Longest Palindromic Substring

Program:

```
#include <stdio.h>
#include <string.h>

// Function to print the longest palindromic substring
void printLongestPalindromicSubstring(char *s) {
   int n = strlen(s);
   if (n == 0) {
      printf("The longest palindromic substring is: \n");
      return;
   }
   // Table to store the palindrome status
   int table[n][n];
   memset(table, 0, sizeof(table));

   int maxLength = 1; // Length of longest palindromic substring
   int start = 0; // Starting index of longest palindromic substring

   // All substrings of length 1 are palindromic
   for (int i = 0; i < n; ++i) {
      table[i][i] = 1;
   }

   // Check for substrings of length 2
   for (int i = 0; i < n - 1; ++i) {
      if (s[i] == s[i + 1]) {
         table[i][i + 1] = 1;
         start = i;
```

```
            maxLength = 2;
        }
    }

    // Check for substrings of length greater than 2
    for (int len = 3; len <= n; ++len) {
        for (int i = 0; i < n - len + 1; ++i) {
            int j = i + len - 1;
            if (s[i] == s[j] && table[i + 1][j - 1]) {
                table[i][j] = 1;
                start = i;
                maxLength = len;
            }
        }
    }

    // Print the longest palindromic substring
    printf("The longest palindromic substring is: ");
    for (int i = start; i < start + maxLength; ++i) {
        putchar(s[i]);
    }
    printf("\n");
}

int main() {
    char s[] = "babad";
    printLongestPalindromicSubstring(s);
    return 0;
}
```