

# ASSIGNMENT DAY-7

Name: Vishnu Priya

Enrolment Number: SU625MR009

Batch/Class: MERN stack

Assignment: Day 7

Date of Submission: 2nd July 2025

---

## Problem Statement:

**List Explorer:** Create an integer array of temperature. Find the sum, average and highest temperature among all by printing all values.

## Algorithm:

- Step 1:** Start the program.
- Step 2:** Read number of days  $n$  from user.
- Step 3:** Declare a float array  $ar$  of size  $n$ .
- Step 4:** Prompt user to input  $n$  temperature values and store them in  $ar$ .
- Step 5:** Display all temperature values using a loop.
- Step 6:** Initialize  $sum = 0$  and loop through array to calculate total sum.
- Step 7:** Compute average as  $avg = sum / n$ .
- Step 8:** Display the sum and average.
- Step 9:** Initialize  $max = ar[0]$  and loop to find the maximum temperature.
- Step 10:** Display the highest temperature.
- Step 11:** End the program

## Pseudo Code:

Start

Prompt user to enter number of days ( $n$ )

Create float array of size  $n$  to store temperatures

Prompt user to enter temperature values for  $n$  days

Loop through array and store input temperatures

Print all entered temperature values

Initialize  $sum$  to 0

Loop through array to calculate sum of temperatures

Calculate  $average = sum / n$

Print sum and average

Initialize max as first temperature

Loop through array to find maximum temperature

Print highest temperature

End

### Code:

```
package DSA;
import java.util.*;
public class Temperature {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the number of days: ");
        int n=sc.nextInt();
        float[] ar=new float[n];
        System.out.println("enter the temperature values of this days: ");
        for (int i=0;i<ar.length;i++) {
            ar[i]=sc.nextFloat();
        }
        System.out.println("the temperature of "+n+" days is: ");
        for (int i=0;i<ar.length;i++) {
            System.out.println(ar[i]);
        }
        float sum=0;
        float avg=0;
        for(int i=0;i<ar.length;i++) {
            sum=sum+ar[i];
        }
        System.out.println("The sum of the temperature is "+sum);
        avg=sum/n;
        System.out.println("average is "+avg);
        float max=ar[0];
        for(int i=0;i<ar.length;i++) {
            if(ar[i]>max) {
                max=ar[i];
            }
        }
        System.out.println("The highest temperature is "+max);
    }
}
```

### Test Case:

Test case	Input	Expected output	Actual output	Status
TC1	22.7 23.8 20.5 23.3 24.3	Sum=114.60 Avg=22.92 Highest =24.3	Sum=114.60 Avg=22.92 Highest =24.3	Pass

TC2	23.5 24.6 22.3	Sum=70.39 Avg=23.46 Highest =24.6	Sum=70.39 Avg=23.46 Highest =24.6	Pass
TC3	27.3 26.4 28	Sum=81.7 Avg=27.23 Highest =28.0	Sum=81.7 Avg=27.23 Highest =28.0	pass

## Output:

TC1:

```

enter the number of days:
5
enter the temperature values of this days:
22.7
23.8
20.5
23.3
24.3
the temperature of 5 days is:
22.7
23.8
20.5
23.3
24.3
The sum of the temperature is 114.600006
average is 22.920002
The highest temperature is 24.3

```

TC2:

```

enter the number of days:
3
enter the temperature values of this days:
23.5
24.6
22.3
the temperature of 3 days is:
23.5
24.6
22.3
The sum of the temperature is 70.399994
average is 23.466665
The highest temperature is 24.6

```

TC3:

```

enter the number of days:
3
enter the temperature values of this days:
27.3

```

```
26.4
28
the temperature of 3 days is:
27.3
26.4
28.0
The sum of the temperature is 81.7
average is 27.233332
The highest temperature is 28.0
```

## Problem Statement:

**Product of evens :** Create a integer array 1-10. Print the product of the all even numbers.

## Algorithm:

**Step 1:**Start the program.

**Step 2:**Declare and initialize array a with values {1,2,3,4,5,6,7,8,9,10}.

**Step 3:**Initialize res = 1 to store the product of even numbers.

**Step 4:**Loop through each element of array using index i.

**Step 5:**If  $a[i] \% 2 == 0$ , multiply res by  $a[i]$ .

**Step 6:**After loop ends, print the value of res as the product of even numbers.

**Step 7:**End the program.

## Pseudo Code:

Start

Initialize array with numbers from 1 to 10

Set result (res) to 1

Loop through each element of the array

    If the element is even, multiply it with result

After loop, display the final product

End

## Code:

```
package DSA;
import java.util.*;
public class EvenNumber {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int[] a= {1,2,3,4,5,6,7,8,9,10};
        int res=1;
        for(int i=0;i<a.length;i++) {
            if(a[i]%2==0) {
                res*=a[i];
            }
        }
    }
}
```

```
        System.out.println("The product of even numbers is "+res);  
    }  
}
```

### Output:

The product of even numbers is 3840

### Problem Statement:

**Reverse MyList:** Create a string array of items. Print the string in reverse order.

### Algorithm:

**Step 1:**Start

**Step 2:**Read a string s from the user.

**Step 3:**Initialize a character array c of size equal to the length of s.

**Step 4:**Copy each character of s into the character array c.

**Step 5:**Print the characters of the array c in original order.

**Step 6:**Print the characters of the array c in reverse order.

**Step 7:**End

## Pseudo Code:

START

PROMPT "Enter the String:"

READ s

SET length  $\leftarrow$  length of s

DECLARE char array c[length]

FOR i  $\leftarrow$  0 to length - 1 DO

    c[i]  $\leftarrow$  s.charAt(i)

END FOR

PRINT "The original array:"

FOR i  $\leftarrow$  0 to length - 1 DO

    PRINT c[i]

END FOR

PRINT "The reverse array:"

FOR i  $\leftarrow$  length - 1 DOWNT0 0 DO

    PRINT c[i]

END FOR

END

## Code:

```
package DSA;
import java.util.*;

public class ReverseList {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the String: ");
        String s = sc.nextLine();

        char[] c = new char[s.length()];
        for (int i = 0; i < c.length; i++) {
            c[i] = s.charAt(i);
        }
    }
}
```

```

System.out.println("The original array: ");
for (int i = 0; i < c.length; i++) {
    System.out.print(c[i]);
}

System.out.println("\nThe reverse array: ");
for (int i = c.length - 1; i >= 0; i--) {
    System.out.print(c[i]);
}
}
}

```

## Output:

```

Enter the String:
program
The original array:
program
The reverse array:
margorp

```

## Problem Statement:

**Word Search:** With an input string as key from the user. Search for the key word

## Algorithm:

- Step 1:**Start
- Step 2:**Initialize a list/array of words: ["all", "is", "well"]
- Step 3:**Prompt the user to enter a word to search
- Step 4:**Read the input word as key
- Step 5:**Set a flag variable a to false
- Step 6:**Loop through each word in the array:
- Step 7:**If the current word equals the key (ignoring case), set a to true
- Step 8:**After the loop:
- Step 9:**If a is true, print "word found!..."
- Step 10:**    Else, print "word not found!..."
- Step 11:**    End

## Pseudo Code:

START

SET word[] ← ["all", "is", "well"]

PROMPT "Enter the word:"

READ key

SET a ← false

```

FOR i ← 0 to length of word[] - 1 DO
    IF word[i] equalsIgnoreCase key THEN
        SET a ← true
    END IF
END FOR
IF a is true THEN
    PRINT "word found!..."
ELSE
    PRINT "word not found!..."
END IF
END

```

## Code:

```

package DSA;
import java.util.*;
public class WordSearch {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        String[] word= {"all","is","well"};
        System.out.println("Enter the word: ");
        String key=sc.next();
        boolean a=false;
        for(int i=0;i<word.length;i++) {
            if(word[i].equalsIgnoreCase(key)) {
                a=true;
            }
        }
        if(a==true) {
            System.out.println("word found!...");
        }else {
            System.out.println("word not found!...");
        }
    }

}

```

## Output:

TC1:

```

Enter the word:
all
word found!...

```



TC2:

Enter the word:

the

word not found!...

## Program Statement:

**Implement GCD:** Using Euclidean Algorithm, write a program that calculates the GCD of the given numbers.

## Algorithm:

**Step 1:**Start

**Step 2:**Read two integers a and b from the user

**Step 3:**While b is not equal to 0, do:

**Step 4:**Store the value of b in a temporary variable temp

**Step 5:**Set  $b = a \% b$

**Step 6:**Set  $a = \text{temp}$

**Step 7:**When the loop ends, a holds the GCD

**Step 8:**Print the result

**Step 9:**End

## Pseudo Code:

START

PROMPT "Enter the numbers:"

READ a, b

WHILE  $b \neq 0$  DO

temp  $\leftarrow$  b

$b \leftarrow a \text{ MOD } b$

$a \leftarrow \text{temp}$

END WHILE

PRINT "The GCD of original a and b is", a

END

## Code:

```
package Maths;
import java.util.*;
public class GCD {
    public static int calculateGCD(int a,int b) {
        while(b!=0) {
            int temp=b;
            b=a%b;
            a=temp;
        }
    }
}
```

```

        return a;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number for ");
        int a=sc.nextInt();
        int b=sc.nextInt();
        int res=calculateGCD(a,b);
        System.out.println("The GCD of "+a+" and "+b+" is "+res);
    }
}

```

## Output:

```

Enter the number for
64
7
The GCD of 64 and 7 is 1

```

## Problem Statement:

**Implement LCM:** Write a program to calculate LCM and GCD

## Algorithm:

**Step 1:**Start

**Step 2:**Read two integers a and b from the user

**Step 3:**Call calculateGCD(a, b) to compute GCD:

**Step 4:**While  $b \neq 0$

- a. Store b in temp
- b. Set  $b = a \% b$
- c. Set  $a = \text{temp}$

**Step 5:**Return a as the GCD

**Step 6:**Use the formula  $\text{LCM} = (\text{original } a \times \text{original } b) / \text{GCD}$

**Step 7:**Print the GCD and LCM

**Step 8:**End

## Pseudo code:

START

PROMPT "Enter the numbers:"

READ a, b

FUNCTION calculateGCD(x, y)

    WHILE  $y \neq 0$  DO

```

temp ← y
y ← x MOD y
x ← temp
END WHILE

RETURN x

END FUNCTION

SET gcd ← calculateGCD(a, b)
SET lcm ← (a × b) / gcd

PRINT "The GCD of a and b is", gcd

PRINT "The LCM of a and b is", lcm

END

```

## Code:

```

package Maths;
import java.util.*;
public class LCM {
    public static int calculateGCD(int a,int b) {
        while(b!=0) {
            int temp=b;
            b=a%b;
            a=temp;
        }
        return a;
    }
    public static int calculateLCM(int a,int b) {
        return (a*b)/calculateGCD(a,b);
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number for ");
        int a=sc.nextInt();
        int b=sc.nextInt();
        int res=calculateGCD(a,b);
        System.out.println("The GCD of "+a+" and "+b+" is "+res);
        int lcm=calculateLCM(a,b);
        System.out.println("The LCM of "+a+" and "+b+" is "+lcm);
    }
}

```

## Output:

```
Enter the number for
```

```
67
```

The GCD of 67 and 4 is 1  
The LCM of 67 and 4 is 268

## Problem Statement:

**Simple Sum Calculator:** Create HTML page that perform simple sum operation.

## Algorithm:

**Step 1:** Start

**Step 2:** Display two input fields for entering numbers (num1 and num2)

**Step 3:** Display a button labeled "**Calculate Sum**"

**Step 4:** When the form is submitted:

**Step 5:** Prevent the default form submission behavior

**Step 6:** Retrieve the values from the two input fields

**Step 7:** Convert them to floating-point numbers

**Step 8:** Calculate the sum of the two numbers

**Step 9:** Display the result below the form

**Step 10:** End

## Pseudo Code:

START

DISPLAY form with:

- Input field for first number (num1)
- Input field for second number (num2)
- Submit button labeled "Calculate Sum"
- Area to display result (e.g., a heading or paragraph tag)

ON form submission DO

PREVENT default form submit behavior

READ num1 from input field

READ num2 from input field

CONVERT num1 and num2 to float numbers

COMPUTE sum = num1 + num2

DISPLAY "Sum: " + sum in result area

END

END

## Code:

```
<!DOCTYPE html>
<html>
```

```

<head>
  <title>Simple Sum Calculator</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 50px;
    }
    input, button {
      padding: 8px;
      margin: 10px;
      font-size: 14px;
    }
  </style>
</head>
<body>
  <h2>Simple Sum Calculator</h2>
  <form id="calculatorSum">
    <input type="number" id="num1" placeholder="Enter first number" required>
    <input type="number" id="num2" placeholder="Enter second number" required>
    <button type="submit">Calculate Sum</button>
  </form>
  <h3 id="result"></h3>
  <script>
    document.getElementById("calculatorSum").addEventListener("submit", function(event) {
      event.preventDefault();
      let num1 = parseFloat(document.getElementById("num1").value);
      let num2 = parseFloat(document.getElementById("num2").value);
      let sum = num1 + num2;
      document.getElementById("result").textContent = "Sum: " + sum;
    });
  </script>
</body>
</html>

```

Output:

## Simple Sum Calculator

56	52	Calculate Sum
----	----	---------------

**Sum: 108**

## Problem Statement:

**Web based GCD/LCM calculator:** With Extend the form to include two buttons, perform GCD and LCM calculation

## Algorithm:

- Step 1:** Start
- Step 2:** Prompt the user to enter the number of days n
- Step 3:** Create a float array ar of size n
- Step 4:** Prompt the user to enter n temperature values
- Step 5:** Store the values in the array
- Step 6:** Print all entered temperature values
- Step 7:** Initialize sum = 0
- Step 8:** Loop through the array to: Add each temperature to sum
- Step 9:** Calculate  $\text{avg} = \text{sum} / n$
- Step 10:** Print sum and avg
- Step 11:** Initialize max = ar[0]
- Step 12:** Loop through the array to:
- Step 13:** If current value > max, update max
- Step 14:** Print the highest temperature
- Step 15:** End

## Pseudo Code:

START

PROMPT "Enter the number of days:"

READ n

DECLARE float array ar[n]

PROMPT "Enter the temperature values for these days:"

FOR i ← 0 TO n - 1 DO

    READ ar[i]

END FOR

PRINT "The temperature of n days is:"

FOR i ← 0 TO n - 1 DO

    PRINT ar[i]

END FOR

SET sum ← 0

FOR i ← 0 TO n - 1 DO

    sum ← sum + ar[i]

END FOR

avg ← sum / n

PRINT "The sum of the temperature is", sum

PRINT "Average is", avg

SET max ← ar[0]

```

FOR i ← 0 TO n - 1 DO
    IF ar[i] > max THEN
        max ← ar[i]
    END IF
END FOR

PRINT "The highest temperature is", max

END

```

## Code:

```

package DSA;
import java.util.*;
public class Temperature {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the number of days: ");
        int n=sc.nextInt();
        float[] ar=new float[n];
        System.out.println("enter the temperature values of this days: ");
        for (int i=0;i<ar.length;i++) {
            ar[i]=sc.nextFloat();
        }
        System.out.println("the temperature of "+n+" days is: ");
        for (int i=0;i<ar.length;i++) {
            System.out.println(ar[i]);
        }
        float sum=0;
        float avg=0;
        for(int i=0;i<ar.length;i++) {
            sum=sum+ar[i];
        }
        System.out.println("The sum of the temperature is "+sum);
        avg=sum/n;
        System.out.println("average is "+avg);
        float max=ar[0];
        for(int i=0;i<ar.length;i++) {
            if(ar[i]>max) {
                max=ar[i];
            }
        }
        System.out.println("The highest temperature is "+max);
    }
}

```

Output:

## GCD and LCM Calculator

**LCM: 90**

**GCD: 3**