**Subject: Algorithm and Data Structure**

**Assignment 1**

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

-Program

-Flow chart

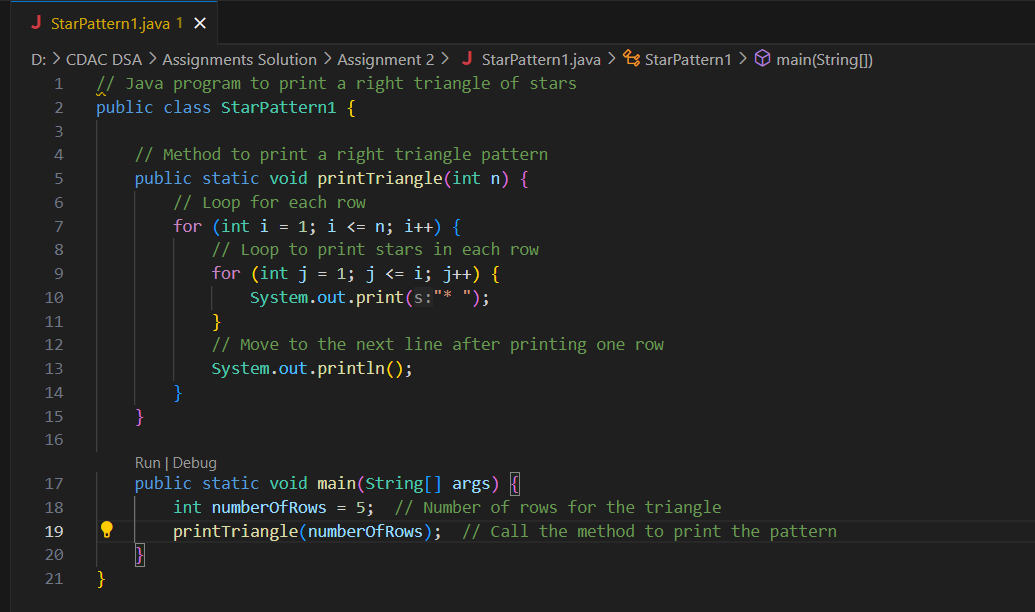
-Explanation

-Output

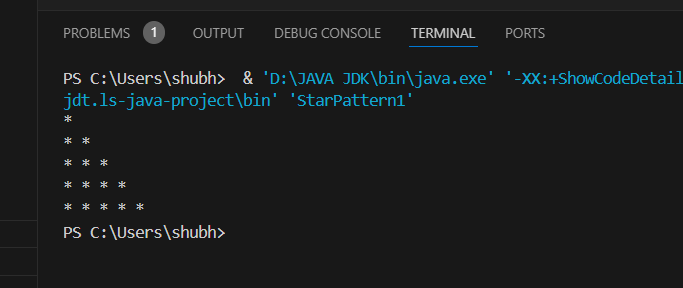
-Time and Space complexity

1. Printing Patterns

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



OUTPUT



**FLOWCHART:**

1. **Start**
2. **Input**: Get the number of rows (n) for the triangle.
3. **Outer Loop (i=1 to n)**: For each row:
   * **Inner Loop (j=1 to i)**: Print stars i times for the current row.
   * Move to the next line after each row.
4. **End**

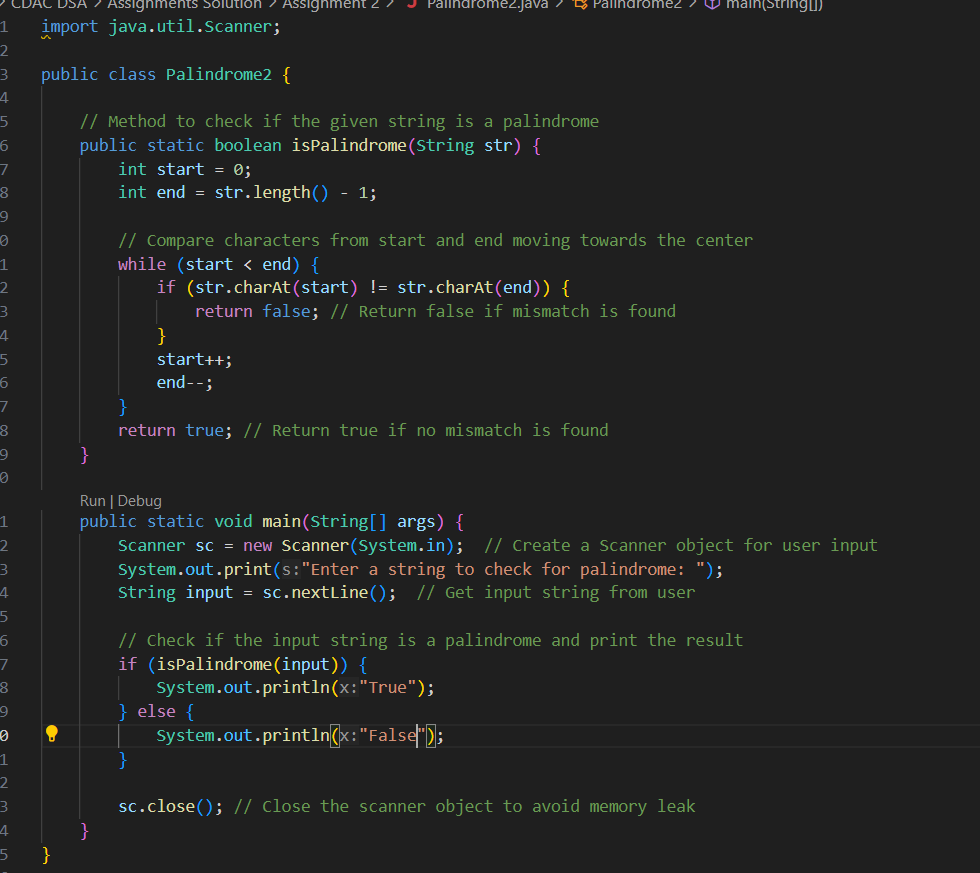
SPACE COMPLEXITY IS O(1)

TIME COMPLEXITY IS O(N^2)

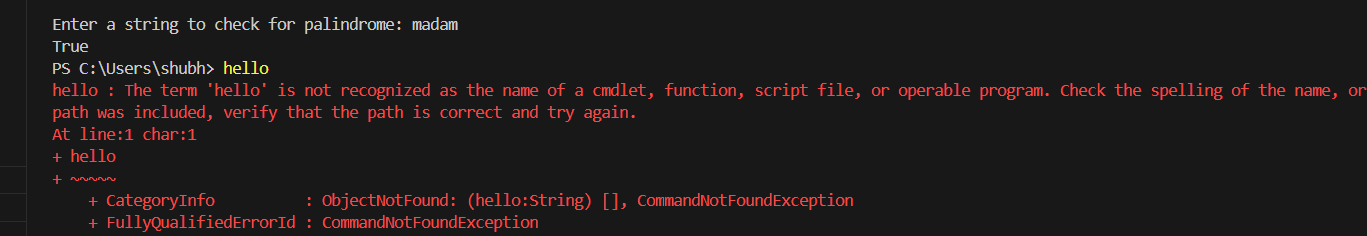
**-------------------------------------------------------------------------------------------------------------------------------**

9. String Palindrome

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



OUTPUT



**Flowchart:**

1. **Start**
2. **Input**: Get a string str from the user.
3. **Initialize**: Set start = 0 and end = str.length() - 1.
4. **Check each character**:
   * If str[start] != str[end], return "Not a palindrome".
   * If they match, increment start and decrement end.
5. If all characters match, return "Palindrome".
6. End

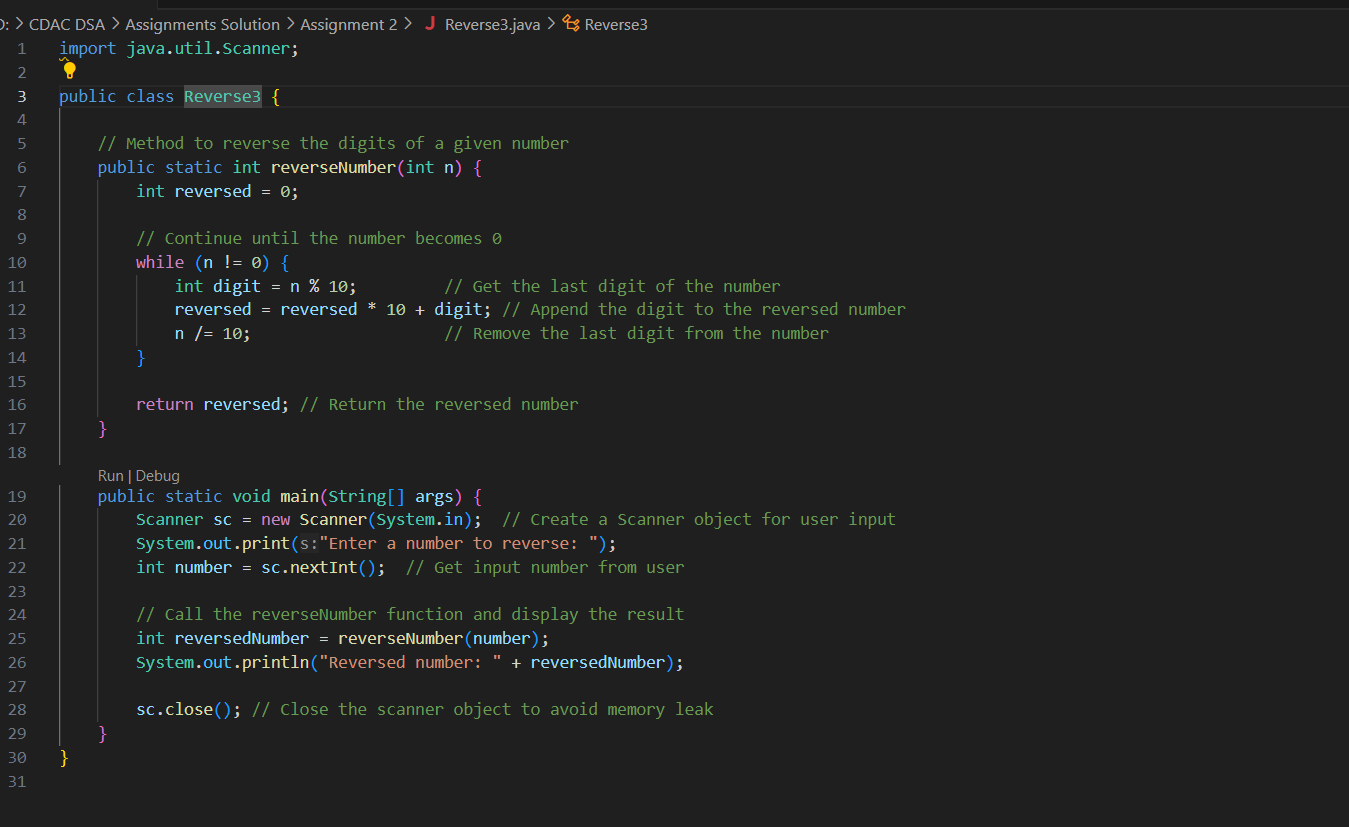
THE TIME COMPLEXITY IS O(N)

THE SPACE COMPLEXITY IS O(1)

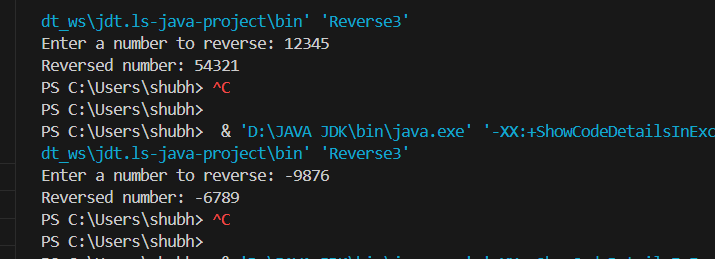
**-------------------------------------------------------------------------------------------------------------------------------**

**7. Reverse a Number**

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

****

**OUTPUT**

****

FLOWCHART:

1. **Start**
2. **Input: Get the number n from the user.**
3. **Initialize: Set reversed = 0.**
4. **Loop (until n != 0):**
   * **Get the last digit using n % 10.**
   * **Append the digit to reversed by multiplying the current reversed by 10 and adding the digit.**
   * **Remove the last digit by updating n = n / 10.**
5. **Once n == 0, output the reversed number.**
6. **End**

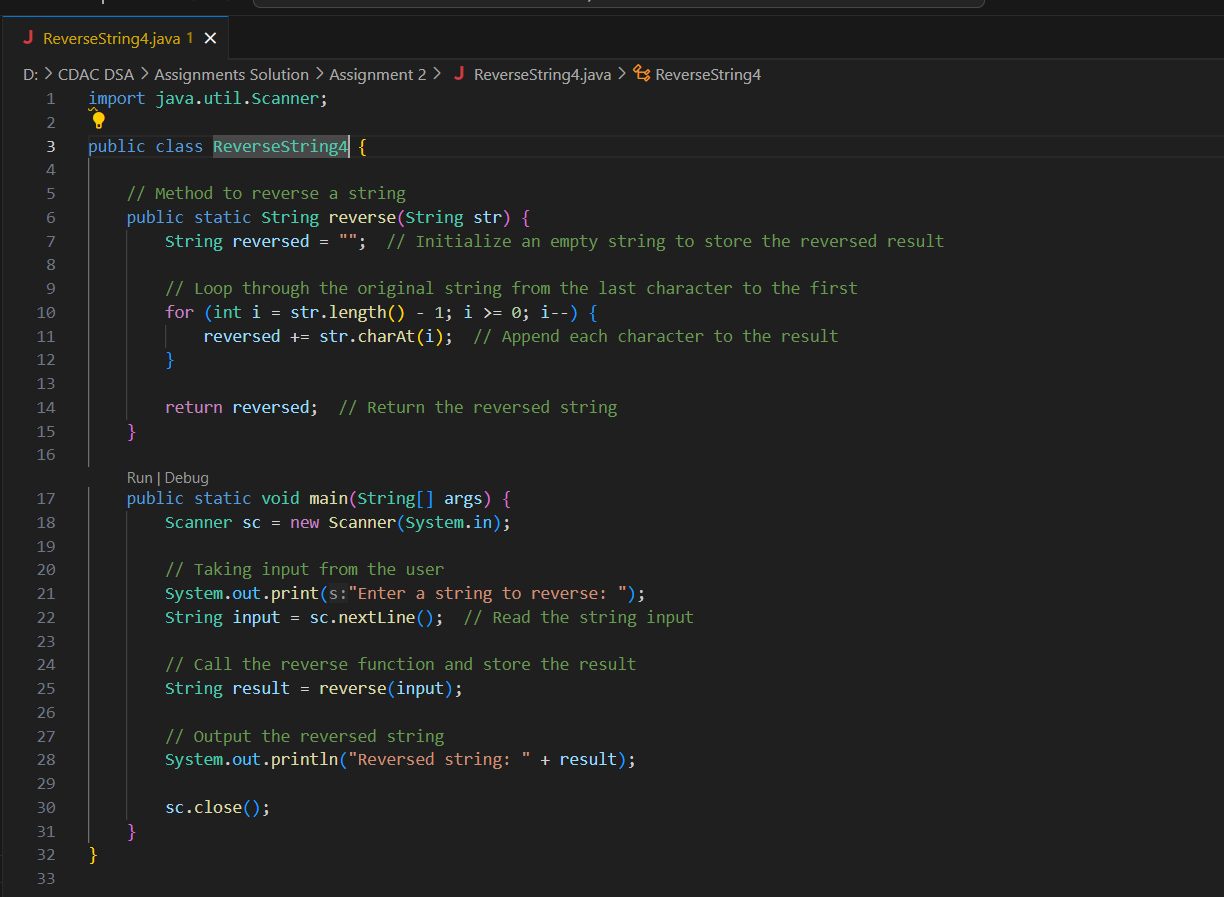
THE TIME COMPLEXITY IS O(D)

THE SPACE COMPLEXITY IS O(1)

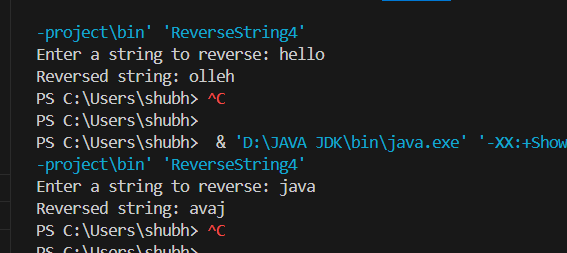
**-------------------------------------------------------------------------------------------------------------------------------**

**4. Reverse a String**

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

****

OUTPUT

****

FLOWCHART:

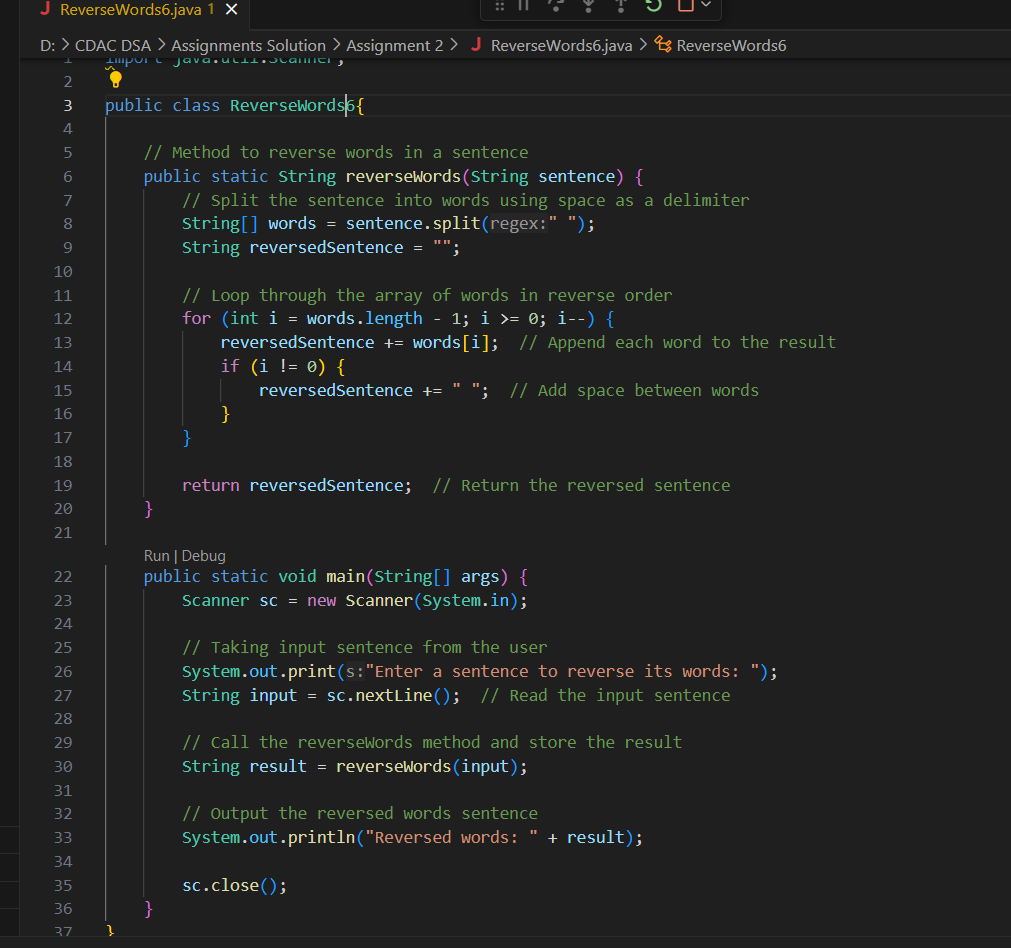
1. **Start**
2. **Input: Prompt the user to enter a string.**
3. **Initialize: Set reversed as an empty string.**
4. **Loop: From the last character of the input string to the first, append each character to reversed.**
5. **Output: Print the reversed string.**
6. **End**

THE TIME COMPLEXITY IS O(N)

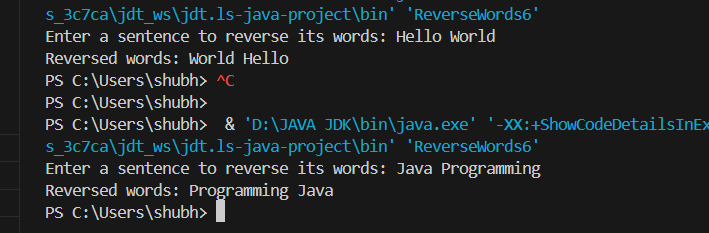
THE SPACE COMPLEXITY IS O(N)

**6. Reverse Words in a String**

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

****

**OUTPUT**

****

FLOWCHART:

1. **Start**
2. **Input: Prompt the user to enter a sentence.**
3. **Split: Split the sentence into words using space as a delimiter.**
4. **Initialize: Set an empty string reversedSentence.**
5. **Loop: Loop through the words array from the last word to the first and append each word to reversedSentence.**
6. **Output: Print the reversed sentence with words in reverse order.**
7. **End**

THE TIME COMPLEXITY IS O(N)

THE SPACE COMPLEXITY IS O(N)