only Rental Charge
101-200	60 Paise per call + Rental Charge
201-300	80 Paise per call + Rental Charge
Above 300	1 Rupee per call + Rental Charge

The details of both classes are given below

Class name : Detail

Data members/ Instance Variables

- String name : To store the name of the customer
- String address : To store the address of the customer
- long tel_no : To store the telephone number of the customer
- double rental : To store the monthly rental charge

Member functions

- Detail() : Parametrized constructor to assign values to the data members
- void show() : To show the details of the customer

Class name : Bill

Data members/ Instance Variables

- int n : To store the number of phone calls
- double amt : To store the amount to be paid by the customer

Member functions

- Bill() : Parametrized constructor to assign values to data members and to initialize
- void call() : Calculates the monthly telephone bill as per the charge given above
- void show() : To display the detail of the customer and amount to be paid.

Specify the class **Detail** giving details of the constructor() and **void show()**. Using the concept of inheritance , specify the class **Bill** giving details of the constructor(), **void call()** and **void show()**. The main function and algorithm need not be written.

```
import java.util.*;
class Detail
{
    String name, address;
    long tel_no;
    double rental;

    Detail(String n, String a, long t, double r)
    {
        name = n;
        address = a;
        tel_no = t;
        rental = r;
    }

    void show()
    {
        System.out.println("Name : "+name);
        System.out.println("Address : "+address);
        System.out.println("Telephone number : "+tel_no);
        System.out.println("Rental Charge : "+rental);
    }
}
```

```

import java.util.*;
class Bill extends Detail
{
    int n;
    double amt;

    Bill(String nam, String add, long tel, double ren, int call)
    {
        super(nam,add,tel,ren);
        n = call;
        amt = 0.0;
    }

    void call()
    {
        if(n>=1 && n<=100)
        {
            amt = rental;
        }
        else if (n>=101 && n<=200)
        {
            amt = (0.6*n)+rental;
        }
        else if (n>=200 && n<=300)
        {
            amt = (0.8*n)+rental;
        }
        else
        {
            amt = n + rental;
        }
    }

    void show()
    {
        super.show();
        System.out.println("The amount to be paid by the customer :"+amt);
    }
}

```

25. A class **Admission** contains the admission number of 100 students. Some of the data members/ member functions are given below:

Class name: Admission

Data members/ Instance Variables

- `int Ad_no[]` : Integer array to store admission number

Member functions

- **Admission()** : Constructor to initialize the array elements
- **void fill_Array** : To accept element of array and rearrange in ascending order.
- **int binSearch(int l, int u, int v)** : To search for particular admission number(v) using binary search and recursive techniques and return 1 if found and -1 if not found

Specify the class **Admission** giving details of the constructor, **void fill_Array()** and **int binSearch(int l, int u, int v)**. Define the main function and call the functions according to enable the task.

```

import java.util.*;
class Admission
{
    int Ad_no[];

    Admission()
    {
        Ad_no = new int[100];
    }

    void fill_array()
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the admission number of the students : ");
        for(int i=0; i<100; i++)
        {
            System.out.print("[ "+(i+1)+" ] : ");
            Ad_no[i] = input.nextInt();
        }
        Arrays.sort(Ad_no);
    }

    int binSearch(int l, int u, int v)
    {
        if (l <= u)
        {
            int mid = l + (u - l) / 2;
            if (Ad_no[mid] == v)
            {
                return 1;
            }
            else if (Ad_no[mid] < v)
            {
                return binSearch(mid + 1, u, v);
            }
            else
            {
                return binSearch(l, mid - 1, v);
            }
        }
        return -1;
    }
}

```



```
public static void main(String Args[])
{
    Scanner input = new Scanner(System.in);
    Admission object = new Admission();
    object.fill_array();
    System.out.print("Enter the admission number you want to search for : ");
    int search = input.nextInt();
    int result = object.binSearch(0,99,search);
    if(result == -1)
    {
        System.out.println(search+" is not present in the list of admission number");
    }
    else
    {
        System.out.println(search+" is present in the list of admission number");
    }
}
}
```