Assignment Day-4

Core Java with DS and Algorithms

Name: Joshnitha Rangolu

Task 1: Array Sorting and Searching

a) Implement a function called BruteForceSort that sorts an array using the brute force approach. Use this function to sort an array created with InitializeArray.

```
package day4; import java.util.Scanner; public
class BruteForceSort { public static void
main(String[] args) { Scanner scanner = new
Scanner(System.in); System.out.print("Enter the
size of the array: "); int size =
scanner.nextInt(); int[] arr =
InitializeArray(size);
System.out.println("Original array: ");
printArray(arr); arr = BruteForceSort(arr);
System.out.println("\nSorted array: ");
printArray(arr);
} public static int[] InitializeArray(int size)
{ int[] arr = new int[size]; for (int i = 0; i
< size; i++) { arr[i] = (int) (Math.random() *
100);
} return
arr;
}
public static int[] BruteForceSort(int[] arr) {
int n = arr.length; for (int i = 0; i < n - 1;
```

```
i++) { for (int j = 0; j < n - i - 1; j++) { if
(arr[j] > arr[j + 1]) { swap(arr, j, j + 1);
}
} } return arr; } public static void swap(int[]
arr, int i, int j) { int temp = arr[i]; arr[i] =
arr[j]; arr[j] = temp; } public static void
printArray(int[] arr) { for (int num : arr) {
System.out.print(num + " ");
}
System.out.println();
}
}
```

```
1  package day4;
2
3  import java.util.Scanner;
4
5  public class BruteForceSort {
6      public static void main(String[] args) {
7          Scanner scanner = new Scanner(System.in);
8
9          System.out.print("Enter the size of the array: ");
10         int size = scanner.nextInt();
11
12         int[] arr = InitializeArray(size);
13         System.out.println("Original array: ");
14         printArray(arr);
15
16         arr = BruteForceSort(arr);
17
18         System.out.println("\nSorted array: ");
19         printArray(arr);
20     }
```

```
Console ×
<terminated> BruteForceSort [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\java
Enter the size of the array: 5
Original array:
27 44 28 98 23

Sorted array:
23 27 28 44 98
```

b) Write a function named PerformLinearSearch that searches for a specific element in an array and returns the index of the element if found or -1 if not found.

```
package day4; import java.util.Scanner; public
class LinearSearch { public static void
main(String[] args) { Scanner scanner = new
```

```
Scanner(System.in); System.out.print("Enter the

size of the array: "); int size =

scanner.nextInt(); int[] arr = new int[size];

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

for (int i = 0; i < size; i++) { arr[i] =

scanner.nextInt();

}

System.out.print("Enter the element to search: ");

int searchElement = scanner.nextInt(); int index =

PerformLinearSearch(arr, searchElement); if (index

!= -1) {


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

} else {

System.out.println("Element " + searchElement + " not found in the
array.");

} } public static int PerformLinearSearch(int[] arr, int

searchElement) { int n = arr.length; for (int i = 0; i < n; i++) { if

(arr[i] == searchElement) { return i;

} } return

-1;

}

}
```

```java
package day4;

import java.util.Scanner;

public class LinearSearch {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the size of the array: ");
        int size = scanner.nextInt();

        int[] arr = new int[size];
        System.out.println("Enter the elements of the array: ");
        for (int i = 0; i < size; i++) {
          arr[i] = scanner.nextInt();
        }

        System.out.print("Enter the element to search: ");
        int searchElement = scanner.nextInt();
```

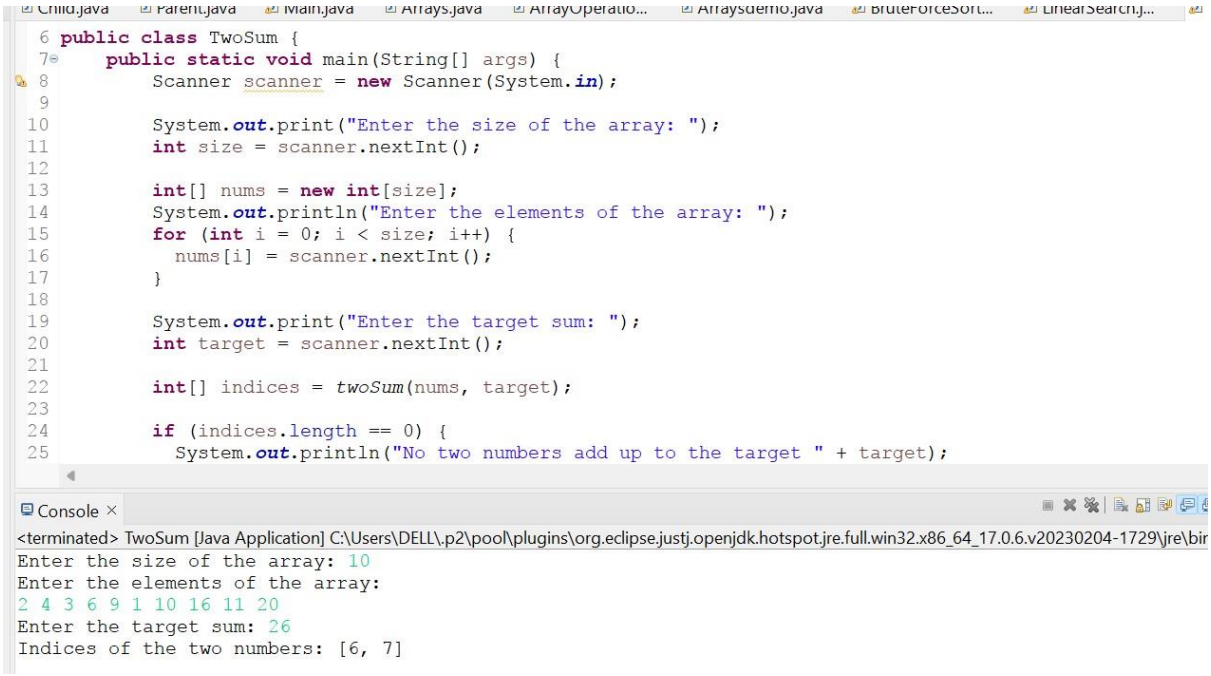
```
Console ×
<terminated> LinearSearch (1) [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\j
Enter the size of the array: 10
Enter the elements of the array:
2 3 1 8 20 21 4 67 54 9
Enter the element to search: 21
Element 21 found at index 5
```

Task 2: Two-Sum Problem

a) Given an array of integers, write a program that finds if there are two numbers that add up to a specific target. You may assume that each input would have exactly one solution, and you may not use the same element twice. Optimize the solution for time complexity.

```
package day4;

import java.util.HashMap;
import java.util.Scanner;

public class TwoSum {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);

            System.out.print("Enter the size of the array: ");
   int size = scanner.nextInt();

            int[] nums = new int[size];
            System.out.println("Enter the elements of the array: ");
            for (int i = 0; i < size; i++) {
              nums[i] = scanner.nextInt();
            }

    System.out.print("Enter the target sum: ");     int
target = scanner.nextInt();

            int[] indices = twoSum(nums, target);

            if (indices.length == 0) {
              System.out.println("No two numbers add up to the target " + target);
            } else {
```

```
                System.out.println("Indices of the two numbers: [" + indices[0] + ", " + indices[1]
+ "]");
            }
        }

        public static int[] twoSum(int[] nums, int target) {
HashMap<Integer, Integer> map = new HashMap<>();
            for (int i = 0; i < nums.length; i++) {
    int complement = target - nums[i];        if
(map.containsKey(complement)) {
                return new int[]{map.get(complement), i};
            }
            map.put(nums[i], i);
        }
        return new int[0];
    }

}
```

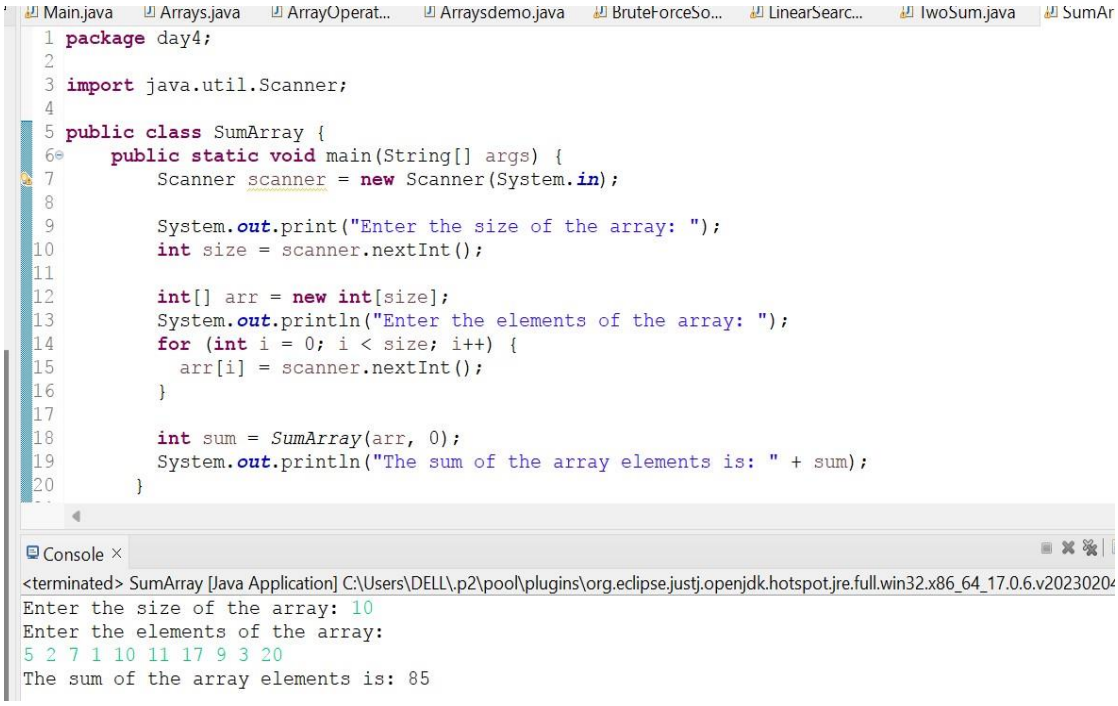


Task 3: Understanding Functions through Arrays

a) Write a recursive function named SumArray that calculates and returns the sum of elements in an array, demonstarte with example.

```
package day4; import java.util.Scanner;

public class SumArray { public static
```

```
void main(String[] args) { Scanner
scanner = new Scanner(System.in);
System.out.print("Enter the size of the array: ");
int size = scanner.nextInt(); int[] arr = new
int[size];
System.out.println("Enter the elements of the array: ");
for (int i = 0; i < size; i++) { arr[i] =
scanner.nextInt();
} int sum = SumArray(arr,
0);
System.out.println("The sum of the array elements is: " + sum);
} public static int SumArray(int[] arr, int index)
{ if (index == arr.length) { return 0; } return
arr[index] + SumArray(arr, index + 1);
}
}
```



Task 4: Advanced Array Operations

a) Implement a method SliceArray that takes an array, a starting index, and an end index, then returns a new array containing the elements from the start to the end index.

```
package day4; import

java.util.Scanner; public

class SliceArray { public

static void main(String[]

args) { Scanner scanner =

new Scanner(System.in);

System.out.print("Enter

the size of the array:

"); int size =

scanner.nextInt(); int[]

arr = new int[size];

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

for (int i = 0; i < size; i++) { arr[i] =

scanner.nextInt();

}

System.out.print("Enter the starting index for slicing (inclusive): ");

int startIndex = scanner.nextInt();

System.out.print("Enter the ending index for slicing (exclusive): ");

int endIndex = scanner.nextInt(); int[] slicedArray = SliceArray(arr,

startIndex, endIndex); System.out.print("Sliced array: ");

printArray(slicedArray);

} public static int[] SliceArray(int[] arr, int startIndex, int endIndex)

{ int sliceLength = endIndex - startIndex; if (sliceLength <= 0 ||

startIndex < 0 || endIndex > arr.length) { System.out.println("Invalid

slicing range. Returning empty array."); return new int[0];
```

```
} int[] slicedArray = new

int[sliceLength]; for (int i = 0; i <

sliceLength; i++) { slicedArray[i] =

arr[startIndex + i];

} return

slicedArray; }

public static void

printArray(int[]

arr) { for (int num

: arr) {

System.out.print(num + " ");

}

System.out.println();

}

}
```

```
1  package day4;
2
3  import java.util.Scanner;
4
5  public class SliceArray {
6      public static void main(String[] args) {
7              Scanner scanner = new Scanner(System.in);
8
9              System.out.print("Enter the size of the array: ");
10             int size = scanner.nextInt();
11
12             int[] arr = new int[size];
13             System.out.println("Enter the elements of the array: ");
14             for (int i = 0; i < size; i++) {
15               arr[i] = scanner.nextInt();
16             }
17
18             System.out.print("Enter the starting index for slicing (inclusive): ");
19             int startIndex = scanner.nextInt();
20
```

```
Console ×
<terminated> SliceArray [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw
Enter the size of the array: 10
Enter the elements of the array:
2 3 4 1 5 8 6 9 8 10
Enter the starting index for slicing (inclusive): 5
Enter the ending index for slicing (exclusive): 9
Sliced array: 8 6 9 8
```

b) Create a recursive function to find the nth element of a Fibonacci sequence and store the first n elements in an array

```
package day4; public class Fibonacci {

public static void main(String[] args) { int

n = 10; int[] fibonacciSequence =

fibonacciArray(n);

System.out.print("Fibonacci sequence (first " + n + " elements): "); for

(int num : fibonacciSequence) {

System.out.print(num + " ");

} } public static int fibonacci(int n) { if

(n <= 1) { return n;  } return fibonacci(n

- 1) + fibonacci(n - 2);

} public static int[] fibonacciArray(int n)

{ int[] arr = new int[n]; for (int i = 0; i

< n; i++) { arr[i] = fibonacci(i);

} return

arr;

}

}
```

ArrayOperat... Arraysdemo.java BruteForceSo... LinearSearc... TwoSum.java SumArray.java SliceArr

```java
public class Fibonacci {
    public static void main(String[] args) {
        int n = 10;
        int[] fibonacciSequence = fibonacciArray(n);

        System.out.print("Fibonacci sequence (first " + n + " elements): ");
        for (int num : fibonacciSequence) {
          System.out.print(num + " ");
        }
      }

      public static int fibonacci(int n) {
        if (n <= 1) {
          return n; // Base case: 0 or 1
        }
        return fibonacci(n - 1) + fibonacci(n - 2); // Recursive case: nth element is
      }

      public static int[] fibonacciArray(int n) {
        int[] arr = new int[n];
```

Console ×

<terminated> SliceArray [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6

```
Enter the size of the array: 10
Enter the elements of the array:
2 3 4 1 5 8 6 9 8 10
Enter the starting index for slicing (inclusive): 5
Enter the ending index for slicing (exclusive): 9
Sliced array: 8 6 9 8
```

Problems Javadoc Declaration TestNG × Coverage Gradle Tasks Gradle Executions

Search: Passed: 0 Failed: 0 Skipped: 0 Tests