UNIVERSITI MALAYA

FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT

Faculty of Computer Science and Information Technology

LAB REPORT

VIVA 2

REPRESENT BY:
NEWBIE MEMERS

Name	Matric Number
TITUS TAN YU FAN	23004883
WONG HOONG LIANG	23053016
SIM JING JIA	23004919
LIM HONG YU	23004973
SOON MING HONG	23004950

Question 1

Problem description:

To check if the user's input is a valid password or not by using java method.

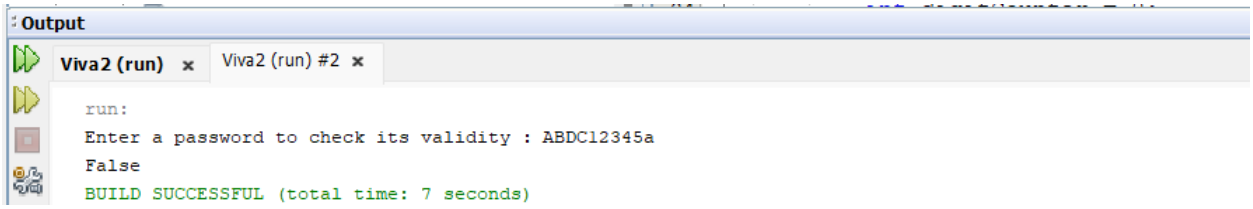
Solution:

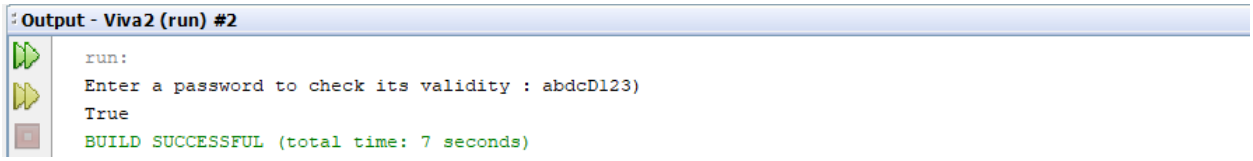
```
import java.util.Scanner;

public class Q1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a password to check its validity : ");
        String input = sc.nextLine();
        if (passwordChecker(input)) {
            System.out.println("True");
        } else {
            System.out.println("False");
        }
        sc.close();
    }

    private static boolean passwordChecker(String pw) {
        boolean length = false,
                CapNSmall = false,
                threeDigit = false,
                oneSChar = false;
        int Cap = 0, smallCaps = 0;
        int digitCounter = 0;
        int SCharCounter = 0;
        if (pw.length() >= 8) {
            length = true;
        }
        for (int i = 0; i < pw.length(); i++) {
            if (CapNSmall == false) {
                if (Character.isUpperCase(pw.charAt(i))) {
                    Cap++;
                } else if (Character.isLowerCase(pw.charAt(i))) {
                    smallCaps++;
                }
```

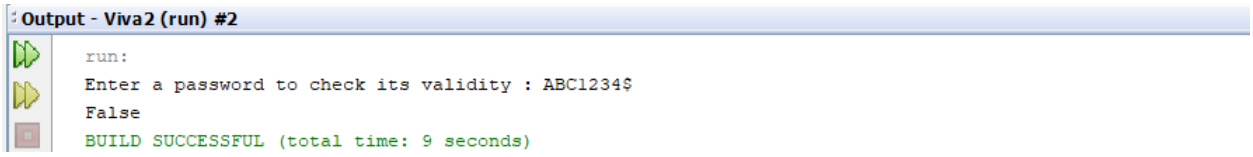
```
                if (Cap >= 1 && smallCaps >= 1) {
                    CapNSmall = true;
                }
            }

            if (pw.charAt(i) >= '0' && pw.charAt(i) <= '9') {
                digitCounter++;
                if (digitCounter >= 3) {
                    threeDigit = true;
                }
            }
            if (!Character.isLetterOrDigit(pw.charAt(i))) {
                SCharCounter++;
                if (SCharCounter == 1) {
                    oneSChar = true;
                }
            }
        }
        boolean validity;
        if (length && CapNSmall && threeDigit && oneSChar) {
            validity = true;
        } else
            validity = false;

        return validity;
    }

}
```

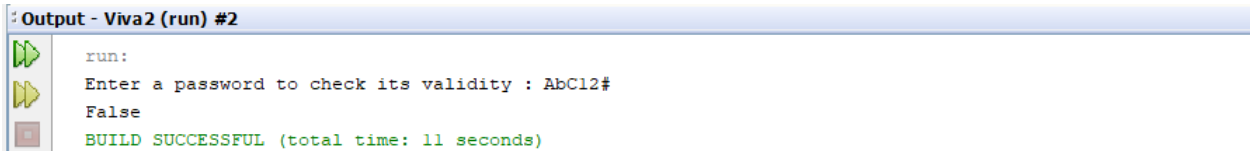
We use a scanner to read the user input. After that, the java program will run java method on the next line. In the java method, the password will be checked by using if else statement. The password that meets all rules will return true and false otherwise.

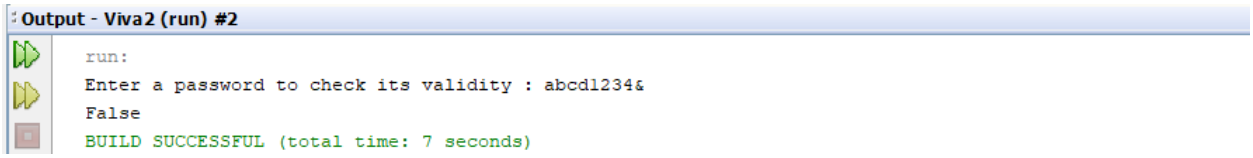
Sample Input & Output:











Question 2

2.Lili works as a data analyst at a customer service company. Currently, she is actively engaged in a project aimed at analyzing the median response time for customer inquiries and complaints. The primary goal of this project is to assess and improve the overall efficiency of their customer service operations.Lili has received two unsorted arrays of double values representing time data, and she is looking to calculate the median of this data. Can you provide a Java method that receives 2 double arrays as input and returns a double value as output to help her solve this problem?

Problem Description:

Receive 2 double array and return a median from the data.

Additional(Receive array.length)

Solution:

```
import java.util.Scanner;
public class Q2 {

	public static void main(String[] args) {
		int input1 = getArraySize("Enter the number of data to be key in
to array 1 : ");
		double[] array1 = getArrayInput(input1, "Enter the input into
array 1: ");
		int input2 = getArraySize("Enter the number of data to be key in
to array 2 : ");
		double[] array2 = getArrayInput(input2, "Enter the input into
array 2: ");

		double median = calculation(array1, array2);
		System.out.printf("The median is: %.3f%n", median);
	}

	private static int getArraySize(String prompt) {
		Scanner sc = new Scanner(System.in);
		int size = 0;
```

```
            boolean validInput = false;

            while (!validInput) {
                System.out.print(prompt);
                if (sc.hasNextInt()) {
                    size = sc.nextInt();
                    sc.nextLine(); // Consume the newline character
                    if (size >= 1 && size <= Integer.MAX_VALUE) {
                        validInput = true;
                    } else {
                        System.out.println("Please enter a valid
input.");//negative int
                    }
                } else {
                    System.out.println("Please enter an integer.");
                    sc.nextLine(); // Consume the invalid input
                }
            }

            return size;
        }

        private static double[] getArrayInput(int size, String prompt) {
            Scanner sc = new Scanner(System.in);
            double[] array = new double[size];

            for (int i = 0; i < size; i++) {
                boolean validInput = false;

                while (!validInput) {
                    System.out.print(prompt);
                    String input = sc.nextLine();

                    if (!input.isEmpty()) {
                        try {
                            array[i] = Double.parseDouble(input);
                            validInput = true;
                        } catch (NumberFormatException e) {
```

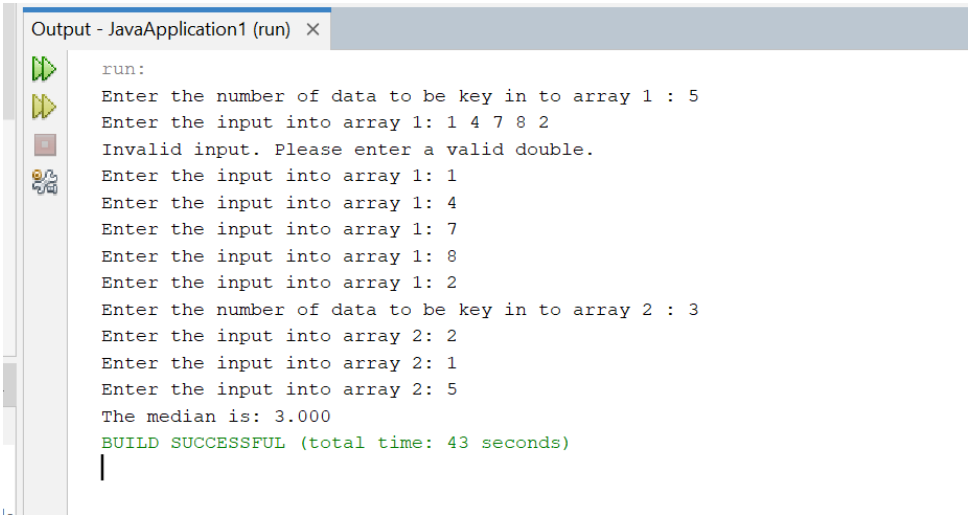
```
                        System.out.println("Invalid input. Please enter a
valid double.");
                    }
                } else {
                    System.out.println("Please do not leave it blank");
                }
            }
        }

        return array;
    }

    private static double calculation(double[] array1, double[] array2) {
        double[] FArray = new double[array1.length + array2.length];
        System.arraycopy(array1, 0, FArray, 0, array1.length);
        System.arraycopy(array2, 0, FArray, array1.length, array2.length);
        for (int j = 0; j < FArray.length - 1; j++) {
            for (int i = 0; i < FArray.length - j - 1; i++) {
                if (FArray[i] > FArray[i + 1]) {
                    double temp = FArray[i];
                    FArray[i] = FArray[i + 1];
                    FArray[i + 1] = temp;
                }
            }
        }
        int num = FArray.length;
        double median;
        if (num % 2 == 0) {
            median = (FArray[(num / 2) - 1] + FArray[(num / 2)]) / 2;
        } else {
            median = FArray[(num-1)/2];
        }
        return median;
    }
}
```

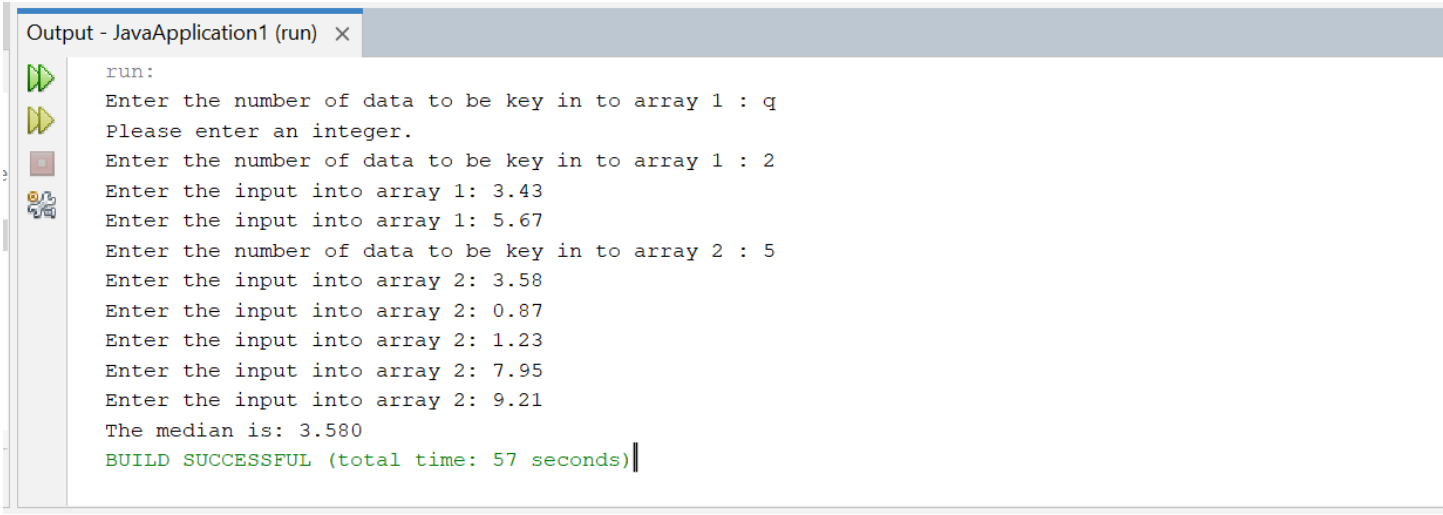
The method getArraySize(String prompt)
is used to read the array1.length and array2.length, if the user enters a non-integer .
System will prompt the user until get the positive integer.

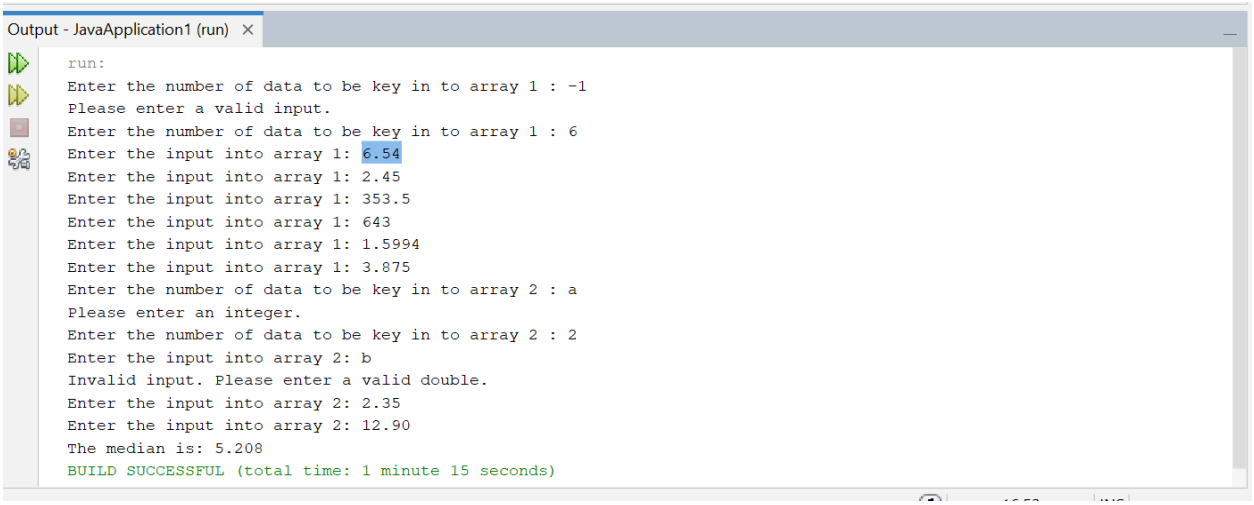
The method getArrayInput((int size, String prompt)
is used to input all elements which is size in array1 and array2 , if the user enters a non-double number. System will prompt the user until get the double number.

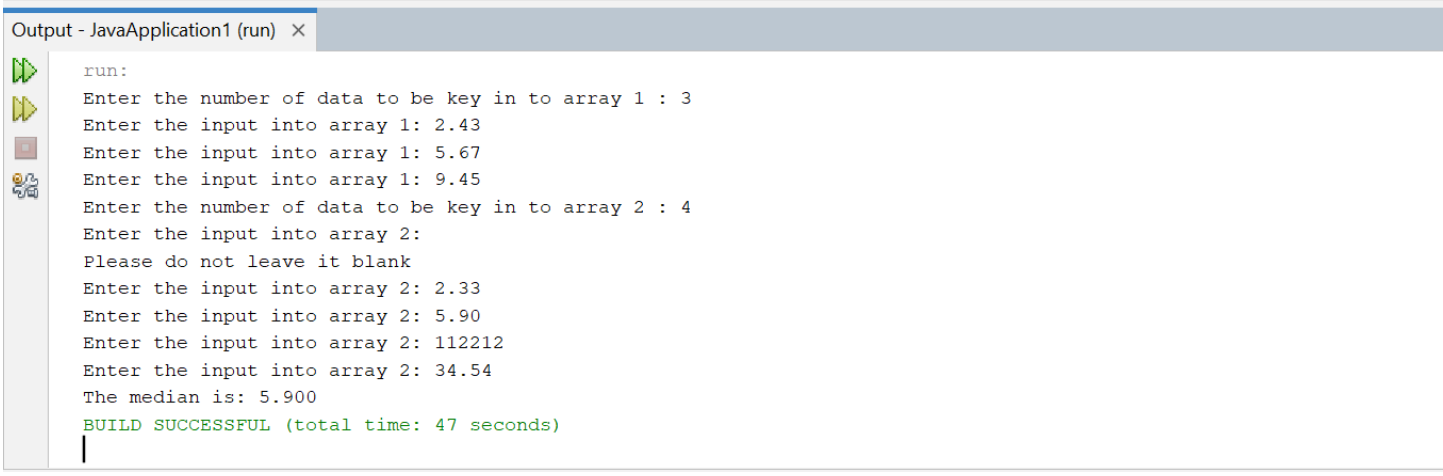
The median is caclulate by using method calculation(double[] array1, double[] array2) This method uses arraycopy to copy the array1 and array2 to FArray
Then we use bubble sort to sort the FArray elements in ascending order.To find the median we use the sum of {n/2} term and {n/2 +1} term for even number and {(n+1)/2} term for odd number
In Java we need to -1 to both as the index start with 0.

so the method gets and return median.
In main method it will call the method and display the median in 3 decimal.




Output - JavaApplication1 (run)
run:
Enter the number of data to be key in to array 1 : q
Please enter an integer.
Enter the number of data to be key in to array 1 : 2
Enter the input into array 1: 3.43
Enter the input into array 1: 5.67
Enter the number of data to be key in to array 2 : 5
Enter the input into array 2: 3.58
Enter the input into array 2: 0.87
Enter the input into array 2: 1.23
Enter the input into array 2: 7.95
Enter the input into array 2: 9.21
The median is: 3.580
BUILD SUCCESSFUL (total time: 57 seconds)


Output - JavaApplication1 (run)
run:
Enter the number of data to be key in to array 1 : -1
Please enter a valid input.
Enter the number of data to be key in to array 1 : 6
Enter the input into array 1: 6.54
Enter the input into array 1: 2.45
Enter the input into array 1: 353.5
Enter the input into array 1: 643
Enter the input into array 1: 1.5994
Enter the input into array 1: 3.875
Enter the number of data to be key in to array 2 : a
Please enter an integer.
Enter the number of data to be key in to array 2 : 2
Enter the input into array 2: b
Invalid input. Please enter a valid double.
Enter the input into array 2: 2.35
Enter the input into array 2: 12.90
The median is: 5.208
BUILD SUCCESSFUL (total time: 1 minute 15 seconds)


Output - JavaApplication1 (run)
run:
Enter the number of data to be key in to array 1 : 3
Enter the input into array 1: 2.43
Enter the input into array 1: 5.67
Enter the input into array 1: 9.45
Enter the number of data to be key in to array 2 : 4
Enter the input into array 2:
Please do not leave it blank
Enter the input into array 2: 2.33
Enter the input into array 2: 5.90
Enter the input into array 2: 112212
Enter the input into array 2: 34.54
The median is: 5.900
BUILD SUCCESSFUL (total time: 47 seconds)

Question 3

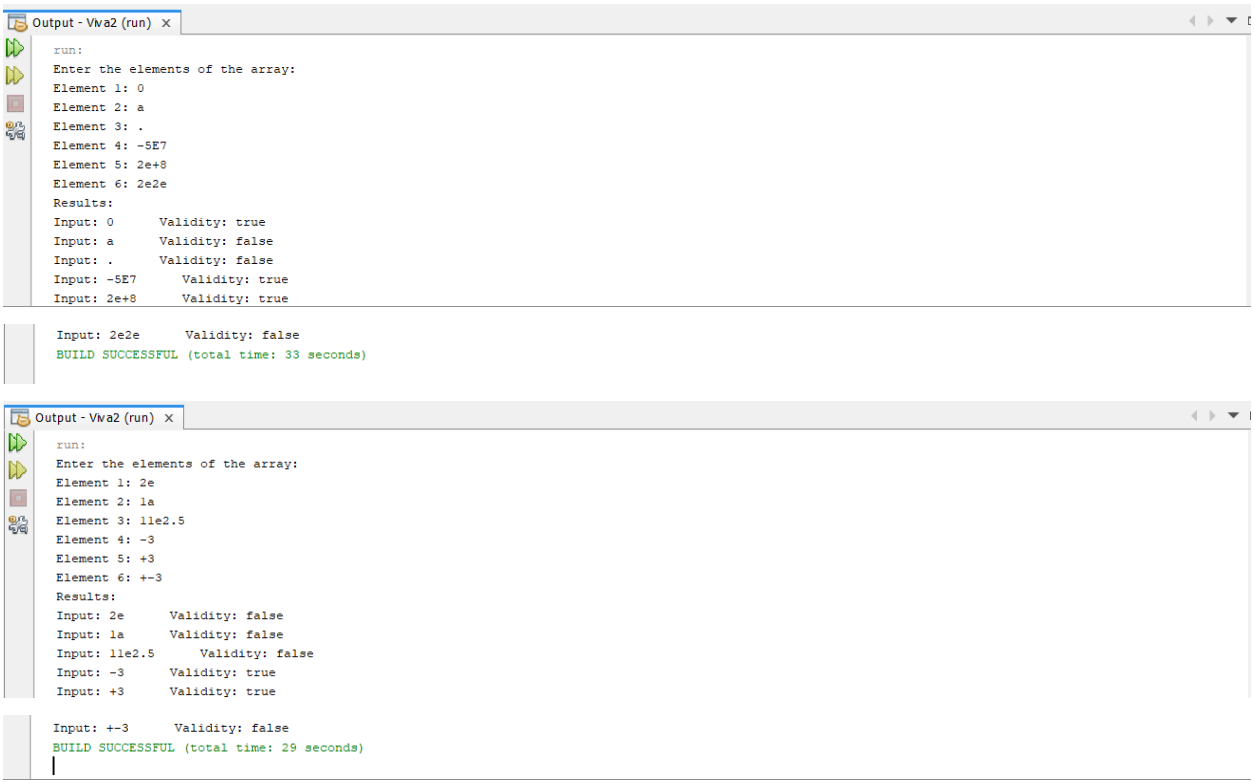
Problem description:

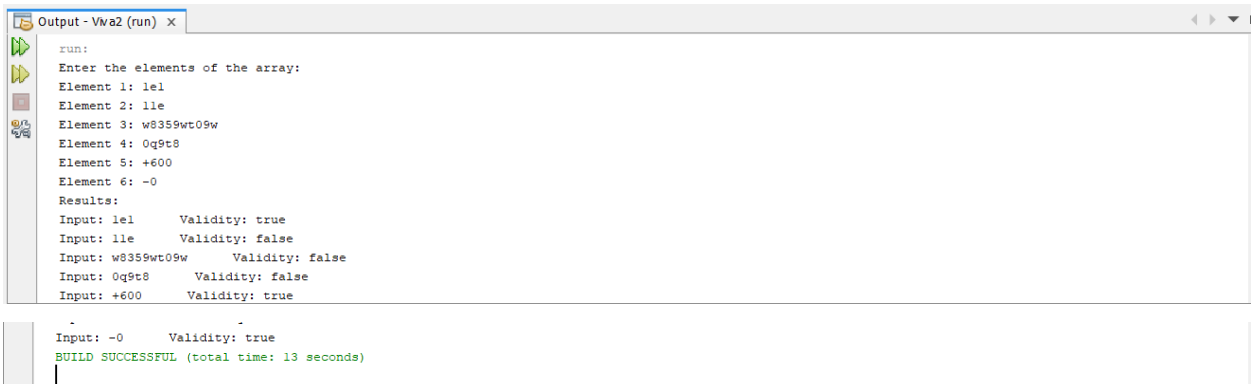
To check the validity of the numerical numbers using a Java method that accepts a String array and returns a Boolean array.

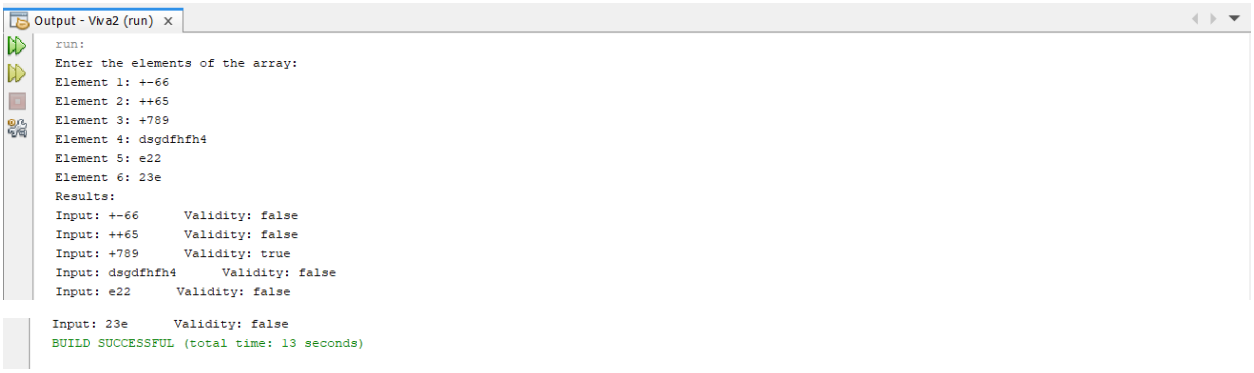
Solution:

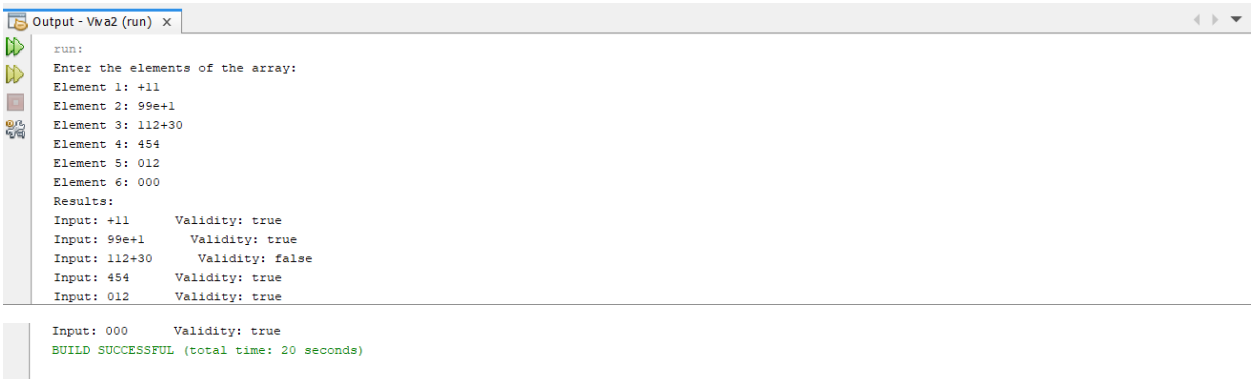
We use a scanner and an array to accept the input as a String. Next, we use a method to check the numerical validity of each element in the array. To check the numerical validity of each element in the array, we use another method that will try to convert the String into a Double. If the conversion is successful, it is returned as true, whereas if the conversion is not successful, it is returned as false. After checking the validity of each element in the array, we print out the results.

Sample Input & Output:




Output - Viva2 (run)
run:
Enter the elements of the array:
Element 1: lel
Element 2: lle
Element 3: w8359wt09w
Element 4: 0q9t8
Element 5: +600
Element 6: -0
Results:
Input: lel Validity: true
Input: lle Validity: false
Input: w8359wt09w Validity: false
Input: 0q9t8 Validity: false
Input: +600 Validity: true
Input: -0 Validity: true
BUILD SUCCESSFUL (total time: 13 seconds)


Output - Viva2 (run)
run:
Enter the elements of the array:
Element 1: +-66
Element 2: ++65
Element 3: +789
Element 4: dsgdfhfh4
Element 5: e22
Element 6: 23e
Results:
Input: +-66 Validity: false
Input: ++65 Validity: false
Input: +789 Validity: true
Input: dsgdfhfh4 Validity: false
Input: e22 Validity: false
Input: 23e Validity: false
BUILD SUCCESSFUL (total time: 13 seconds)


Output - Viva2 (run)
run:
Enter the elements of the array:
Element 1: +11
Element 2: 99e+1
Element 3: 112+30
Element 4: 454
Element 5: 012
Element 6: 000
Results:
Input: +11 Validity: true
Input: 99e+1 Validity: true
Input: 112+30 Validity: false
Input: 454 Validity: true
Input: 012 Validity: true
Input: 000 Validity: true
BUILD SUCCESSFUL (total time: 20 seconds)

Source code:

```
        // Check the numerical validity of each element in the array
        boolean[]resultArr = checkNumValid(inputArr);

        // print the results
        printResult(inputArr, resultArr);
    }

    // Method to check if a given string is a valid numeric representation
    public static boolean isNumeric(String str) {
        try {
            Double.parseDouble(str);  // Used to convert a String representing a floating-point number into its double repre
            return true;  // If it is a numeric number (the conversion is successful), return true
        }
        catch (NumberFormatException e) {
            return false;  // If it is not a numerical number, return false
        }
    }

    // Method to check the numerical validity of each element in the array
    public static boolean[]checkNumValid(String[]inputArr) {
        boolean[]resultArr = new boolean[inputArr.length];
```

```
        for (int i = 0; i < inputArr.length; i++) {
            resultArr[i] = isNumeric(inputArr[i]);  // Using the isNumeric method to check if the element is a valid numeric
        }

        return resultArr;
    }

    // Method to print the results
    public static void printResult(String[] inputArr, boolean[]resultArr) {
        System.out.println("Results:");
        for (int i = 0; i < inputArr.length; i++) {
            System.out.println("Input: " + inputArr[i] + "      Validity: " + resultArr[i]);
        }
    }
}
```