

Subject: Algorithm and Data Structure Assignment 1

Solve the assignment with following thing to be added in each question.

- Program
- Flow chart
- Explanation
- Output
- Time and Space complexity

1. Printing Patterns

Problem: Write a Java program to print patterns such as a right triangle of stars.

Test Cases:

Input: n = 3

Output:

*

**

Input: n = 5

Output:

*

**

Ans:

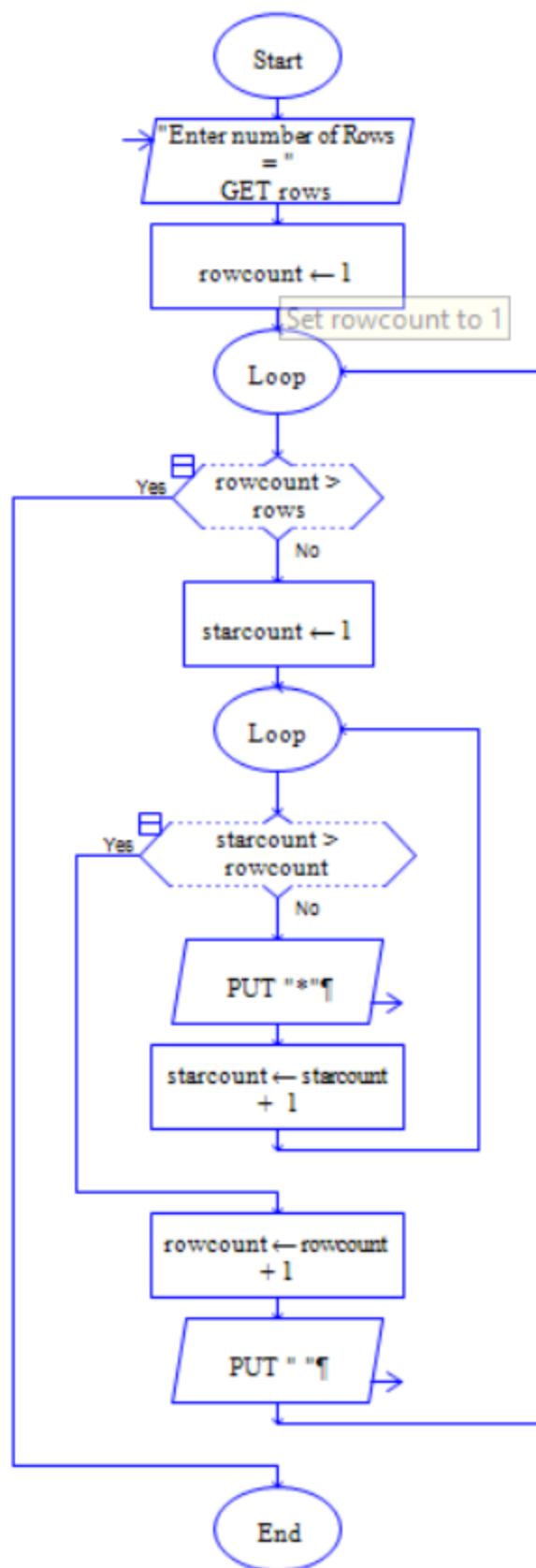
```
import java.util.Scanner;
```

```
public class PrintingPatterns {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the number of rows: ");  
        int n = scanner.nextInt();  
  
        for (int i = 1; i <= n; i++) {  
            for (int j = 1; j <= i; j++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
  
        scanner.close();  
    }  
}
```

}

```
J PrintingPatterns.java > PrintingPatterns > main(String[])
1  import java.util.Scanner;
2
3  public class PrintingPatterns {
    Run | Debug
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          System.out.print(s:"Enter the number of rows: ");
7          int n = scanner.nextInt();
8
9          for (int i = 1; i <= n; i++) {
10             for (int j = 1; j <= i; j++) {
11                 System.out.print(s:"*");
12             }
13             System.out.println();
14         }
15
16         scanner.close();
17     }
18 }
19
```

```
PS C:\Users\Sumit\Downloads\ADS Assignment 2> javac PrintingPatterns.java
PS C:\Users\Sumit\Downloads\ADS Assignment 2> java PrintingPatterns
Enter the number of rows: 5
*
**
***
****
*****
PS C:\Users\Sumit\Downloads\ADS Assignment 2> |
```



Time Complexity: ($O(n^2)$)

Space Complexity: ($O(1)$)

2. Remove Array Duplicates

Problem: Write a Java program to remove duplicates from a sorted array and return the new length of the array.

Test Cases:

Input: arr = [1, 1, 2]

Output: 2

Input: arr = [0, 0, 1, 1, 2, 2, 3, 3]

Output: 4

Ans:

```
public class RemoveDuplicates {
    public static int removeDuplicates(int[] nums) {
        if (nums.length == 0) return 0;

        int uniqueIndex = 1;
        for (int i = 1; i < nums.length; i++) {
            if (nums[i] != nums[i - 1]) {
                nums[uniqueIndex] = nums[i];
                uniqueIndex++;
            }
        }
        return uniqueIndex;
    }

    public static void main(String[] args) {
        int[] arr1 = {1, 1, 2};
        int[] arr2 = {0, 0, 1, 1, 2, 2, 3, 3};

        System.out.println("New length for arr1: " + removeDuplicates(arr1)); // Output: 2
        System.out.println("New length for arr2: " + removeDuplicates(arr2)); // Output: 4
    }
}
```

• }

```
J ArrayDuplicates.java > ...
1 public class ArrayDuplicates {
2     public static int removeDuplicates(int[] nums) {
3         if (nums.length == 0) return 0;
4
5         int uniqueIndex = 1;
6         for (int i = 1; i < nums.length; i++) {
7             if (nums[i] != nums[i - 1]) {
8                 nums[uniqueIndex] = nums[i];
9                 uniqueIndex++;
10            }
11        }
12        return uniqueIndex;
13    }
14
15    Run | Debug
16    public static void main(String[] args) {
17        int[] arr1 = {1, 1, 2};
18        int[] arr2 = {0, 0, 1, 1, 2, 2, 3, 3};
19
20        System.out.println("New length for arr1: " + removeDuplicates(arr1)); // Output: 2
21        System.out.println("New length for arr2: " + removeDuplicates(arr2)); // Output: 4
22    }
23 }
```

```
PS C:\Users\Sumit\Downloads\ADS Assignment 2> javac ArrayDuplicates.java
PS C:\Users\Sumit\Downloads\ADS Assignment 2> java ArrayDuplicates
New length for arr1: 2
New length for arr2: 4
PS C:\Users\Sumit\Downloads\ADS Assignment 2> █
```

Time Complexity: ($O(n)$)
Space Complexity: ($O(1)$)

3. Remove White Spaces from String

Problem: Write a Java program to remove all white spaces from a given string.

Test Cases:

Input: "Hello World"

Output: "HelloWorld"

Input: " Java Programming "

Output: "JavaProgramming"

Ans:

```
public class RemoveWhiteSpaces {  
    public static void main(String[] args) {  
        String str1 = "Hello World";  
        String str2 = " Java Programming ";  
  
        System.out.println("Original: \"" + str1 + "\" -> Without spaces: \"" + removeWhiteSpaces(str1) +  
        "\"");  
        System.out.println("Original: \"" + str2 + "\" -> Without spaces: \"" + removeWhiteSpaces(str2) +  
        "\"");  
    }  
  
    public static String removeWhiteSpaces(String str) {  
        return str.replaceAll("\\s", "");  
    }  
}
```

```
PS C:\Users\Sumit\Downloads\ADS Assignment 2> javac RemoveWhiteSpaces.java  
PS C:\Users\Sumit\Downloads\ADS Assignment 2> java RemoveWhiteSpaces  
Original: "Hello World" -> Without spaces: "HelloWorld"  
Original: " Java Programming " -> Without spaces: "JavaProgramming"  
PS C:\Users\Sumit\Downloads\ADS Assignment 2> 
```

```
J RemoveWhiteSpaces.java > ...
1  public class RemoveWhiteSpaces {
2      Run | Debug
3      public static void main(String[] args) {
4          String str1 = "Hello World";
5          String str2 = " Java Programming ";
6
7          System.out.println("Original: \"" + str1 + "\" -> Without spaces: \"" + removeWhiteSpaces(str1) + "\"");
8          System.out.println("Original: \"" + str2 + "\" -> Without spaces: \"" + removeWhiteSpaces(str2) + "\"");
9      }
10
11     public static String removeWhiteSpaces(String str) {
12         return str.replaceAll(regex:"\\s", replacement:"");
13     }
14 }
15
```

Time Complexity: $O(n)$

Space Complexity: $O(n)$

4. Reverse a String

Problem: Write a Java program to reverse a given string.

Test Cases:

Input: "hello"

Output: "olleh"

Input: "Java"

Output: "avaJ"

Ans:

```
public class ReverseString {
    public static void main(String[] args) {
        String str1 = "hello";
        String str2 = "Java";

        System.out.println("Input: \"" + str1 + "\"");
        System.out.println("Output: \"" + reverseString(str1) + "\"");

        System.out.println("Input: \"" + str2 + "\"");
        System.out.println("Output: \"" + reverseString(str2) + "\"");
    }

    public static String reverseString(String str) {
        StringBuilder reversedStr = new StringBuilder(str);
        return reversedStr.reverse().toString();
    }
}
```


J ReverseString.java > ...

```
1  public class ReverseString {  
    Run | Debug  
2      public static void main(String[] args) {  
3          String str1 = "hello";  
4          String str2 = "Java";  
5  
6          System.out.println("Input: \"" + str1 + "\"");  
7          System.out.println("Output: \"" + reverseString(str1) + "\"");  
8  
9          System.out.println("Input: \"" + str2 + "\"");  
10         System.out.println("Output: \"" + reverseString(str2) + "\"");  
11     }  
12  
13     public static String reverseString(String str) {  
14         StringBuilder reversedStr = new StringBuilder(str);  
15         return reversedStr.reverse().toString();  
16     }  
17 }  
18  
19
```

PS C:\Users\Sumit\Downloads\ADS Assignment 2>

javac ReverseString.java

PS C:\Users\Sumit\Downloads\ADS Assignment 2> java ReverseString

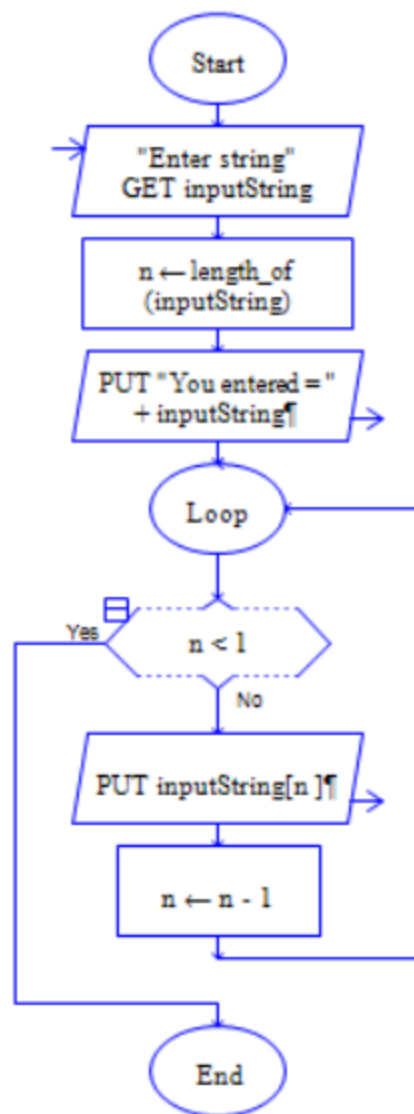
Input: "hello"

Output: "olleh"

Input: "Java"

Output: "avaJ"

PS C:\Users\Sumit\Downloads\ADS Assignment 2> █



Time Complexity: $O(n)$
Space Complexity: $O(n)$

5. Reverse Array in Place

Problem: Write a Java program to reverse an array in place.

Test Cases:

Input: arr = [1, 2, 3, 4]

Output: [4, 3, 2, 1]

Input: arr = [7, 8, 9]

Output: [9, 8, 7]

Ans:

```
public class ReverseArray {  
    public static void main(String[] args) {  
        int[] arr1 = {1, 2, 3, 4};  
        int[] arr2 = {7, 8, 9};  
  
        reverseArray(arr1);  
        reverseArray(arr2);  
  
        System.out.println("Reversed arr1: " + java.util.Arrays.toString(arr1));  
        System.out.println("Reversed arr2: " + java.util.Arrays.toString(arr2));  
    }  
  
    public static void reverseArray(int[] arr) {  
        int left = 0, right = arr.length - 1;  
        while (left < right) {  
            int temp = arr[left];  
            arr[left] = arr[right];  
            arr[right] = temp;  
            left++;  
            right--;  
        }  
    }  
}
```

}

```
J ReverseArray.java > ReverseArray
1  public class ReverseArray {
    Run | Debug
2      public static void main(String[] args) {
3          int[] arr1 = {1, 2, 3, 4};
4          int[] arr2 = {7, 8, 9};
5
6          reverseArray(arr1);
7          reverseArray(arr2);
8
9          System.out.println("Reversed arr1: " + java.util.Arrays.toString(arr1));
10         System.out.println("Reversed arr2: " + java.util.Arrays.toString(arr2));
11     }
12
13     public static void reverseArray(int[] arr) {
14         int left = 0, right = arr.length - 1;
15         while (left < right) {
16             int temp = arr[left];
17             arr[left] = arr[right];
18             arr[right] = temp;
19             left++;
20             right--;
21         }
22     }
23 }
24
```

```
PS C:\Users\Sumit\Downloads\ADS Assignment 2> javac ReverseArray.java
PS C:\Users\Sumit\Downloads\ADS Assignment 2> java ReverseArray
Reversed arr1: [4, 3, 2, 1]
Reversed arr2: [9, 8, 7]
PS C:\Users\Sumit\Downloads\ADS Assignment 2>
```

Time Complexity: (O(n))
Space Complexity: (O(1))

6. Reverse Words in a String

Problem: Write a Java program to reverse the words in a given sentence.

Test Cases:

Input: "Hello World"

Output: "World Hello"

Input: "Java Programming"

Output: "Programming Java"

Ans:

```
public class Reversewords {
    public static String reverseWords(String sentence) {
        // Split the sentence into words
        String[] words = sentence.split(" ");
        StringBuilder reversedSentence = new StringBuilder();

        // Iterate over the words in reverse order
        for (int i = words.length - 1; i >= 0; i--) {
            reversedSentence.append(words[i]);
            if (i != 0) {
                reversedSentence.append(" ");
            }
        }
        return reversedSentence.toString();
    }

    public static void main(String[] args) {
        String input1 = "Hello World";
        String input2 = "Java Programming";

        System.out.println("Input: " + input1);
        System.out.println("Output: " + reverseWords(input1));

        System.out.println("Input: " + input2);
        System.out.println("Output: " + reverseWords(input2));
    }
}
```

```

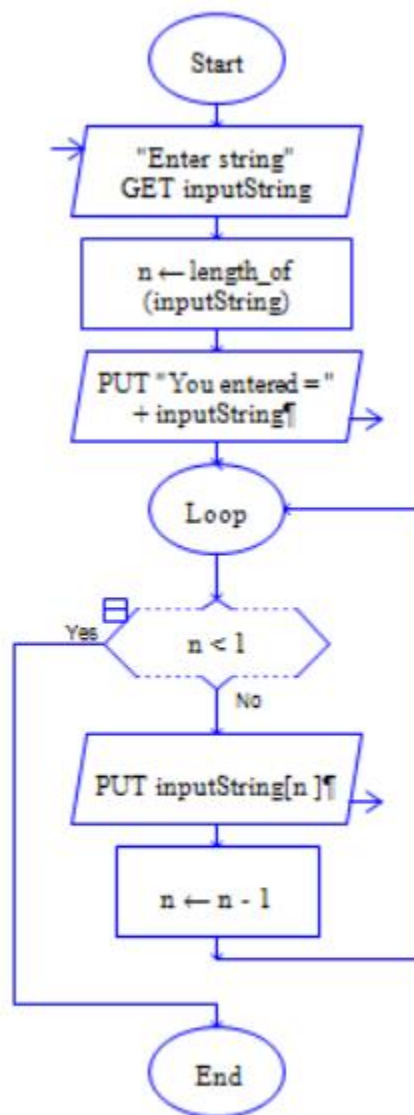
J Reversewords.java > Reversewords
1  public class Reversewords {
2      public static String reverseWords(String sentence) {
3          // Split the sentence into words
4          String[] words = sentence.split(regex:" ");
5          StringBuilder reversedSentence = new StringBuilder();
6
7          // Iterate over the words in reverse order
8          for (int i = words.length - 1; i >= 0; i--) {
9              reversedSentence.append(words[i]);
10             if (i != 0) {
11                 reversedSentence.append(str:" ");
12             }
13         }
14         return reversedSentence.toString();
15     }
16
17     Run | Debug
18     public static void main(String[] args) {
19         String input1 = "Hello World";
20         String input2 = "Java Programming";
21
22         System.out.println("Input: " + input1);
23         System.out.println("Output: " + reverseWords(input1));
24
25         System.out.println("Input: " + input2);
26         System.out.println("Output: " + reverseWords(input2));
27     }
28
29

```

```

PS C:\Users\Sumit\Downloads\ADS Assignment 2>
javac Reversewords.java
PS C:\Users\Sumit\Downloads\ADS Assignment 2> java Reversewords
Input: Hello World
Output: World Hello
Input: Java Programming
Output: Programming Java
PS C:\Users\Sumit\Downloads\ADS Assignment 2>

```



Time Complexity: $O(n)$
Space Complexity: $O(n)$

7. Reverse a Number

Problem: Write a Java program to reverse a given number.

Test Cases:

Input: 12345

Output: 54321

Input: -9876

Output: -6789

Ans:

```
public class ReverseNumber {
    public static int reverseNumber(int num) {
        int reversed = 0;
        int sign = num < 0 ? -1 : 1;
        num = Math.abs(num);

        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }

        return reversed * sign;
    }

    public static void main(String[] args) {
        int input1 = 12345;
        int input2 = -9876;

        System.out.println("Input: " + input1);
        System.out.println("Output: " + reverseNumber(input1));

        System.out.println("Input: " + input2);
        System.out.println("Output: " + reverseNumber(input2));
    }
}
```


J ReverseNumber.java > ...

```
1  ∨ public class ReverseNumber {
2  ∨      public static int reverseNumber(int num) {
3          int reversed = 0;
4          int sign = num < 0 ? -1 : 1;
5          num = Math.abs(num);
6
7          while (num != 0) {
8              int digit = num % 10;
9              reversed = reversed * 10 + digit;
10             num /= 10;
11         }
12
13         return reversed * sign;
14     }
15
16     Run | Debug
17     ∨ public static void main(String[] args) {
18         int input1 = 12345;
19         int input2 = -9876;
20
21         System.out.println("Input: " + input1);
22         System.out.println("Output: " + reverseNumber(input1));
23
24         System.out.println("Input: " + input2);
25         System.out.println("Output: " + reverseNumber(input2));
26     }
27
28
```

```
PS C:\Users\Sumit\Downloads\ADS Assignment 2>
```

```
javac ReverseNumber.java
```

```
PS C:\Users\Sumit\Downloads\ADS Assignment 2> java ReverseNumber
```

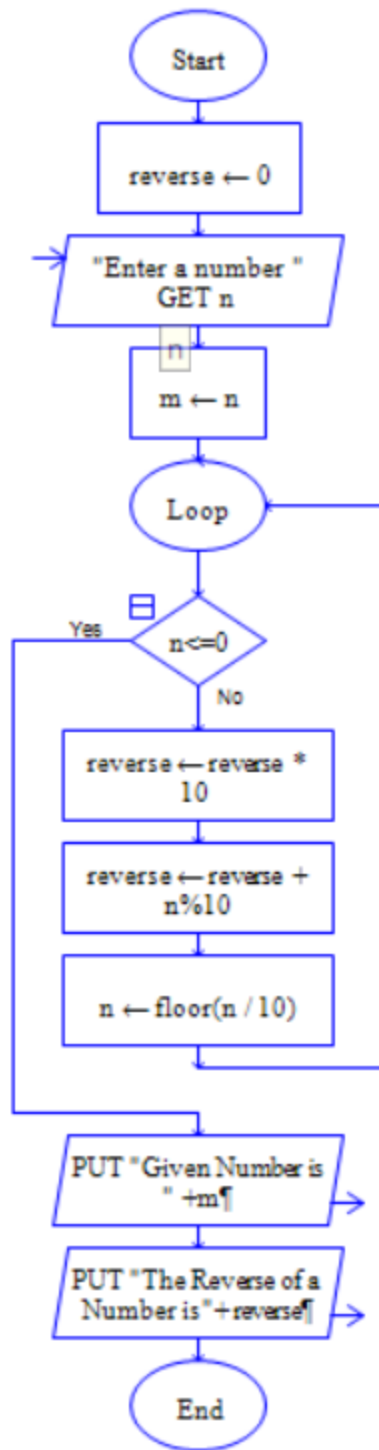
```
Input: 12345
```

```
Output: 54321
```

```
Input: -9876
```

```
Output: -6789
```

```
PS C:\Users\Sumit\Downloads\ADS Assignment 2> 
```



Time Complexity: $O(d)$
Space Complexity: $O(1)$

8. Array Manipulation

Problem: Perform a series of operations to manipulate an array based on range update queries. Each query adds a value to a range of indices.

Test Cases:

Input: n = 5, queries = [[1, 2, 100], [2, 5, 100], [3, 4, 100]]

Output: 200

Input: n = 4, queries = [[1, 3, 50], [2, 4, 70]]

Output: 120

Ans:

```
import java.util.Arrays;
```

```
public class ArrayManipulation {
    public static long arrayManipulation(int n, int[][] queries) {
        long[] array = new long[n + 1];

        // Apply the range updates
        for (int[] query : queries) {
            int start = query[0] - 1;
            int end = query[1];
            int value = query[2];

            array[start] += value;
            if (end < n) {
                array[end] -= value;
            }
        }

        // Calculate the maximum value after all updates
        long max = 0;
        long current = 0;
        for (long num : array) {
            current += num;
            if (current > max) {
                max = current;
            }
        }

        return max;
    }

    public static void main(String[] args) {
        int n1 = 5;
        int[][] queries1 = {{1, 2, 100}, {2, 5, 100}, {3, 4, 100}};
        System.out.println("Output: " + arrayManipulation(n1, queries1)); // Output: 200
    }
}
```

```

    int n2 = 4;
    int[][] queries2 = {{1, 3, 50}, {2, 4, 70}};
    System.out.println("Output: " + arrayManipulation(n2, queries2)); // Output: 120
}
}

```

```

J ArrayManipulation.java > ...
1  import java.util.Arrays;
2
3  public class ArrayManipulation {
4      public static long arrayManipulation(int n, int[][] queries) {
5          long[] array = new long[n + 1];
6
7          // Apply the range updates
8          for (int[] query : queries) {
9              int start = query[0] - 1;
10             int end = query[1];
11             int value = query[2];
12
13             array[start] += value;
14             if (end < n) {
15                 array[end] -= value;
16             }
17         }
18
19         // Calculate the maximum value after all updates
20         long max = 0;
21         long current = 0;
22         for (long num : array) {
23             current += num;
24             if (current > max) {
25                 max = current;
26             }
27         }
28
29         return max;
30     }

```

```

PS C:\Users\Sumit\Downloads\ADS Assignment 2> javac ArrayManipulation.java
PS C:\Users\Sumit\Downloads\ADS Assignment 2> java ArrayManipulation
Output: 200
Output: 120
PS C:\Users\Sumit\Downloads\ADS Assignment 2>

```

Time Complexity: $O(n + m)$

Space Complexity: $O(n)$

9. String Palindrome

Problem: Write a Java program to check if a given string is a palindrome.

Test Cases:

Input: "madam"

Output: true

Input: "hello"

Output: false

Ans:

```
public class Palindrome {
    public static boolean isPalindrome(String str) {
        int left = 0;
        int right = str.length() - 1;

        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }

    public static void main(String[] args) {
        String input1 = "madam";
        String input2 = "hello";

        System.out.println("Input: " + input1);
        System.out.println("Output: " + isPalindrome(input1));

        System.out.println("Input: " + input2);
        System.out.println("Output: " + isPalindrome(input2));
    }
}
```

}

```
J Palindrome.java > Palindrome
1 public class Palindrome {
2     public static boolean isPalindrome(String str) {
3         int left = 0;
4         int right = str.length() - 1;
5
6         while (left < right) {
7             if (str.charAt(left) != str.charAt(right)) {
8                 return false;
9             }
10            left++;
11            right--;
12        }
13        return true;
14    }
15
16    Run | Debug
17    public static void main(String[] args) {
18        String input1 = "madam";
19        String input2 = "hello";
20
21        System.out.println("Input: " + input1);
22        System.out.println("Output: " + isPalindrome(input1));
23
24        System.out.println("Input: " + input2);
25        System.out.println("Output: " + isPalindrome(input2));
26    }
27
28 }
```

```
PS C:\Users\Sumit\Downloads\ADS Assignment 2>
javac Palindrome.java
PS C:\Users\Sumit\Downloads\ADS Assignment 2> java Palindrome
Input: madam
Output: true
Input: hello
Output: false
PS C:\Users\Sumit\Downloads\ADS Assignment 2> 
```

Time Complexity: $O(n)$
Space Complexity: $O(1)$

10. Array Left Rotation

Problem: Write a Java program to rotate an array to the left by d positions.

Test Cases:

Input: arr = [1, 2, 3, 4, 5], d = 2

Output: [3, 4, 5, 1, 2]

Input: arr = [10, 20, 30, 40], d = 1

Output: [20, 30, 40, 10]

Ans:

```
public class ArrayLeftRotation {
    public static void rotateLeft(int[] arr, int d) {
        int n = arr.length;
        d = d % n; // In case d is greater than n
        reverseArray(arr, 0, d - 1);
        reverseArray(arr, d, n - 1);
        reverseArray(arr, 0, n - 1);
    }

    private static void reverseArray(int[] arr, int start, int end) {
        while (start < end) {
            int temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;
            start++;
            end--;
        }
    }

    public static void main(String[] args) {
        int[] arr1 = {1, 2, 3, 4, 5};
        int d1 = 2;
        rotateLeft(arr1, d1);
        System.out.println("Output: " + java.util.Arrays.toString(arr1)); // Output: [3, 4, 5, 1, 2]

        int[] arr2 = {10, 20, 30, 40};
        int d2 = 1;
        rotateLeft(arr2, d2);
        System.out.println("Output: " + java.util.Arrays.toString(arr2)); // Output: [20, 30, 40, 10]
    }
}
```



```
J ReverseWords.java J ReverseNumber.java J ArrayManipulation.java 1 J Palindrome.java J ArrayLeftRotation.java X ▸ ▾
J ArrayLeftRotation.java > ...
1 public class ArrayLeftRotation {
2     public static void rotateLeft(int[] arr, int d) {
3         int n = arr.length;
4         d = d % n; // In case d is greater than n
5         reverseArray(arr, start:0, d - 1);
6         reverseArray(arr, d, n - 1);
7         reverseArray(arr, start:0, n - 1);
8     }
9
10    private static void reverseArray(int[] arr, int start, int end) {
11        while (start < end) {
12            int temp = arr[start];
13            arr[start] = arr[end];
14            arr[end] = temp;
15            start++;
16            end--;
17        }
18    }
19
20    Run | Debug
21    public static void main(String[] args) {
22        int[] arr1 = {1, 2, 3, 4, 5};
23        int d1 = 2;
24        rotateLeft(arr1, d1);
25        System.out.println("Output: " + java.util.Arrays.toString(arr1)); // Output: [3, 4, 5, 1, 2]
26
27        int[] arr2 = {10, 20, 30, 40};
28        int d2 = 1;
29        rotateLeft(arr2, d2);
30        System.out.println("Output: " + java.util.Arrays.toString(arr2)); // Output: [20, 30, 40, 10]
```

```
PS C:\Users\Sumit\Downloads\ADS Assignment 2>
javac ArrayLeftRotation.java
PS C:\Users\Sumit\Downloads\ADS Assignment 2> java ArrayLeftRotation
Output: [3, 4, 5, 1, 2]
Output: [20, 30, 40, 10]
PS C:\Users\Sumit\Downloads\ADS Assignment 2> █
```

Time Complexity: $O(n)$
Space Complexity: $O(1)$