Laboratory Assignment File

for

Data Structure and Algorithms

Master of Technology

in

Computer Science & Engineering

Submitted by

Yugal

(Roll no. 25903053)

Department of Computer Science & Engineering
Dr. B R Ambedkar National Institute of Technology Jalandhar
Punjab, India-144008
May, 2026

Contents

List of Figures

Assignment 1: Basic Programs

1.1 Hello World

```java
public class Hello{
    public static void main(String arg[]){
    System.out.println("Hello");
    }
}
```

1.2 Write a program to show Array

```java
// Store and display the marks
public class Array{
    public static void main(String... args){
        int[] marks = {70,75,80,90,95};
        for(int i=0;i<marks.length;i++){
            // System.out.println("Student Marks: "+marks[i]);
            System.out.println("Student " + (i+1) +":"+marks[i]);
        }
    }

}
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/bin/env /usr/lib/jvm/java-11-open
jdk-amd64/bin/java -cp /home/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979c97b038c3
955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin Array
Student 1:70
Student 2:75
Student 3:80
Student 4:90
Student 5:95
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 1: Program to show Array

1.3 Write a program to show sum and average

```java
// Sum and average of array elements
class Sum_Average {

    public static void main(String[] args){
        int[] a = {20,30,40,50,60};
        int sum = 0;
```

```
7
8           for(int x:a){
9               sum += x;
10          }
11          double average = (double) sum/a.length;
12
13          System.out.println("Sum: "+ sum);
14          System.out.println("Average: "+ average);
15      }
16  }
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/bin/env /usr/lib/jvm/java-11-open
jdk-amd64/bin/java -cp /home/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979c97b038c3
955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin Sum_Average
Sum: 200
Average: 40.0
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 2: Program to show sum and average

Assignment 2: Sorting Algorithms

2.1 Write a program to show Bubble Sort

```java
// Sorting of array in Bubble Sort
public class BubbleSort{
    public static void main(String[] args){
        int arr[] = {5,1,4,2,8};
        for(int i=0; i<arr.length-1; i++){
            for(int j=0; j<arr.length-i-1; j++){
                if(arr[j]>arr[j+1]){
                    int temp = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = temp;
                }
            }
        }
        for(int num : arr){
            System.out.print(num + " ");
        }
    }
}
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/bin/env /usr/lib/jvm/java-11-open
jdk-amd64/bin/java -cp /home/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979c97b038c3
955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin BubbleSort
1 2 4 5 8 (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 3: Program to show Bubble Sort

2.2 Write a program to show Insertion Sort

```java
// Insertion Sort
public class InsertionSort {
    public static void main(String[] args) {
        int arr[] = {1,4,5,2,3,7,9,8,0,6};
        for(int i = 1;i<arr.length; i++){
            int key = arr[i];
            int j = i-1;
            while (j>=0 && arr[j]>key) {
                arr[j+1] = arr[j];
                j--;
            }
```

```
            arr[j+1] = key;
        }
        for(int num : arr){
            System.out.print(num + " ");
        }
    }
}
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/bin/env /usr/lib/jvm/java-11-open
jdk-amd64/bin/java -cp /home/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979c97b038c3
955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin InsertionSort
0 1 2 3 4 5 6 7 8 9 (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 4: Program to show Insertion Sort

2.3 Write a program to show Selection Sort

```
    //Selection Sort
public class SelectionSort {
    public static void main(String[] args) {
        int arr[] = {3,5,2,6,-1,0,4,7,9,8,-2};

        for(int i=0;i<arr.length-1;i++){
            int minIndex = i;
            for(int j = i+1; j<arr.length;j++){
                if(arr[j]<arr[minIndex]){
                    minIndex = j;

                }
            }
            int temp = arr[minIndex];
            arr[minIndex] = arr[i];
            arr[i] = temp;
        }
        for(int num : arr){
            System.out.print(num + " ");

        }
    }

}
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/bin/env /usr/lib/jvm/java-11-open
jdk-amd64/bin/java -cp /home/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979c97b038c3
955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin SelectionSort
-2 -1 0 2 3 4 5 6 7 8 9 (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 5: Program to show Selection Sort

2.4 Write a program to show Merge Sort

```java
    //Merge Sort : Divide and Conquer
import java.util.Arrays;

public class MergeSort {

    // Main merge sort function
    public static void mergeSort(int[] arr, int left, int right){
        if(left < right){
            int mid =  left + (right-left)/2;
            // Sort Left Half
            mergeSort(arr, left, mid);
            // Sort Right Half
            mergeSort(arr, mid+1, right);
            // Merge Both Halves
            merge(arr, left, mid, right);
        }
    }

    // Merge two sorted subarrays
    public static void merge(int[] arr, int left, int mid, int right){
        int n1 = mid-left+1;
        int n2 = right-mid;
        int[] l = new int[n1];
        int[] r = new int[n2];

        // Copy data to temp arrays
        for(int i=0; i<n1; i++)
            l[i] = arr[left+i];
        for(int j=0; j<n2; j++)
            r[j] = arr[mid+1+j];

        int i = 0, j=0, k=left;

        // Merge temp arrays back into array/arr
        while (i<n1 && j<n2) {
```

```
            if (l[i]<=r[j]) {
                arr[k++] = l[i++];
            }
            else{
                arr[k++] = r[j++];
            }
        }

        // Copy remaining elements
        while (i<n1) {
            arr[k++] = l[i++];
        }
        while (j<n2) {
            arr[k++] = r[j++];
        }
    }

    // Driver Code
    public static void main(String[] args) {

        int[] arr = {38,27,43,3,9,82,10};
        System.out.println("Before Sorting: ");
        System.out.println(Arrays.toString(arr));

        mergeSort(arr, 0, arr.length-1);

        System.out.println("After Sorting: ");
        System.out.println(Arrays.toString(arr));

    }
}
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  cd /home/yugal/Desktop/Sessions/DSA ;
/usr/bin/env /usr/lib/jvm/java-11-openjdk-amd64/bin/java -cp /home/yugal/.config/Code/User/works
paceStorage/13eba2f5617b89979c97b038c3955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin MergeSort
Before Sorting:
[38, 27, 43, 3, 9, 82, 10]
After Sorting:
[3, 9, 10, 27, 38, 43, 82]
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 6: Program to show Merge Sort

2.5 Write a program to show Quick Sort

```java
    // Quick Sort : small -- pivot -- larger
import java.util.Arrays;

public class QuickSort {

    // Main quick sort function
    public static void quickSort(int[] arr, int low, int high){
        if (low<high) {
            int pivotIndex = partition(arr, low, high);

            // Sort elements before and after partition
            quickSort(arr, low, pivotIndex-1);
            quickSort(arr, pivotIndex+1, high);
        }
    }

    // Partition Function
    public static int partition(int[] arr, int low, int high){
        int pivot = arr[high]; // Choose last element as pivot
        int i = low-1;
        for(int j=low; j<high; j++){
            if (arr[j]<pivot) {
                i++;

                // Swap arr[i] and arr[j]
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }

        // Place pivot at correct position
        int temp = arr[i+1];
        arr[i+1] = arr[high];
        arr[high] = temp;

        return i+1;
    }

    // Driver code
    public static void main(String[] args) {

        int[] arr = {34,2,56,4,89,65,1,44};
```

```
44          System.out.println("Before Sorting: ");
45          System.out.println(Arrays.toString(arr));
46
47          quickSort(arr, 0, arr.length-1);
48          System.out.println("After Sorting: ");
49          System.out.println(Arrays.toString(arr));
50
51      }
52  }
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  cd /home/yugal/Desktop/Sessions/DSA ;
/usr/bin/env /usr/lib/jvm/java-11-openjdk-amd64/bin/java -cp /home/yugal/.config/Code/User/works
paceStorage/13eba2f5617b89979c97b038c3955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin QuickSort
Before Sorting:
[34, 2, 56, 4, 89, 65, 1, 44]
After Sorting:
[1, 2, 4, 34, 44, 56, 65, 89]
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 7: Program to show Quick Sort

Assignment 3: String

3.1 Write a program to Count Vowels in a String

```java
package Strings;
// Count vowels name in String
public class CountVowels {
    public static void main(String[] args) {
        String str = "Eleven";
        int count = 0;
        for(int i=0; i<str.length();i++){
            char ch = str.charAt(i);
            if("AEIOUaeiou".indexOf(ch) != -1){
                count++;
            }
        }
        System.out.println("Vowels: "+ count);
    }
}
```

Output

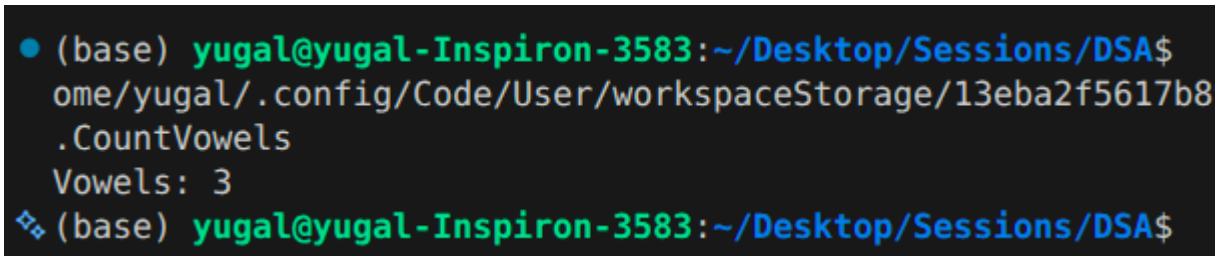

Figure 8: Output of Count Vowels Program

3.2 Palindrome String

```java
package Strings;
// Word is spelled in same way forward or backward. Eg: NITIN, MADAM
//Paldindrome
public class Palindrome {
    public static void main(String[] args) {
        String str = "nitin";
        String rev = "";

        for(int i=str.length()-1; i>=0;i--){
            rev += str.charAt(i);
        }
```

```
12
13          if (str.equals(rev)) {
14              System.out.println(str + " is a palindrome.");
15          }else{
16              System.out.println(str + " is not a palindrome.");
17          }
18      }
19  }
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
ome/yugal/.config/Code/User/workspaceStorage/13eba2f5617b
.Palindrome
nitin is a palindrome.
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 9: Output of Palindrome Program

3.3 Remove Spaces from a String

```
1  package Strings;
2  // Remove spaces from string
3  public class RemoveSpaces {
4      public static void main(String[] args) {
5          String str = "Every Expert Was Once A Beginner.";
6          System.out.println(str.replace(" ", ""));
7      }
8  }
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  c
/java-11-openjdk-amd64/bin/java -cp /home/yugal/.config/Cod
t.java/jdt_ws/DSA_1e4e73cd/bin Strings.RemoveSpaces
EveryExpertWasOnceABeginner.
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 10: Output of Remove Spaces Program

3.4 Reverse a String

```java
package Strings;

public class ReverseString {

    public static void main(String[] args) {

        String str = "Hello";
        String rev = "";

        for(int i = str.length()-1; i>=0; i--){
            rev = rev + str.charAt(i);
        }
        System.out.println("Reversed String: "+ rev);

    }
}
```

Output

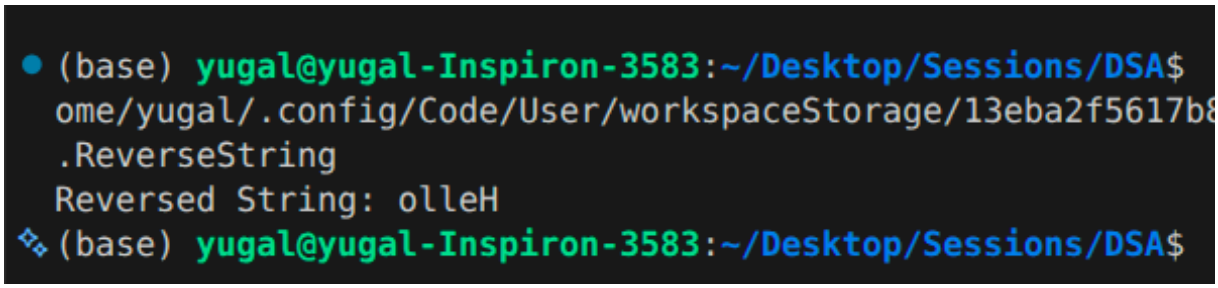

Figure 11: Output of Reverse String Program

Assignment 4: Iteration

4.1 Factorial using Iteration

```java
package Iteration;
// Factorial program using iteration
public class FactorialUsingIteration {
    // Driver code
    public static void main(String[] args) {
        int n = 5;
        int fact = 1;
        for(int i=1; i<=n; i++){
            fact *= i;
        }
        System.out.println("Factorial : "+fact);
    }
}
```

Output

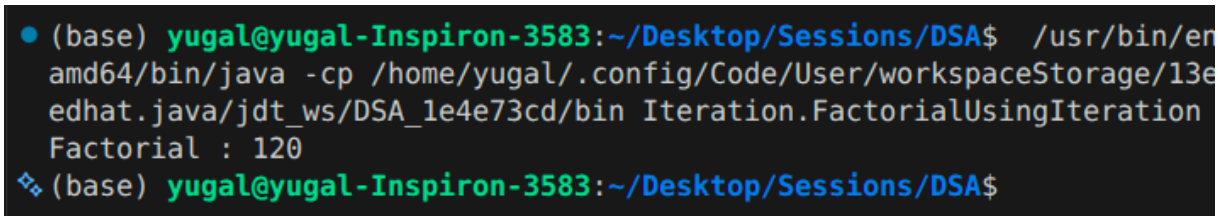

Figure 12: Output of Factorial using Iteration Program

Assignment 5: Searching

5.1 Binary Search

```java
// Binary Search
package Searching;
import java.util.Scanner;

public class BinarySearch {

    //Driver Code
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[20];// OK
        int n = 0, choice, key;

        do{
            System.out.println("\n ---Binary Search---");
            System.out.println("1. Insert Elements");
            System.out.println("2. Display Elements");
            System.out.println("3. Binary Search");
            System.out.println("4. Exit");

            System.out.print("Enter your choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter number of elements: ");
                    n = sc.nextInt();
                    System.out.println("Enter Elements: ");

                    for(int i=0; i<n; i++){
                        arr[i] = sc.nextInt();
                    }
                    break;

                case 2:
                    System.out.print("Array Elements: ");
                    for(int i=0; i<n; i++){
                        System.out.print(arr[i] + " ");
                    }
                    System.out.println();
                    break;
```

```
41
42                  case 3:
43                      System.out.print("Enter element to search: ");
44                      key = sc.nextInt();
45                      int low = 0, high = n-1;
46                      boolean found = false;
47                      while (low <= high) {
48                          int mid = (low+high) / 2;
49                          if (arr[mid] == key) {
50                              System.out.println("Element found at index:
                                    " + mid);
51                              found = true;
52                              break;
53                          }
54                          else if (arr[mid] < key) {
55                              low = mid+1;
56                          }
57                          else{
58                              high = mid-1;
59                          }
60                      }
61
62                      if (!found)
63                          System.out.println("Element not found");
64                      break;
65
66                  case 4:
67                      System.out.println("Exiting Program");
68                      break;
69
70                  default:
71                      System.out.println("Invalid Choice"); // break not
                            compulsory
72              }
73          }
74          while (choice != 4);
75          sc.close();
76      }
77  }
```

Output

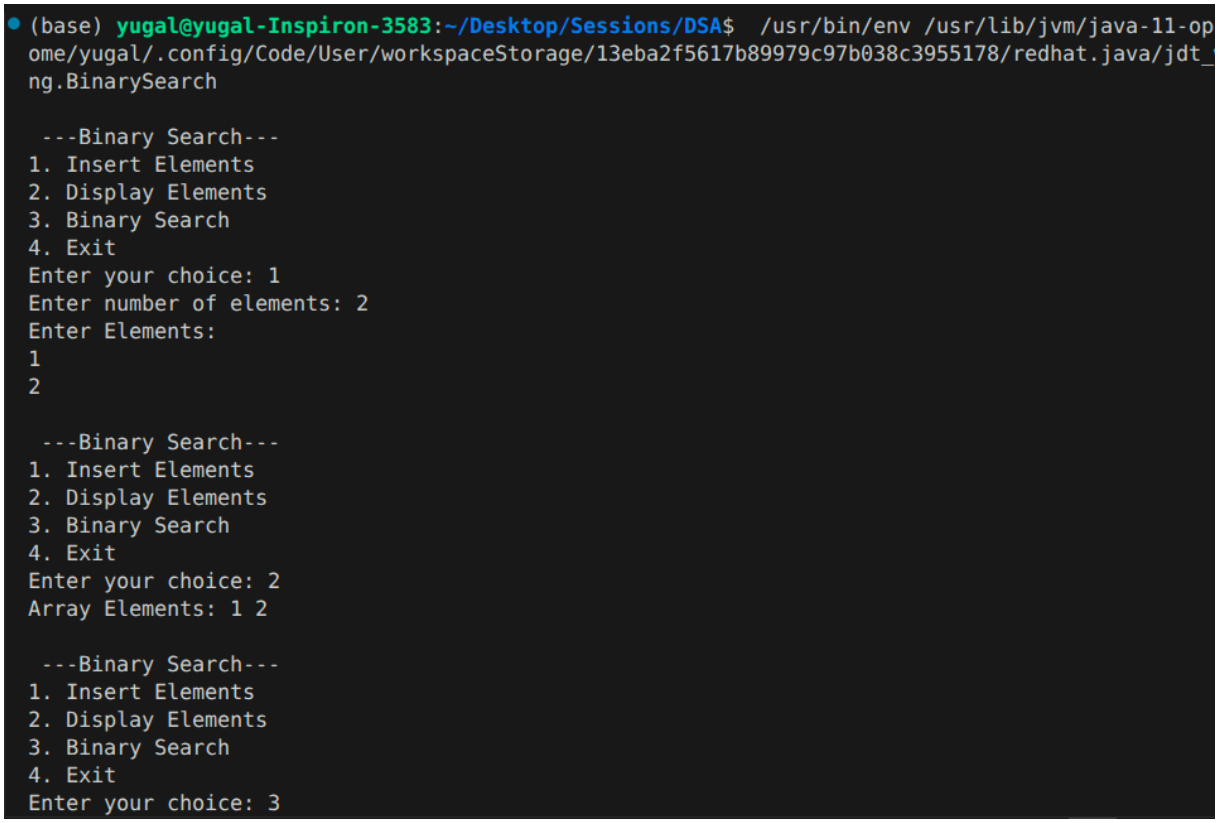

Figure 13: Output of Binary Search Program

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/b
amd64/bin/java -cp /home/yugal/.config/Code/User/workspaceStorag
edhat.java/jdt_ws/DSA_1e4e73cd/bin Searching.BinarySearch

 ---Binary Search---
1. Insert Elements
2. Display Elements
3. Binary Search
4. Exit
Enter your choice: 2
Array Elements: 1 2

 ---Binary Search---
1. Insert Elements
2. Display Elements
3. Binary Search
4. Exit
Enter your choice: 3
Enter element to search: 2
Element found at index: 1

 ---Binary Search---
1. Insert Elements
2. Display Elements
3. Binary Search
4. Exit
Enter your choice: 4
Exiting Program
```

Figure 14: Output of Binary Search Program

5.2 Linear Search

```java
package Searching;

import java.util.Scanner;

public class LinearSearch {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[10];
        int n = 0, choice, key;
        boolean found;

```

```
        do{
            System.out.println("\n ---Linear Search Menu---");
            System.out.println("1. Insert Elements");
            System.out.println("2. Display Elements");
            System.out.println("3. Linear Search");
            System.out.println("4. Exit");

            System.out.print("Enter your choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter number of elements: ");
                    n = sc.nextInt();
                    System.out.println("Enter Elements: ");

                    for(int i=0; i<n; i++){
                        arr[i] = sc.nextInt();
                    }
                    break;

                case 2:
                    System.out.print("Array Elements: ");
                    for(int i=0; i<n; i++){
                        System.out.print(arr[i] + " ");
                    }
                    System.out.println();
                    break;

                case 3:
                    System.out.print("Enter element to search: ");
                    key = sc.nextInt();
                    found=false;

                    for(int i=0; i<n; i++){
                        if (arr[i] == key) {
                            System.out.println("Element found at index:
                               " + i);
                            found = true;
                            break;
                        }
                    }

                    if (!found)
```

```
55                          System.out.println("Element not found");
56                          break;
57
58                  case 4:
59                      System.out.println("Exiting Program");
60                      break;
61
62                  default:
63                      System.out.println("Invalid Choice");
64                      break;
65              }
66          }
67
68          while (choice != 4);
69          sc.close();
70      }
71  }
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr
ome/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979
ng.LinearSearch

 ---Linear Search Menu---
1. Insert Elements
2. Display Elements
3. Linear Search
4. Exit
Enter your choice: 1
Enter number of elements: 2
Enter Elements:
1
2

 ---Linear Search Menu---
1. Insert Elements
2. Display Elements
3. Linear Search
4. Exit
Enter your choice: 2
Array Elements: 1 2

 ---Linear Search Menu---
1. Insert Elements
2. Display Elements
3. Linear Search
4. Exit
Enter your choice: 3
```

Figure 15: Output of Linear Search Program

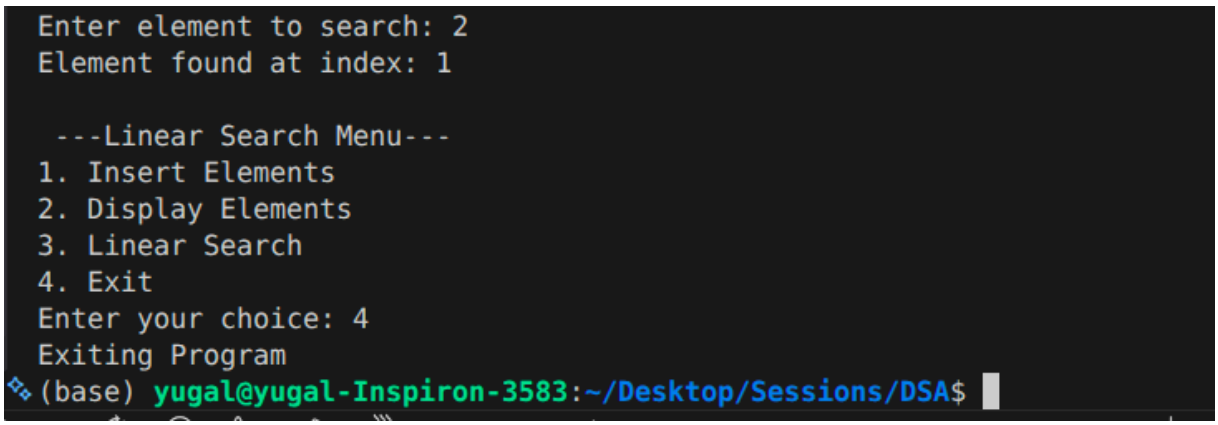

Figure 16: Output of Linear Search Program

Assignment 6: Recursion

6.1 Factorial of a Number using Recursion

```java
package Recursion;

import java.util.Scanner;

// Factorial Program --> eg: 4! = 4*3*2*1 = 24
public class Factorial {

    static int factorial(int n){
        if (n == 0)
            return 1;
        return n * factorial(n-1);
    }

    //Driver Code
    public static void main(String[] args) {

        System.out.println("Enter any number: ");
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        System.out.println("Factorial of "+ num + " is "+ factorial(num)
            );
        sc.close();
    }
}
```

Output

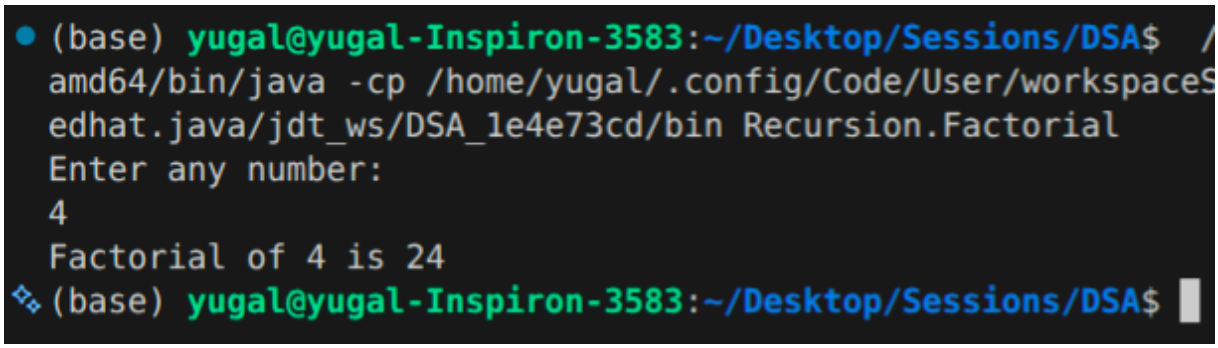

Figure 17: Output of Factorial Program

6.2 Fibonacci Series using Recursion

```java
package Recursion;
import java.util.Scanner;

//Fibonacci Series --> 0,1,0+1=1,1+1=2,2+1=3...
public class Fibonacci {
    static int fibonacci(int n){
        if (n == 0)
            return 0;
        if (n == 1)
            return 1;
        return fibonacci(n-1) + fibonacci(n-2);
    }
    //Driver Code
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Element: ");

        int terms = sc.nextInt();
        for(int i = 0; i<terms; i++){
            System.out.print(fibonacci(i) + " ");
            sc.close();
        }
    }
}
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  cd /home/yug
/java-11-openjdk-amd64/bin/java -cp /home/yugal/.config/Code/User/work
t.java/jdt_ws/DSA_1e4e73cd/bin Recursion.Fibonacci
Enter Element:
5
0 1 1 2 3 (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 18: Output of Fibonacci Program

6.3 Palindrome String using Recursion

```java
package Strings;
// Remove spaces from string
public class RemoveSpaces {
    public static void main(String[] args) {
```

```
        String str = "Every Expert Was Once A Beginner.";
        System.out.println(str.replace(" ", ""));
    }
}
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
ome/yugal/.config/Code/User/workspaceStorage/13eba2f5617b8
on.PalindromeRecursion
true
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 19: Output of Palindrome Program

6.4 Power of a number using Recursion

```
package Strings;

public class ReverseString {

    public static void main(String[] args) {

        String str = "Hello";
        String rev = "";

        for(int i = str.length()-1; i>=0; i--){
            rev = rev + str.charAt(i);
        }
        System.out.println("Reversed String: "+ rev);

    }
}
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
ome/yugal/.config/Code/User/workspaceStorage/13eba2f5617b8
on.PowerOfNumber
25
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 20: Output of Power of Number Program

6.5 Print numbers using Recursion

```
1  package Strings;
2
3  public class ReverseString {
4
5      public static void main(String[] args) {
6
7          String str = "Hello";
8          String rev = "";
9
10         for(int i = str.length()-1; i>=0; i--){
11             rev = rev + str.charAt(i);
12         }
13         System.out.println("Reversed String: "+ rev);
14
15     }
16 }
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/bin/env /usr/
ome/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979c97b038c39551
on.PrintNumbers
1 2 3 4 5 6 7 8 9 (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 21: Output of Print Numbers Program

6.6 Recursion Factorial of a Number

```
1  package Strings;
2
3  public class ReverseString {
```

```
4
5       public static void main(String[] args) {
6
7           String str = "Hello";
8           String rev = "";
9
10          for(int i = str.length()-1; i>=0; i--){
11              rev = rev + str.charAt(i);
12          }
13          System.out.println("Reversed String: "+ rev);
14
15      }
16  }
```

Output

```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
ome/yugal/.config/Code/User/workspaceStorage/13eba2f5617b8
on.RecursiveFactorial
Factorial of 5 is : 120
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 22: Output of Recursion Factorial Program

6.7 Reverse Number using Recursion

```
1   package Strings;
2
3   public class ReverseString {
4
5       public static void main(String[] args) {
6
7           String str = "Hello";
8           String rev = "";
9
10          for(int i = str.length()-1; i>=0; i--){
11              rev = rev + str.charAt(i);
12          }
13          System.out.println("Reversed String: "+ rev);
14
15      }
16  }
```

Output

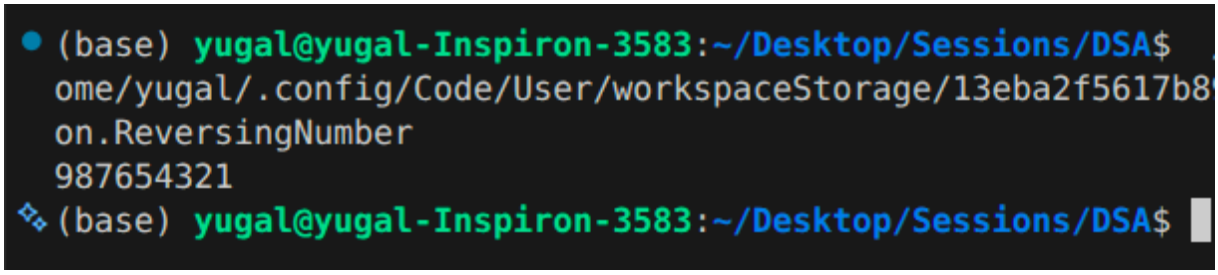

Figure 23: Output of Reverse Number Program

6.8 String Recursion Menu

```
1  package Strings;
2
3  public class ReverseString {
4
5      public static void main(String[] args) {
6
7          String str = "Hello";
8          String rev = "";
9
10         for(int i = str.length()-1; i>=0; i--){
11             rev = rev + str.charAt(i);
12         }
13         System.out.println("Reversed String: "+ rev);
14
15     }
16 }
```

Output

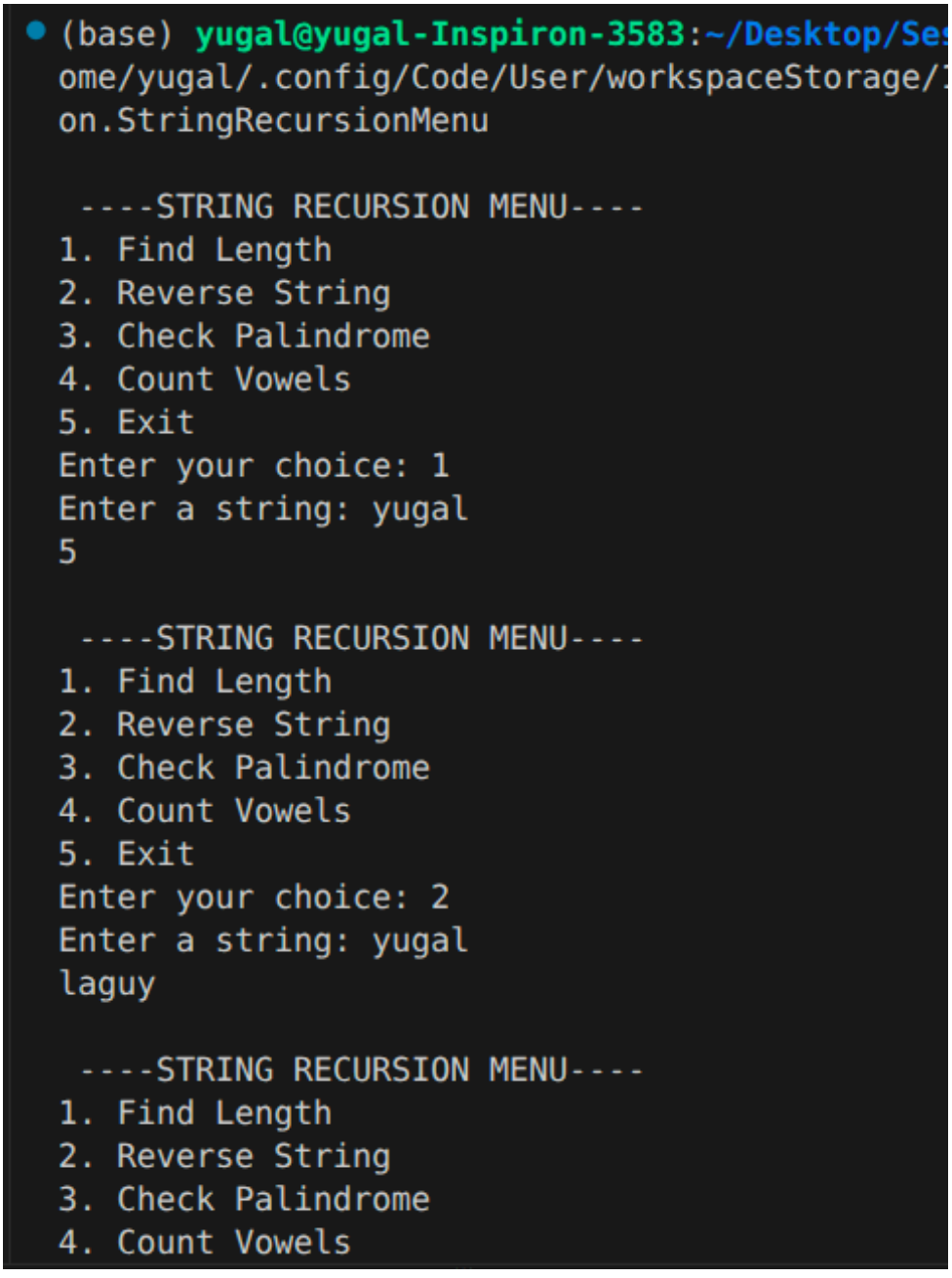

Figure 24: Output of String Recursion Menu Program

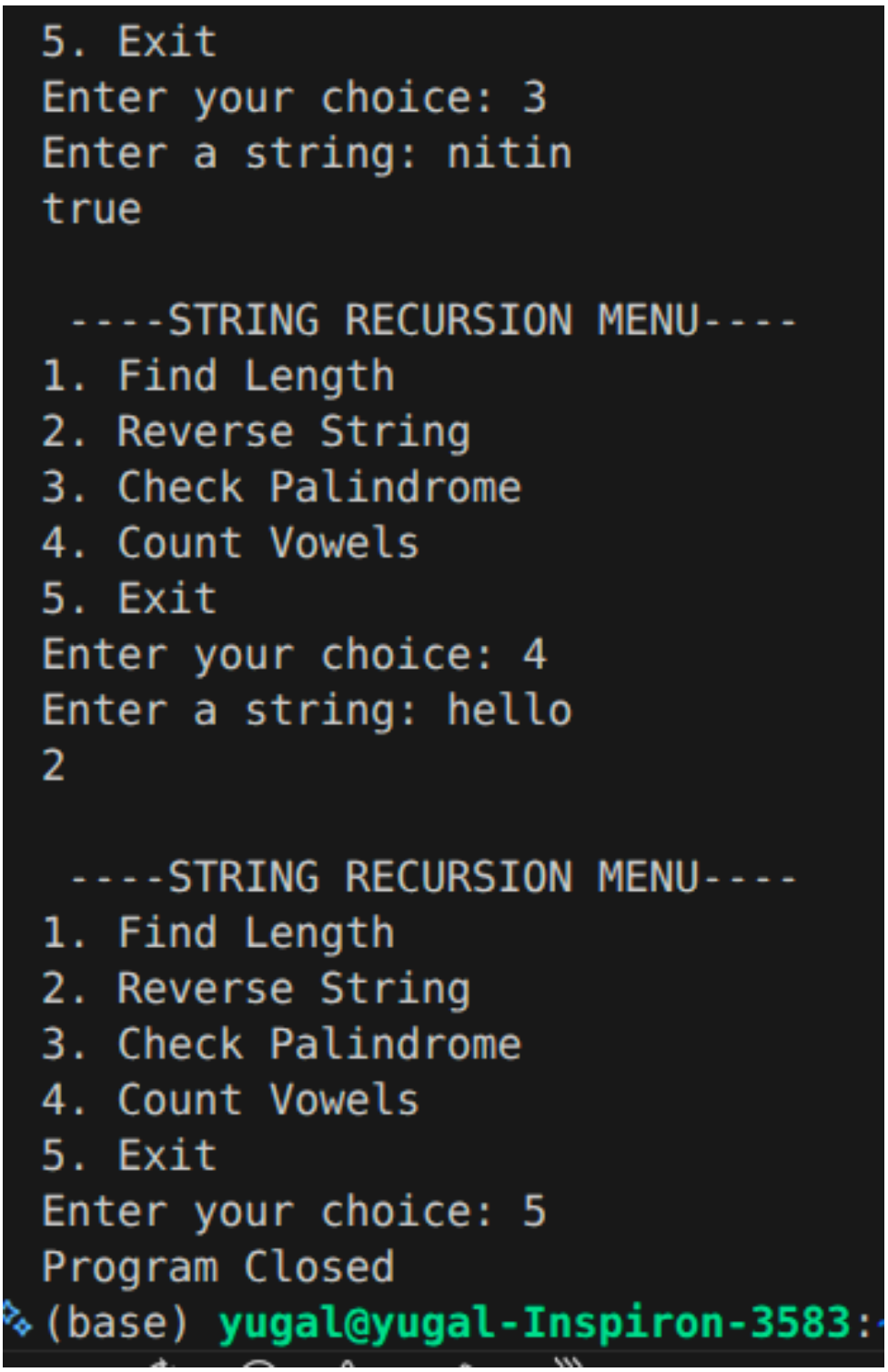

Figure 25: Output of String Recursion Menu Program

6.9 Sum of digits using Recursion

```
1 package Strings;
```

```java

public class ReverseString {

    public static void main(String[] args) {

        String str = "Hello";
        String rev = "";

        for(int i = str.length()-1; i>=0; i--){
            rev = rev + str.charAt(i);
        }
        System.out.println("Reversed String: "+ rev);

    }
}
```

Output

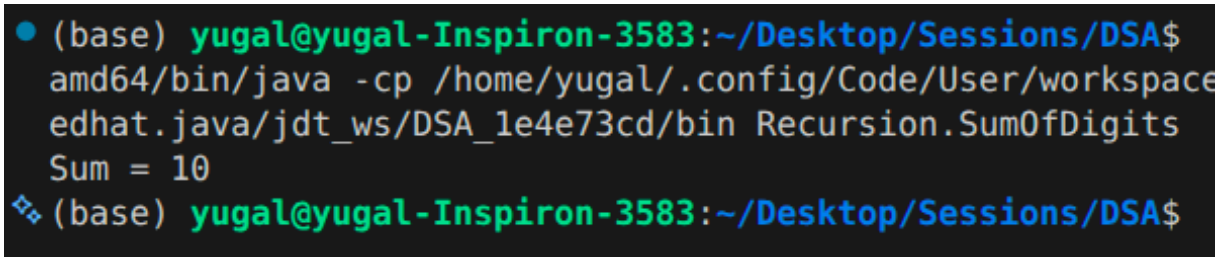

Figure 26: Output of Sum of Digits Program

Assignment 7: Linked List

7.1 Singly-Linked-List

```
public class SinglyLinkedList {
    // Node class
    class Node{
        int data;
        Node next;
        Node(int data){
            this.data = data;
            this.next = null;
        }
    }

    Node head = null;
    // Insert at end
    public void insertEnd(int data){
        Node newNode = new Node(data);
        if(head == null){
            head = newNode;
            return;
        }

        Node temp = head;
        while(temp.next != null){
            temp = temp.next;
        }
        temp.next = newNode;
    }

    // Insert at beginning
    public void insertBeginning(int data){
        Node newNode = new Node(data);
        newNode.next = head;
        head = newNode;
    }

    // Delete by value
    public void delete(int key){
        if (head == null)
            return;
        if (head.data == key) {
            head = head.next;
```

```
            return;
        }

        Node temp = head;
        while (temp.next != null && temp.next.data != key) {
            temp = temp.next;
        }
        if(temp.next != null){
            temp.next = temp.next.next;
        }
    }

    // Traverse
    public void display(){
        Node temp = head;
        while(temp != null){
            System.out.print(temp.data + "->");
            temp = temp.next;
        }

        System.out.println("null");
    }

    // Driver code
    public static void main(String[] args) {
        SinglyLinkedList list = new SinglyLinkedList();
        list.insertEnd(10);
        list.insertEnd(20);
        list.insertEnd(30);
        System.out.println("---Insert End---");
        list.display();
        list.insertBeginning(1);
        list.insertBeginning(2);
        list.insertBeginning(3);
        list.insertBeginning(4);
        list.insertBeginning(5);
        System.out.println("---Insert Beginning---");
        list.display();

        list.delete(30);
        System.out.println("---Delete 30---");
        list.display();
    }

```

```
85
86 }
```

Output

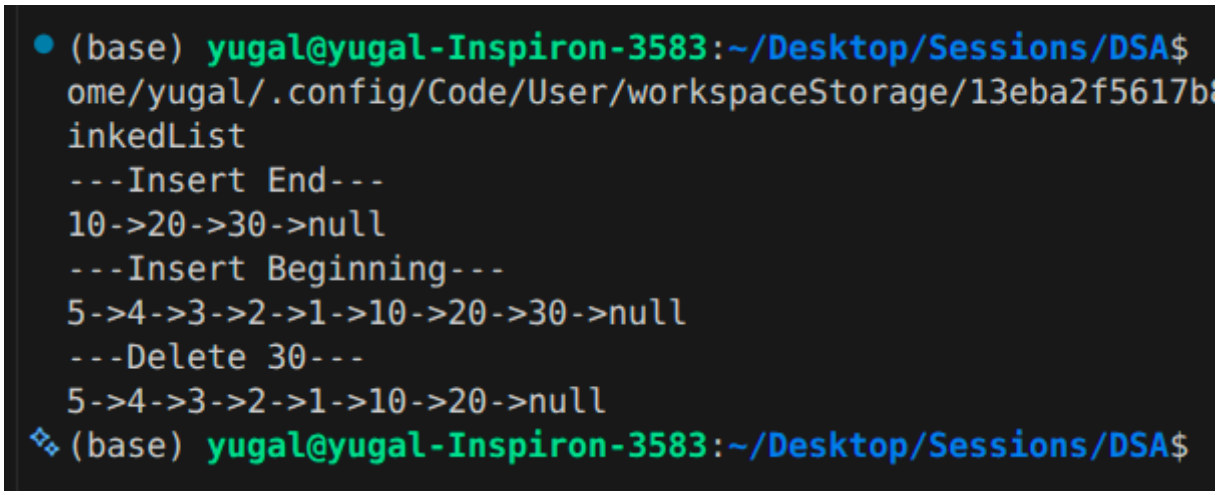

Figure 27: Singly Linked List Output

7.2 Doubly-Linked-List

```java
// Doubly Linked List Program
public class DoublyLinkedList {
    // Node class
    class Node{
        int data;
        Node prev; // Previous
        Node next; //Next Pointer
        Node(int data){
            this.data = data;
            this.prev = null;
            this.next = null;
        }
    }
    Node head = null;

    // Insert at beginning
    public void insertAtBeginning(int data){
        Node newNode = new Node(data);
        if(head == null){
            head = newNode;
            return;
        }
```

```
23          newNode.next = head;
24          head.prev = newNode;
25          head = newNode;
26      }
27
28      // Insert at end
29      public void insertAtEnd(int data){
30          Node newNode = new Node(data);
31          if(head == null){
32              head = newNode;
33              return;
34          }
35          Node temp = head;
36          while (temp.next != null) {
37              temp = temp.next;
38          }
39          temp.next = newNode;
40          newNode.prev = temp;
41      }
42
43      // Insert at Position (1- Based Index)
44      public void insertAtPosition(int data, int position){
45          if (position <= 0) {
46              System.out.println("Invalid Position");
47              return;
48          }
49          if (position == 1) {
50              insertAtBeginning(data);
51              return;
52          }
53          Node newNode = new Node(data);
54          Node temp = head;
55
56          for(int i=1; temp != null && i<position-1; i++){
57              temp = temp.next;
58          }
59          if (temp == null) {
60              System.out.println("Position out of range");
61              return;
62          }
63          newNode.next = temp.next;
64          if (temp.next != null) {
65              temp.next.prev = newNode;
66          }
```

```
67          temp.next = newNode;
68          newNode.prev = temp;
69      }
70
71      // Delete from Beginning
72      public void deleteFromBeginning(){
73          if (head == null) {
74              System.out.println("List is Empty");
75              return;
76          }
77          head = head.next;
78          if (head != null) {
79              head.prev = null;
80          }
81      }
82
83      // Delete from End
84      public void deleteFromEnd(){
85          if (head == null) {
86              System.out.println("List is Empty");
87              return;
88          }
89          if (head.next == null) {
90              head = null;
91              return;
92          }
93          Node temp = head;
94          while (temp.next != null) {
95              temp = temp.next;
96          }
97          temp.prev.next = null;
98      }
99
100     // Delete from Position
101     public void deleteFromPosition(int position){
102         if (head == null || position <= 0) {
103             System.out.println("Invalid Operation");
104             return;
105         }
106         if(position == 1){
107             deleteFromBeginning();
108             return;
109         }
110         Node temp = head;
```

```
111         for(int i =1; temp != null && i < position; i++){
112             temp = temp.next;
113         }
114         if (temp == null) {
115             System.out.println("Position out of Range.");
116             return;
117         }
118         if (temp.next != null) {
119             temp.next.prev = temp.prev;
120         }
121         if (temp.prev != null) {
122             temp.prev.next = temp.next;
123         }
124     }
125 
126     // Search
127     public void search(int key){
128         Node temp = head;
129         int position = 1;
130         while (temp != null) {
131             if (temp.data == key) {
132                 System.out.println("Element found at position: "+
                        position);
133                 return;
134             }
135             temp = temp.next;
136             position++;
137         }
138         System.out.println("Element Not Found!");
139     }
140 
141     // Display forward
142     public void displayForward(){
143         Node temp = head;
144         System.out.print("Forward: ");
145         while (temp != null) {
146             System.out.print(temp.data + " ");
147             temp = temp.next;
148         }
149         System.out.println();
150     }
151     // Display Backward
152     public void displayBackward(){
153         if (head == null)
```

```
            return;
        Node temp = head;
        while (temp.next != null) {
            temp = temp.next;
        }
        System.out.print("Backward: ");
        while (temp != null) {
            System.out.print(temp.data + " ");
            temp = temp.prev;
        }
        System.out.println();
    }

    // Driver code
    public static void main(String[] args) {
        DoublyLinkedList dll = new DoublyLinkedList();
        dll.insertAtBeginning(10);
        dll.insertAtBeginning(5);
        dll.insertAtBeginning(50);
        dll.insertAtBeginning(100);
        dll.insertAtEnd(20);
        dll.insertAtEnd(1000);
        dll.insertAtEnd(2000);
        dll.insertAtEnd(3000);
        dll.insertAtEnd(25);
        dll.insertAtPosition(15, 3);
        dll.insertAtPosition(500, 1);
        dll.displayForward();
        dll.displayBackward();
        dll.search(15);
        dll.deleteFromBeginning();
        dll.deleteFromEnd();
        dll.deleteFromPosition(2);
        dll.displayForward();
        dll.displayBackward();

    }
}
```

Output

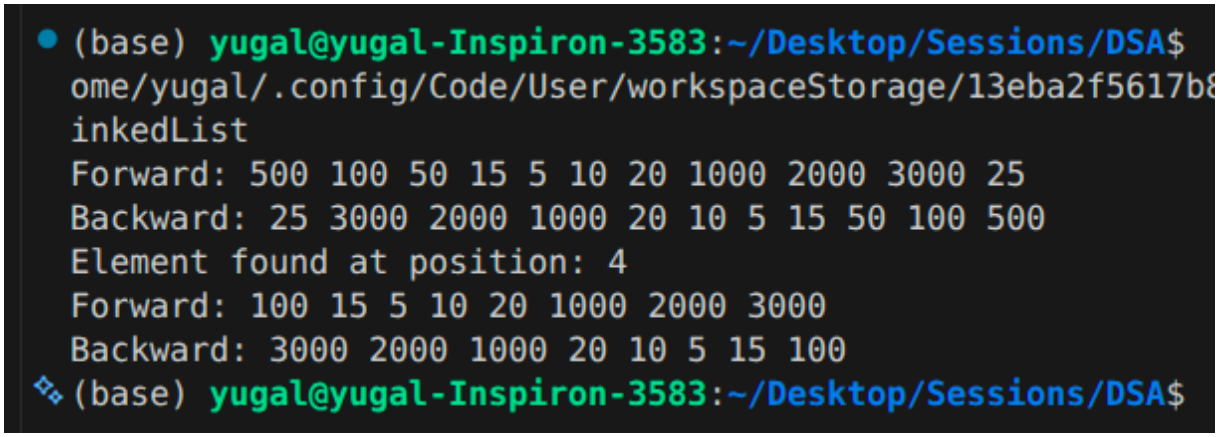

Figure 28: Output of Doubly Linked List Program