ASSIGNMENT 3: LAB 2

Q1: Using the command line input arguments obtain 4 floating point numbers e.g. 3.141593.

- Arrange them in ascending order (*hint: use the compare() method of the Float class*).
- Print the numbers, correct to 2 decimal places (*hint: use the format() method of PrintStream class*).
- Print the total of the 4 numbers, correct to 2 decimal places (*hint: instead of the* '+' *operator, use the sum() method of Float class*)

Test cases:
0, -234, 6.89823, 2999999.000000001

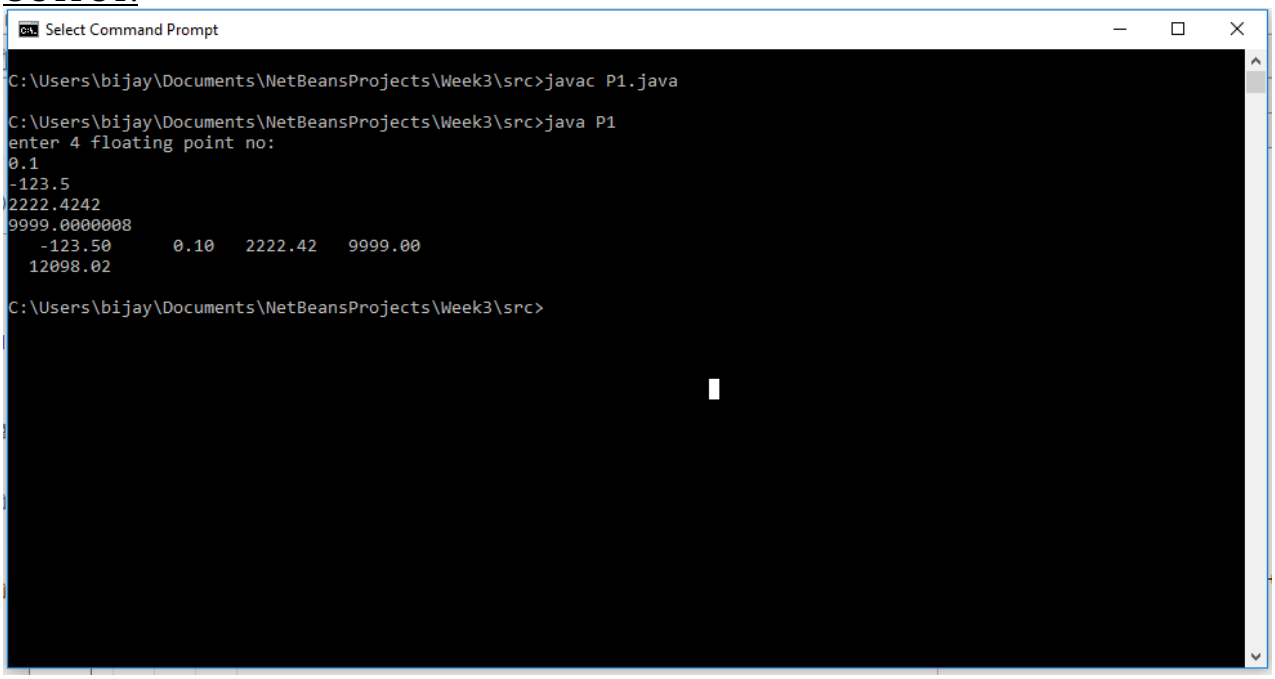
Expected outputs:
-234.00, 0.00, 6.90, 2999999.00
2999771.90

SOURCE CODE:

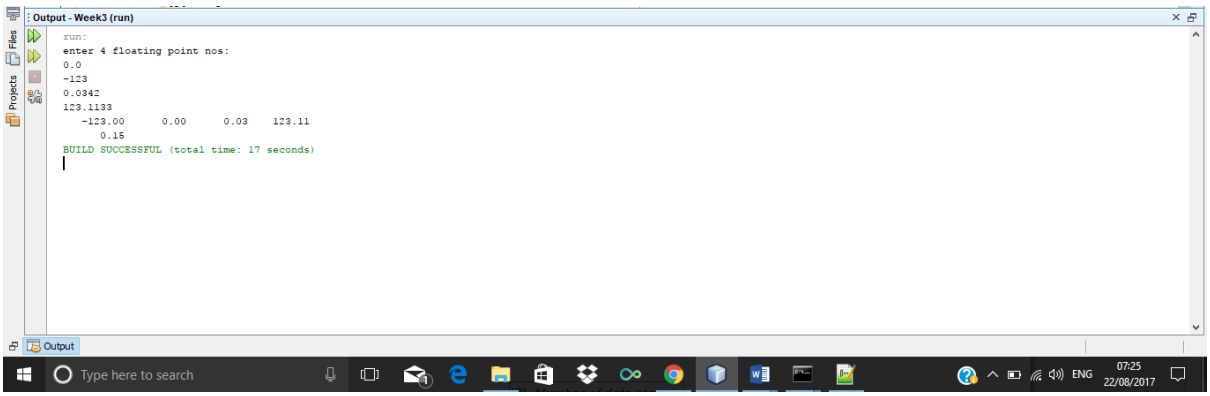
```
/*
author - harsh


*/
import java.util.Scanner;
public class P1 {
   public static void main(String[] args) {
      int i;
      float a[] = new float[4];
      Scanner sc = new Scanner(System.in);
      System.out.println("enter 4 floating point nos:");
      for (i = 0; i < 4; i++) {
         a[i] = sc.nextFloat();
      }
       float temp,sum=0;
      for (i = 0; i < 3; i++) {
         for(int j=i;j<4;j++)
```

```
            //ascending order arrangement
          if (Float.compare(a[i], a[j]) > 0) {
             temp=a[i];
             a[i]=a[j];
          a[j]=temp;}
          }
         for (i = 0; i < 4; i++){
            //print of the order
            System.out.format("%10.2f",+a[i] );
         sum+=a[i];}
            //print of the sum of the entered numbers.
       System.out.format("\n%10.2f%n",+sum );
  }
}
```

OUTPUT:



-using command prompt



-using NetBeans

Q2. The following methods work on an array which can be used to store up to 11 integers. From the second element to the last element of the array, they are used to store the integer data. The first element of the array is used to store the number of integer data that are stored in the array. Therefore, the array should be created to contain at most 11 elements. The array structure is shown below:

[0]	*Number of data stored*
[1]	*int data*
[2]	*int data*
[3]	*int data*
…	
…	
…	
…	
…	
…	
[10]	*int data*

Write the code for the following methods and then a main() method to test them. In the main() method, a menu that can support the following methods should be given: (1) initialize; (2) insert; (3) remove; (4) display; (5) quit. Then, the user selects an option from the menu. After the user has selected an option, the corresponding method will then be executed. If option (5) is not selected, then the menu will be repeated, and the user can select another option for execution.

SOURCE CODE

```
/*
author - harsh
*/
import java.util.Scanner;
public class P2
{
        int i,p,number,c=0;
public static void main(String args[])
```

```
{
        Scanner in=new Scanner(System.in);
        int arr[]=new int[11];
        int ch;
        P2 array=new P2();

        do
        {
                System.out.println("Enter "+"\n"+"1 for initialize"+"\n"+"2 for
insert"+"\n"+"3 for delete"+"\n"+"4 for display "+"\n"+"5 for quit");


          ch=in.nextInt();

                switch(ch)
                {
                case 1:
                   array.initialisation(arr);
                break;
                case 2:
                        array.insert(arr);
                        break;
                case 3:
                        array.delete(arr);
                        break;
                case 4:
                        array.show(arr);


                        break;

                case 5:
```

```
                System.out.println(" The End !!");
                break;
        default:
                System.out.println("Wrong input!! Please try Again. Exiting");
        }

    }while(ch!=5);
}
void initialisation(int a[])
{
    Scanner in=new Scanner(System.in);
    System.out.println("Enter 1st element of array");
  //position 1 of array
    a[0]=in.nextInt();


        System.out.println("Enter array elements");
        for(i=1;i<=a[0];i++)
        {
                a[i]=in.nextInt();
        }


}
//inserting a number
void insert(int a[])
{
    Scanner in=new Scanner(System.in);
    System.out.println("Enter number");
    number=in.nextInt();
```

```
        for(i=1;i<=a[0];i++)
        {
                if(number<a[i])
                {
                        c=1;
                        p=i;
                        break;
                }
        }
        if(c==1)
        {
        for(i=a[0]+1;i>p;i--)
        {
                a[i]=a[i-1];
        }
        a[i]=number;

        }
        else
                a[a[0]+1]=number;
        a[0]++;
      System.out.println("Operation - Display ");
        show(a);
}
void delete(int a[])
{
        Scanner in=new Scanner(System.in);
        System.out.println("Enter element to be deleted");
        number=in.nextInt();
        c=0;
        for(i=1;i<=a[0];i++)
        {
```

```
            if(a[i]==number)
            {
                  c=1;
                  p=i;
                  break;
            }
      }
      //checking if the number is found and therefore deleting it
      if(c==1)
      {
      for(i=p;i<a[0];i++)
      {
            a[i]=a[i+1];
      }
      a[0]--;
      show(a);
      }

      else
            System.out.println("Element not found");
    //Element not found to be deleted

}
//final output
void show(int a[])
{
      System.out.println("Display ALL the elements in Ascending Order");
      for(i=0;i<=a[0];i++)
            System.out.println(a[i]);
}

}
```

OUTPUT


Output - Week3 (run)
run:
Enter
1 for initialize
2 for insert
3 for delete
4 for display
5 for quit
1
Enter 1st element of array
3
Enter array elements
1
2
3
Enter
1 for initialize
2 for insert
3 for delete
4 for display
5 for quit
2
Enter number
4
Operation - Display
Display ALL the elements in Ascending Order
4
1


Output - Week3 (run)
3
Enter element to be deleted
2
Display ALL the elements in Ascending Order
3
1
3
4
Enter
1 for initialize
2 for insert
3 for delete
4 for display
5 for quit
4
Display ALL the elements in Ascending Order
3
1
3
4
Enter
1 for initialize
2 for insert
3 for delete
4 for display
5 for quit
5
The End !!