Name : Rachita Yalamali

Usn : 2GI20CS104

Termwork 1

Practice Problem 1.1 : It is required to store and analyze data about 6 car manufacturer's sales data in all the 12 months of a year. Demonstrate how you would store the data in a two dimensional matrix and do the following

1. Write a function to Find for a given car manufacturer, the month in which, maximum no. of cars are sold.

2. Write a function to Find the average number of cars sold for each car manufacturer

3. Write a function to Find the total number of cars sold for each car manufacturer.

4. Write a function to find standard deviation for a given car manufacturer

Assume – row index 0 - 'Maruti Suzuki', 1 – 'Hundai' 2 – 'Tata Motors' 3-'KIA' 4 – 'BMW' 5 – 'Renault'

Col index 0 –'Jan', 1-'Feb'…………………………………..11 –'Dec'

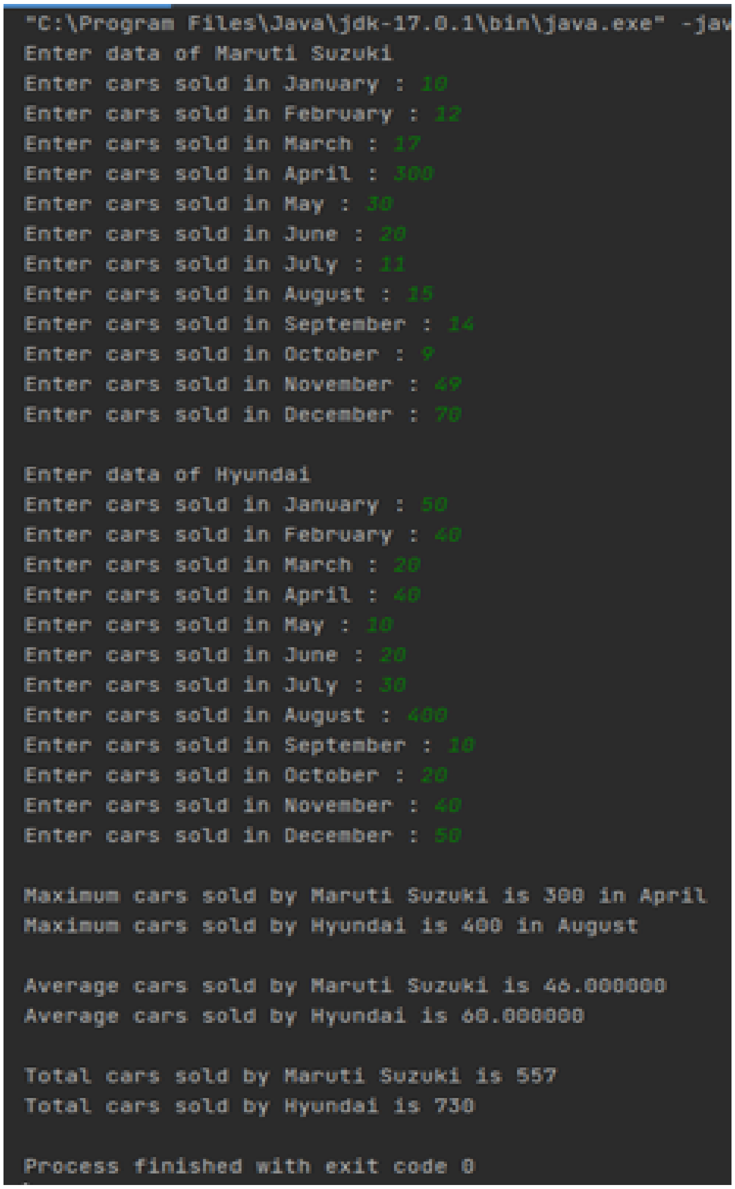
Demonstrate the working of the program with appropriate values for each car manufacturer and the months.

Program:

```
import java.util.Scanner;
public class tw1practice {
        public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String[] brands = {"Maruti Suzuki","Hyundai","Tata Motors","Kia","BMW","Renault"};
        String[] month = {"January","February","March","April","May","June","July",
                "August","September","October","November","December"};
        int[][] data = new int[2][12];
        for(int i=0;i<data.length;i++) {
        System.out.println("Enter data of "+brands[i]);
        for(int j=0;j<data[0].length;j++) {
                System.out.printf("Enter cars sold in %s : ",month[j]);
                data[i][j] = sc.nextInt();
        }
        System.out.println();
        }
        maxCarsSold(data,brands,month,data.length,data[0].length);
        avgCarsSold(data,brands,data.length,data[0].length);
        totalCarsSold(data,brands,data.length,data[0].length);
        }
        static void maxCarsSold(int[][] data, String[] brands,String[] month, int m, int n) {
        int max,ind;
        for(int i=0;i<m;i++) {
        max = data[i][0];
        ind = 0;
        for(int j=1;j<n;j++) {
                if(max<data[i][j]) {
                max = data[i][j];
                ind = j;
                }
        }
        System.out.printf("Maximum cars sold by %s is %d in %s\n",brands[i],max,month[ind]);
        }
        System.out.println();
        }
        static void avgCarsSold(int[][] data, String[] brands, int m, int n) {
        int sum;
        double avg;
        for(int i=0;i<m;i++) {
        sum = 0;
        avg=0.0;
        for(int j=0;j<n;j++) {
                sum += data[i][j];
        }
        avg = sum / n;
        System.out.printf("Average cars sold by %s is %f\n",brands[i],avg);
        }
        System.out.println();
        }
```

```
    static void totalCarsSold(int[][] data, String[] brands, int m, int n) {
    int sum;
    for(int i=0;i<m;i++) {
    sum = 0;
    for(int j=0;j<n;j++) {
        sum += data[i][j];
    }
    System.out.printf("Total cars sold by %s is %d\n",brands[i],sum);
        }
    }
}
```

Output:



```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" -jav
Enter data of Maruti Suzuki
Enter cars sold in January : 10
Enter cars sold in February : 12
Enter cars sold in March : 17
Enter cars sold in April : 300
Enter cars sold in May : 30
Enter cars sold in June : 20
Enter cars sold in July : 11
Enter cars sold in August : 15
Enter cars sold in September : 14
Enter cars sold in October : 9
Enter cars sold in November : 49
Enter cars sold in December : 70

Enter data of Hyundai
Enter cars sold in January : 50
Enter cars sold in February : 40
Enter cars sold in March : 20
Enter cars sold in April : 40
Enter cars sold in May : 10
Enter cars sold in June : 20
Enter cars sold in July : 30
Enter cars sold in August : 400
Enter cars sold in September : 10
Enter cars sold in October : 20
Enter cars sold in November : 40
Enter cars sold in December : 50

Maximum cars sold by Maruti Suzuki is 300 in April
Maximum cars sold by Hyundai is 400 in August

Average cars sold by Maruti Suzuki is 46.000000
Average cars sold by Hyundai is 60.000000

Total cars sold by Maruti Suzuki is 557
Total cars sold by Hyundai is 730

Process finished with exit code 0
```

Practice Problem 1.4 :

A company has 10 zonal sales offices in four zones namely, North, East, West and South. The company wants to organize the sales data of each of the office in each zone and find answers to queries such as,

1. Which office has performed the highest sales in each zone?
2. What is the average sales done by all the offices in each zone?
3. Which office among each zone is the poorly office?

You are required to answer the following:

1. How do you organize the above data?
2. How do you provide answers to the above queries?

Design a Java application for the same and demonstrate the correctness of the solution.

Program:

```
import java.util.Scanner;
import java.util.Random;
public class OneFour {
    public static void main(String[] args) {
    Scanner read = new Scanner(System.in);
    Random rand = new Random();
    int data[][] = new int[4][10];
    int p, i, j;
    int bestOff[] = new int[4];
    int worstOff[] = new int[4];
    int maxs = 0, mins = 9999;
    int max[] = new int[4];
    int min[] = new int[4];
    String zone[] = {"North", "South", "East", "West"};
    float average[] = new float[4];
    for (i = 0; i < 4; i++) {
    System.out.println("The zone is: " + zone[i]);
    float sum = 0;
    maxs = 0;
    mins = 9999;
    for (j = 0; j < 10; j++) {
        System.out.println("enter the sales made by office " + j + " :");
        data[i][j] = rand.nextInt(20, 80);
        System.out.print(data[i][j] + "\n");
```

```
          sum = sum + data[i][j];
          if (data[i][j] > maxs ) {
          maxs = data[i][j];
          bestOff[i] = j;
          }
          if (mins > data[i][j]) {
          mins = data[i][j];
          worstOff[i] = j;
          }
  }
  average[i] = sum / 10;
  max[i] = maxs;
  min[i] = mins;
  }
  for (i = 0; i < 4; i++) {
  System.out.println("\nThe best office in the " + zone[i] + " zone is:");
  System.out.print(bestOff[i]);
  System.out.println("\n this office made " + max[i] + " sales");
}
  System.out.println("\n");
  for (i = 0; i < 4; i++) {
  System.out.println("\nThe worst office in the " + zone[i] + " zone is:");
  System.out.print(worstOff[i]);
  System.out.println("\n this office made " + min[i] + " sales");
  }
```

```
    System.out.println("\n");
    for (i = 0; i < 4; i++) {
    System.out.println("\nThe average sales in the " + zone[i] + " zone is :");
    System.out.print(average[i]);
    }
    System.out.println("\n");
    }
}
```

Output:

```
The zone is: North
enter the sales made by offices :
77 71 42 46 50 31 62 22 42 20
The zone is: South
enter the sales made by offices :
73 43 59 79 21 54 55 25 59 23
The zone is: East
enter the sales made by offices :
35 67 65 35 26 73 36 21 55 38
The zone is: West
enter the sales made by offices :
70 37 50 37 41 41 45 43 68 66
The best office in the North zone is:
0
 this office made 77 sales
The best office in the South zone is:
3
 this office made 79 sales
The best office in the East zone is:
5
 this office made 73 sales
The best office in the West zone is:
0
 this office made 70 sales
 The worst office in the North zone is:
9
this office made 20 sales
The worst office in the South zone is:
4
this office made 21 sales
The worst office in the East zone is:
7
this office made 21 sales
The worst office in the West zone is:
1
this office made 37 sales

The average sales in the North zone is :
46.3
The average sales in the South zone is :
49.1
The average sales in the East zone is :
45.1
The average sales in the West zone is :
49.8
```

Name : Rachita Yalamali

Usn : 2GI20CS104

Termwork 2 :

Practice problem 2.1 :

Design a class by name myCircle to model Circle geometrical object with its center and radius that enables:

1. Initializing the center, radius and
2. Compute area, perimeter, and diameter of the circle object/s.

TASK 1: Identify member variable/s and their types

TASK 2: Identify Constructor/s along with their arguments (if any) to initialize the member variables

TASK 3: Identify the methods along with their arguments and return types.

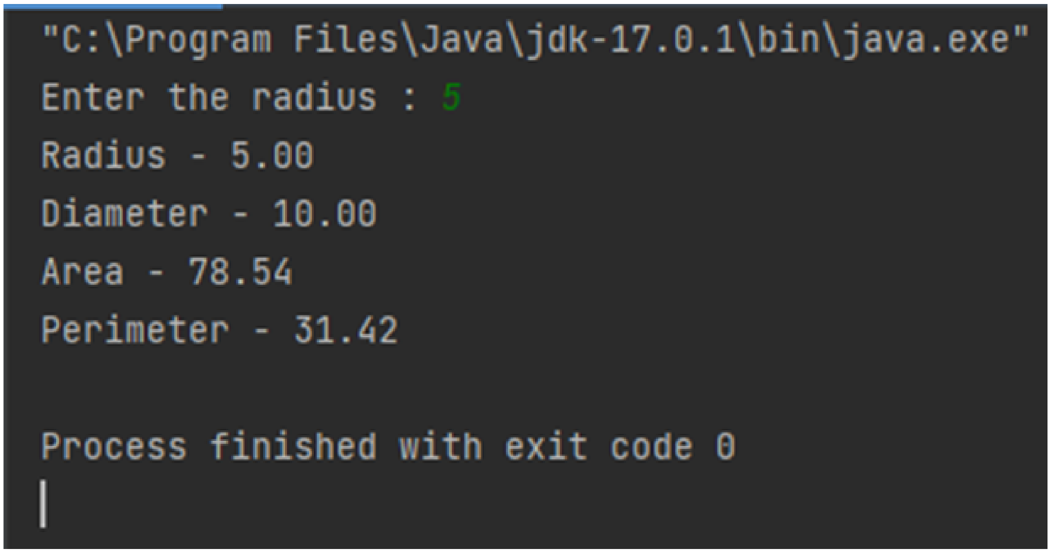
TASK 4: Identify member variable getters/setters (if needed)

Program:

```
import java.util.Scanner;
class myCircle{
        double radius,perimeter,area,diameter;
        myCircle(){
        this.radius = 0.0;
        }
        myCircle(double radius){
        this.radius = radius;
        }
        public void compute() {
        this.diameter = this.radius * 2;
        this.perimeter = 2*Math.PI*this.radius;
        this.area = Math.PI*this.radius*this.radius;
        }
        public void printCircle() {
        System.out.printf("Radius - %.2f\n",this.radius);
        System.out.printf("Diameter - %.2f\n",this.diameter);
        System.out.printf("Area - %.2f\n",this.area);
        System.out.printf("Perimeter - %.2f\n",this.perimeter);
        }
}

public class TwoOne {
        public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);

        myCircle c1 = new myCircle();

        System.out.print("Enter the radius : ");

        c1.radius = sc.nextDouble();

        c1.compute();

        c1.printCircle();

        }

}
```

Output:



```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe"
Enter the radius : 5
Radius - 5.00
Diameter - 10.00
Area - 78.54
Perimeter - 31.42

Process finished with exit code 0
```

Practice problem 2.3:

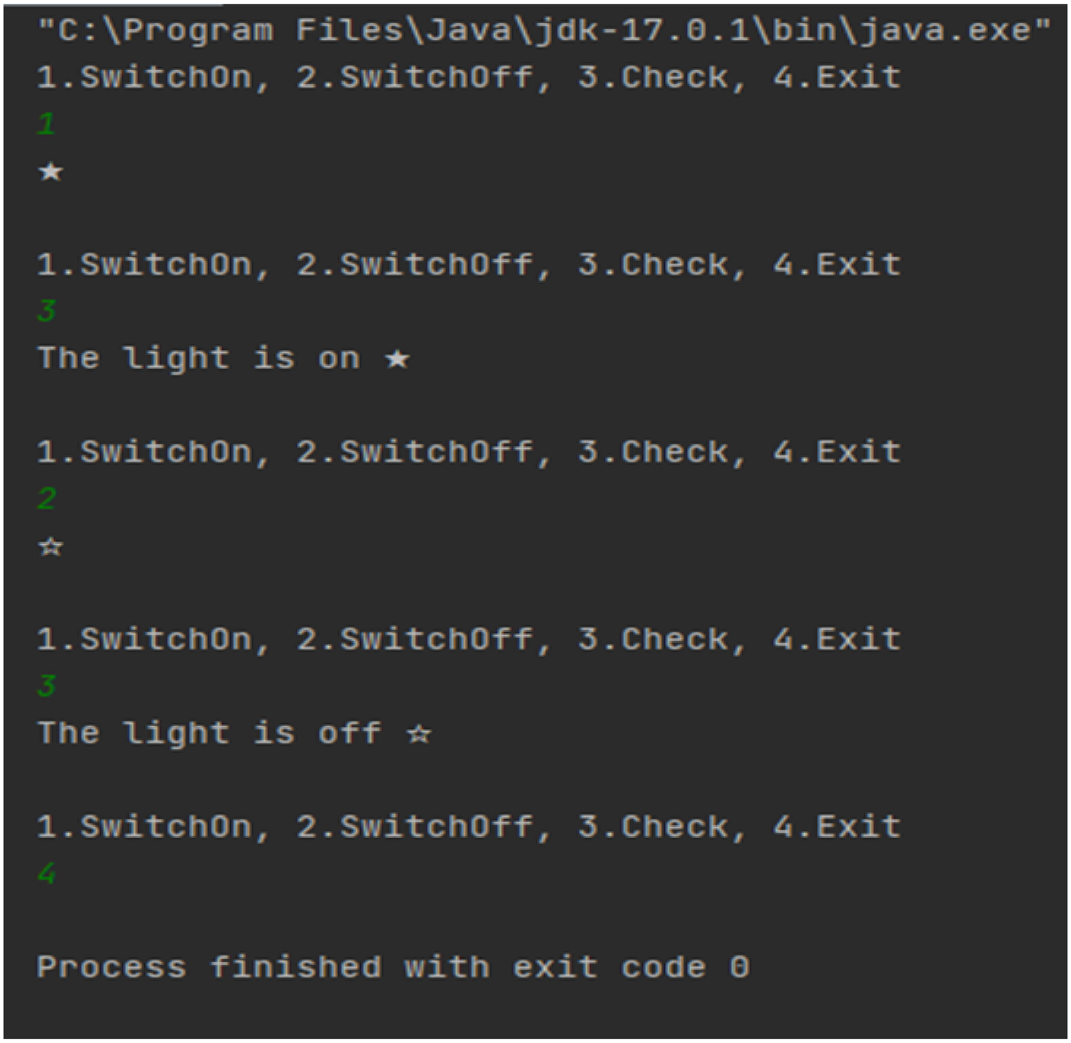
Write a Java program to define a class Lamp. It can be in on or off state. You can turn on and turn off lamp (behavior). It makes use of class and its member methods.

Program:

```
import java.util.Scanner;

import static java.lang.System.exit;

class Lamp {

        boolean status;

        void SwitchOn() {

                System.out.println("★\n");

        status = true;

}

void SwitchOff() {

        System.out.println("☆\n");

        status = false;

}

void state() {

        if (status)

        System.out.println("The light is on ★\n");

        else

        System.out.println("The light is off ☆\n");

        }

}

public class TwoTwo {

        static Scanner read = new Scanner(System.in);

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

```
    Lamp l1 = new Lamp();

    while (true) {

    System.out.println("1.SwitchOn, 2.SwitchOff, 3.Check, 4.Exit");

    int choice = read.nextInt();

    int br = 0;

    switch (choice) {

        case 1 -> l1.SwitchOn();

        case 2 -> l1.SwitchOff();

        case 3 -> l1.state();

        case 4 -> exit(0);

        default -> System.out.println("Unexpected Value");

            }

        }

    }

}
```

Output :



```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe"
1.SwitchOn, 2.SwitchOff, 3.Check, 4.Exit
1
★

1.SwitchOn, 2.SwitchOff, 3.Check, 4.Exit
3
The light is on ★

1.SwitchOn, 2.SwitchOff, 3.Check, 4.Exit
2
☆

1.SwitchOn, 2.SwitchOff, 3.Check, 4.Exit
3
The light is off ☆

1.SwitchOn, 2.SwitchOff, 3.Check, 4.Exit
4

Process finished with exit code 0
```

Name : Rachita Yalamali

Usn : 2GI20CS104

Termwork 3 :

Practice Problem 3.2 :

Define a class to represent a rectangle in which constructors and parameterized methods are used. It also has a method to compute area of rectangle.

i. First make a class rectangle in which we declare the parameterized constructor.

ii. Then demonstrate the use of parameterized method.

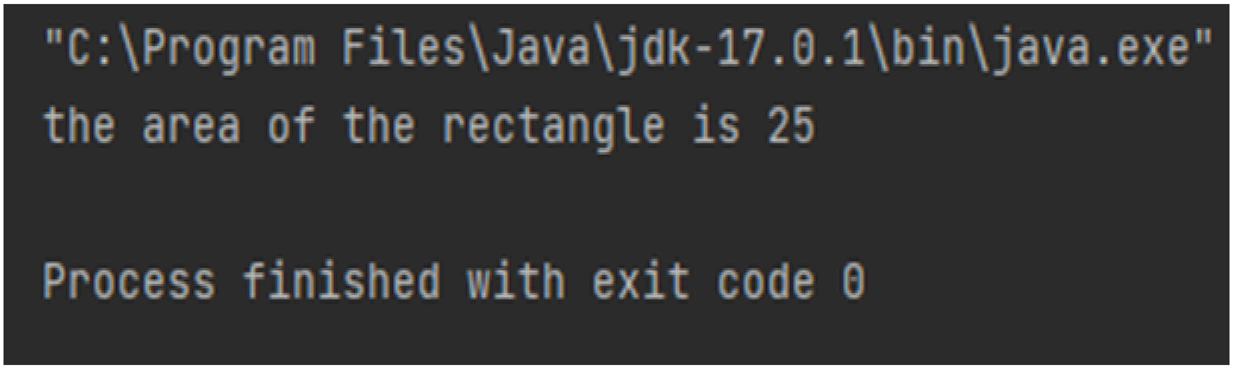
iii. Use a method to compute area of rectangle.

iv. Create a class to demonstrate the call of the methods in previously created class rectangle holding constructors, parameterized methods and method to compute area of rectangle.

Program :

```
class rectangle{
    int len,br;
    rectangle(){
    len=0;
    br=0;
    }
    rectangle(int len,int br){
    this.len=len;
    this.br=br;
    }
    void area(){
    System.out.println("the area of the rectangle is "+ len*br);
    }
}

public class ThreeTwo {
    public static void main(String[] args) {
    rectangle r1=new rectangle(5,5);
    r1.area();
    }
}
```

Output :



```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe"
the area of the rectangle is 25

Process finished with exit code 0
```

Practice Problem 3.3 :

Write a Java program to represent a Complex number. Include member functions to:

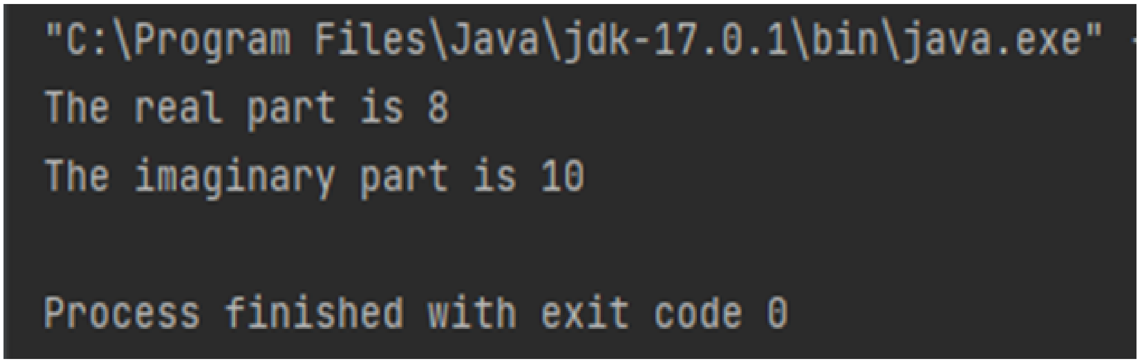
1. Initialize a complex number to a default value of zero (default constructor)
2. Initialize a complex number to a user defined value (parameterized constructor)
3. Add two complex numbers and return the result. (Parameterized method)
4. Subtract two complex numbers and return the result. (Parameterized method)
5. Display a complex number. (non-parameterized method)

```
}
```

Program :

```
class complex{
        int r,i;
        complex(){
        this.i=0;
        this.r=0;
        }
        complex(int r, int i){
        this.i=i;
        this.r=r;
        }
        complex add(complex c1,complex c2){
        complex temp=new complex();
        temp.r=c1.r+c2.r;
         temp.i=c1.i+c2.i;
        return temp;
        }
        complex sub(complex c1,complex c2){
        complex temp=new complex();
        temp.r=c1.r-c2.r;
        temp.i=c1.i-c2.i;
        return temp;
}
void display(){

	System.out.println("the real part is"+r);

	System.out.println("the imaginary part is"+i);

	}

}

public class ThreeThree {

	public static void main(String[] args) {

	complex c1 = new complex(4, 5);

	complex c2 = new complex(4, 5);

	complex result = new complex(0, 0);

	result = result.add(c1, c2);

	result.display();

	}

}
```

Output :



```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe"
The real part is 8
The imaginary part is 10

Process finished with exit code 0
```

Name : Rachita Yalamali

Usn : 2GI20CS104

Termwork 5 A :

Practice Problem 5.2 :

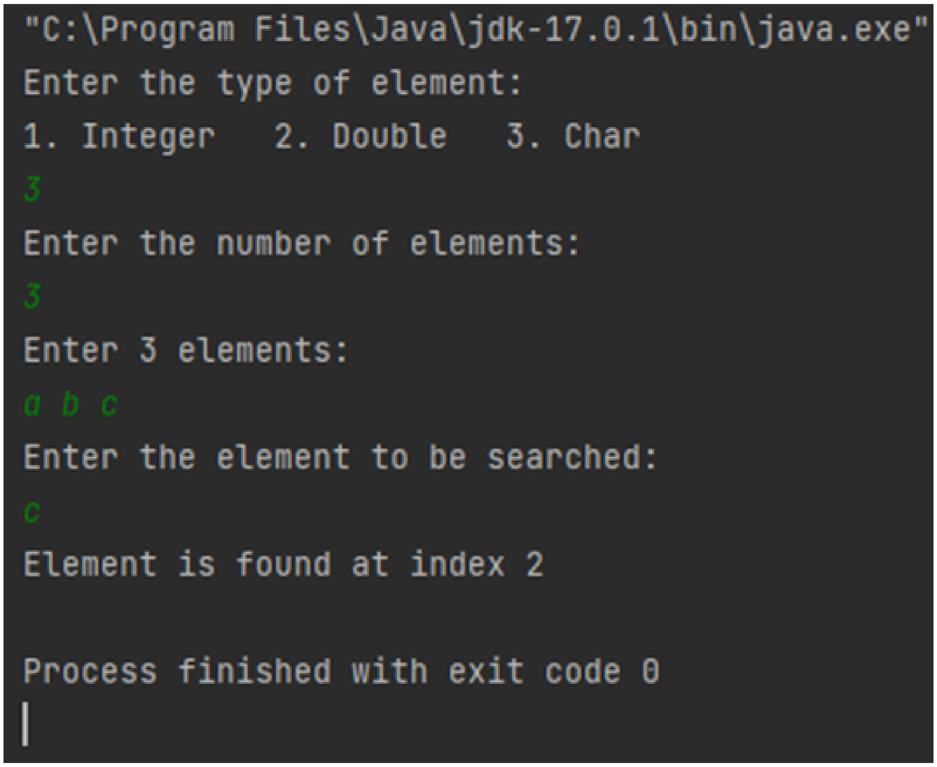
Implement a linear search function by using method overloading concept for an array of integers, double and character elements.

Program :

```
import java.util.Scanner

class MyLinearSearch {

        int linearSearch(int[] c, int key2) {

        for (int i = 0; i < c.length; i++) {

        if (c[i] == key2) {

                return i;

        }

        }

        return -1;

        }

        double linearSearch(double[] a, double key) {

        for (int i = 0; i < a.length; i++) {

        if (a[i] == key) {

                return i;

        }

        }

        return -1;

        }

        int linearSearch(char[] a, char key) {

        for (int i = 0; i < a.length; i++) {

        if (a[i] == key) {

                return i;
        }

        }

        return -1;

        }

}

public class FiveTwo {

        public static void main(String[] args) {

        MyLinearSearch m = new MyLinearSearch();

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the type of element: ");

        System.out.println("1. Integer \t 2. Double \t 3. Char");

        int choice = in.nextInt();

        System.out.println("Enter the number of elements: ");

        int n = in.nextInt();

        System.out.println("Enter " + n + " elements: ");

        switch (choice) {

        case 1 -> {

                int[] a = new int[n];

                for (int i = 0; i < n; i++) {

                a[i] = in.nextInt();

                }

                System.out.println("Enter the element to be searched: ");

                int key = in.nextInt();
                int position = m.linearSearch(a, key);
                check(position);
        }
        case 2 -> {
                double[] a1 = new double[n];
                for (int i = 0; i < n; i++) {
                a1[i] = in.nextDouble();
                }
                System.out.println("Enter the element to be searched: ");
                double key1 = in.nextInt();
                int position1 = (int) m.linearSearch(a1, key1);
                check(position1);
        }
        case 3 -> {
                char[] c = new char[n];
                for (int i = 0; i < n; i++) {
                c[i] = in.next().charAt(0);
        }
        System.out.println("Enter the element to be searched: ");
        char key2 = in.next().charAt(0);
         int position2 = (int) m.linearSearch(c, key2);
                check(position2);
        }
```

```
    }

    }

    public static void check(int position1) {

    if (position1 >= 0)

    System.out.println("Element is found at index " + (position1));

    else

    System.out.println("Element not found.\n");

    }

}
```

Output :



```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe"
Enter the type of element:
1. Integer   2. Double   3. Char
3
Enter the number of elements:
3
Enter 3 elements:
a b c
Enter the element to be searched:
c
Element is found at index 2

Process finished with exit code 0
```