Computer Assignment / Project Questions XII

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.
2. Write a program to input the sentence and reprint it after exchanging the first and last words of the sentence.
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.
4. Write a program to input a sentence and print those words which do not contain a vowel, also count those words.
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.
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)
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)
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.
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.
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.
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 avearage marks of the students whose average marks if below 40.
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.

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.
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.
15. Write a program to input the numbers in a 4 x 5 array and find the biggest element in each row.
16. Write a program to input the numbers in a 4 x 4 array and display the sum of its left and right diagonals.
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.

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.

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
 - **void diffAngle(Angle A, Angle B) :** To find the difference between the two a
```

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.

20. Specify the class **WordPile** giving the details of the constructor,**void pushChar(char)** and **char popChar()**. The main function and the algorithm need not be written. Write a program in Java for conversion of WordPile into stack.
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
 - **void inputMarks() :** To input marks from the user and store it in the array
 - **void pushMarks(int v) :** To push the marks into the stack at top location if
 - **int popMarks() :** To return marks from the stack if possible otherwise retur
 - **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()**.

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

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
    200 - 301 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
  - **int 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 i
  - **void call() :** Calculates the monthly telephone bill as per the charge given
  - **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.

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.