Java Program to Find Longest Palindromic Substring

Problem Statement:

Write a Java program to find the longest palindromic substring in a given string without using any built-in substring or reverse functions. The program should manually check all substrings and verify whether each is a palindrome.

Example:

Input: "babad"

Output: "bab" or "aba"

Java Code:

```
public class LongestPalindromicSubstring {

    // Function to find the longest palindromic substring
    public static String longestPalindrome(String s) {
        if (s == null || s.length() == 0) {
            return "";
        }

        int n = s.length();
        int maxLength = 1; // Palindrome length minimum is 1
        int start = 0;

        // Function to check if a given substring is a palindrome
```

```
for (int i = 0; i < n; i++) {
    for (int j = i; j < n; j++) {
        boolean isPalindrome = true;

        // Check manually if the substring s[i...j] is a palindrome
        for (int k = 0; k < (j - i + 1) / 2; k++) {
            if (s.charAt(i + k) != s.charAt(j - k)) {
                isPalindrome = false;
                break;
            }
        }

        // If the current substring is a palindrome and its length is greater than maxLength
        if (isPalindrome && (j - i + 1) > maxLength) {
            start = i;
            maxLength = j - i + 1;
        }
    }
}

// Building the result string manually
String result = "";
for (int i = start; i < start + maxLength; i++) {
    result += s.charAt(i);
}
```

```
        return result;
    }

    public static void main(String[] args) {
        String input = "babad";
        System.out.println("Longest Palindromic Substring: " + longestPalindrome(input));
    }
}
```

Sample Input and Output:

Sample Input:

babad

Sample Output:

Longest Palindromic Substring: bab

(or "aba" as both are valid palindromes)