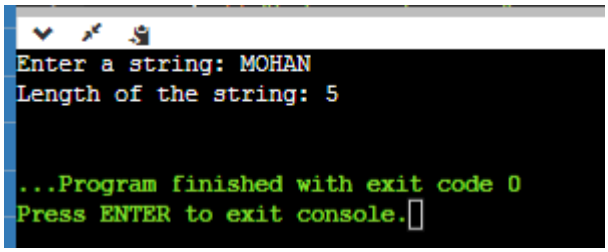
ASSIGNMENT-6--POINTER

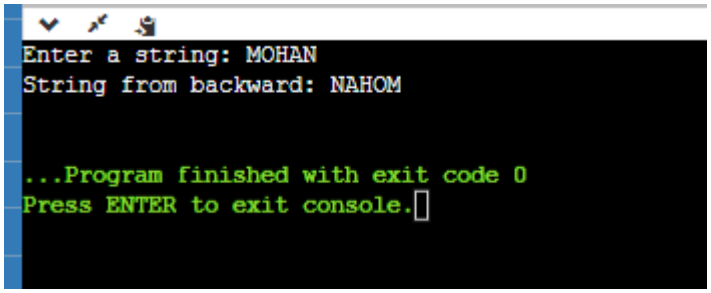
1. Write a program to find the length of string.

```
#include <iostream>
using namespace std;

int main() {
   char str[100];
   cout << "Enter a string: ";
   cin >> str;

   char *ptr = str;
   int length = 0;

   while (*ptr != '\0') {
      length++;
      ptr++;
   }

   cout << "Length of the string: " << length << endl;

   return 0;
}
```



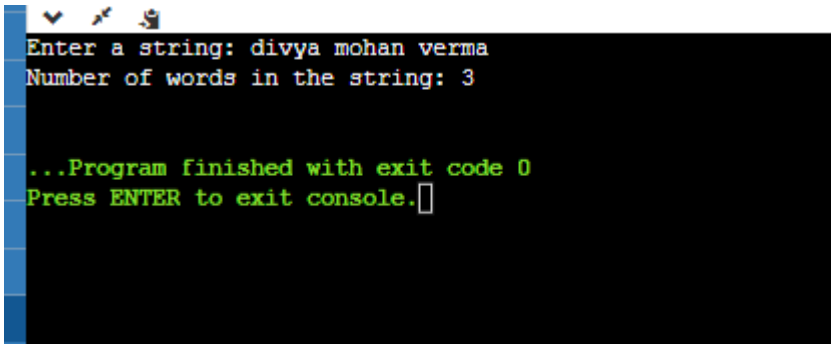
2. Write a program to display string from backward.

```
#include <iostream>
using namespace std;

int main() {
   char str[100];
   cout << "Enter a string: ";
   cin >> str;

   char *ptr = str;
   int length = 0;

   while (*ptr != '\0') {
      length++;
      ptr++;
   }

   cout << "String from backward: ";
   for (int i = length - 1; i >= 0; i--) {
      cout << str[i];
   }
   cout << endl;

   return 0;
}
```



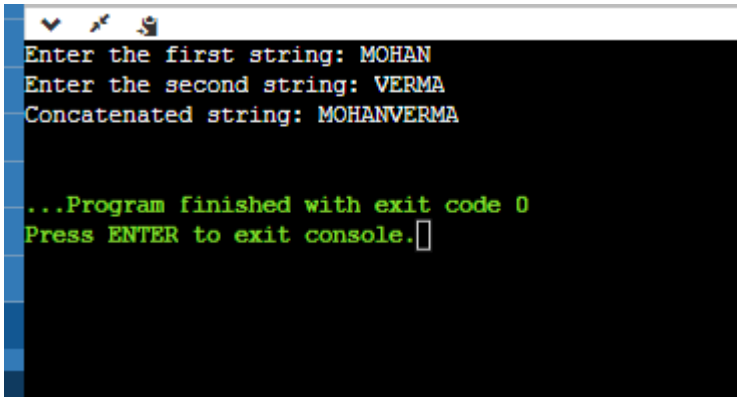
3. Write a program to count number of words in string.

```
#include <iostream>
using namespace std;

int main() {
    char str[100];
    cout << "Enter a string: ";

    char c;
    int i = 0;

    while (true) {
        c = cin.get();

        if (c == '\n' || c == '\0') {
            break;
        }

        str[i++] = c;
    }

    str[i] = '\0';

    char *ptr = str;
    int wordCount = 0;
    bool inWord = false;

    while (*ptr != '\0') {
        if (!isspace(*ptr)) {
            inWord = true;
        } else if (inWord) {
            wordCount++;
            inWord = false;
        }
        ptr++;
    }

    if (inWord) {
        wordCount++;
    }

    cout << "Number of words in the string: " << wordCount << endl;

    return 0;
}
```



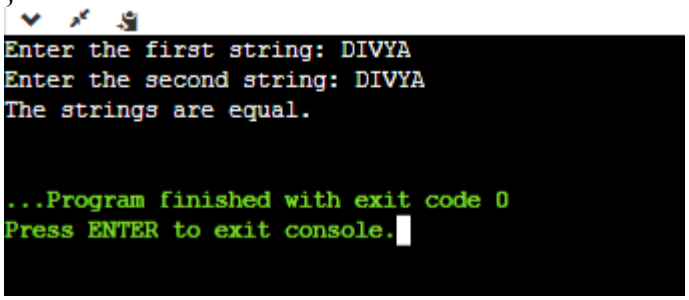
4. Write a program to concatenate one string contents to another.

```
#include <iostream>
using namespace std;

int main() {
   char str1[100], str2[100];
   cout << "Enter the first string: ";
   cin >> str1;

   cout << "Enter the second string: ";
   cin >> str2;

   char *ptr1 = str1;
   char *ptr2 = str2;

   while (*ptr1 != '\0') {
      ptr1++;
   }

   while (*ptr2 != '\0') {
      *ptr1 = *ptr2;
      ptr1++;
      ptr2++;
   }

   *ptr1 = '\0';

   cout << "Concatenated string: " << str1 << endl;

   return 0;
}
```



5. Write a program to compare two strings they are exact equal or not.

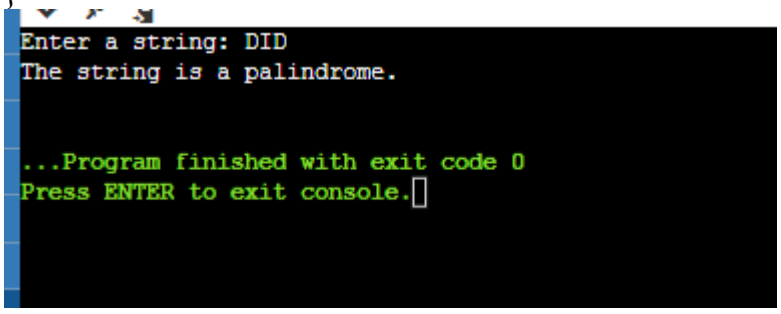
```
#include <iostream>
using namespace std;

int main() {
    char str1[100], str2[100];
    cout << "Enter the first string: ";
    cin >> str1;

    cout << "Enter the second string: ";
    cin >> str2;

    char *ptr1 = str1;
    char *ptr2 = str2;

    while (*ptr1 == *ptr2) {
        if (*ptr1 == '\0') {
            cout << "The strings are equal." << endl;
            return 0;
        }
        ptr1++;
        ptr2++;
    }

    cout << "The strings are not equal." << endl;

    return 0;
}
```



6. Write a program to check a string is palindrome or not.

```
#include <iostream>
using namespace std;

int main() {
```

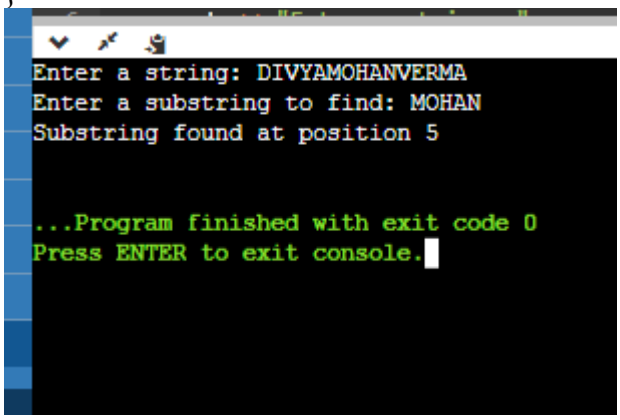
```
    char str[100];
    cout << "Enter a string: ";
    cin >> str;

    char *start = str;
    char *end = str;

    while (*end != '\0') {
        end++;
    }
    end--;

    bool isPalindrome = true;

    while (start < end) {
        if (*start != *end) {
            isPalindrome = false;
            break;
        }
        start++;
        end--;
    }

    if (isPalindrome) {
        cout << "The string is a palindrome." << endl;
    } else {
        cout << "The string is not a palindrome." << endl;
    }

    return 0;
}
```



7. Write a program to find a substring within a string. If found display its starting position.

```
#include <iostream>
using namespace std;

int main() {
    char str[100], substring[100];
    cout << "Enter a string: ";
    cin >> str;

    cout << "Enter a substring to find: ";
    cin >> substring;
```

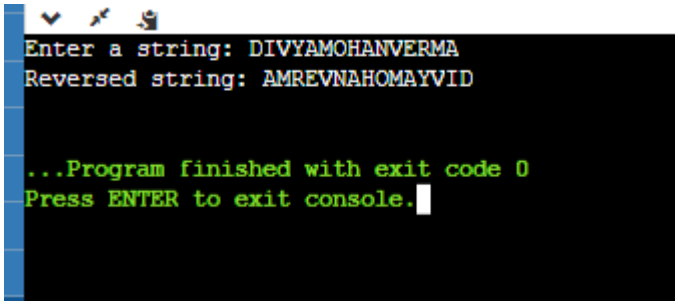
```
    char *strPtr = str;
    char *subPtr = substring;

    int position = -1;

    while (*strPtr != '\0') {
        char *tempStrPtr = strPtr;
        subPtr = substring;

        while (*tempStrPtr == *subPtr && *tempStrPtr != '\0' && *subPtr != '\0') {
            tempStrPtr++;
            subPtr++;
        }

        if (*subPtr == '\0') {
            position = strPtr - str;
            break;
        }

        strPtr++;
    }

    if (position != -1) {
        cout << "Substring found at position " << position << endl;
    } else {
        cout << "Substring not found." << endl;
    }

    return 0;
}
```



8. Write a program to reverse a string.

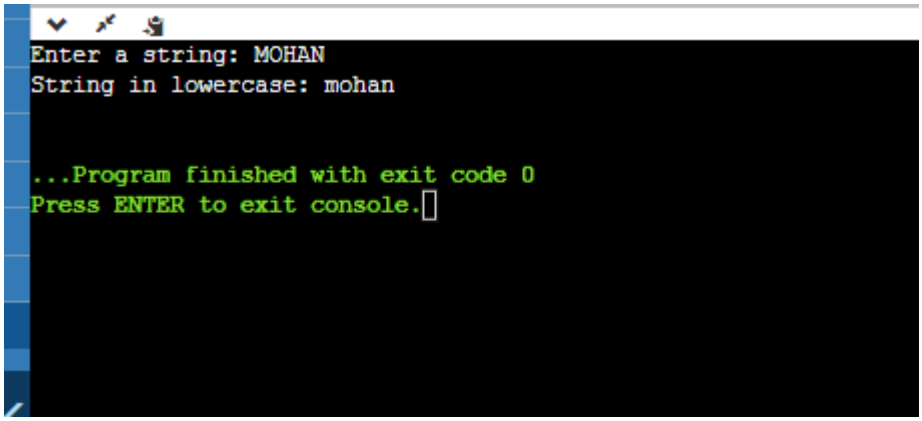
```
#include <iostream>
using namespace std;

int main() {
    char str[100];
    cout << "Enter a string: ";
    cin >> str;

    char *ptr = str;
    int length = 0;
```

```
    while (*ptr != '\0') {
        length++;
        ptr++;
    }

    cout << "Reversed string: ";
    for (int i = length - 1; i >= 0; i--) {
        cout << str[i];
    }
    cout << endl;

    return 0;
}
```



9. Write a program to convert a string in lowercase.

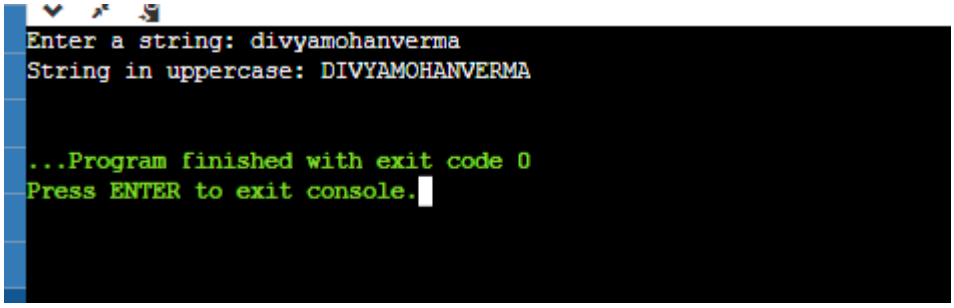
```
#include <iostream>
using namespace std;

void toLowercase(char* str) {
    while (*str != '\0') {
        if (*str >= 'A' && *str <= 'Z') {
            *str = *str + 32;
        }
        str++;
    }
}

int main() {
    char str[100];

    cout << "Enter a string: ";
    cin >> str;

    char* ptrToLower = str;
    toLowercase(ptrToLower);

    cout << "String in lowercase: " << str << endl;

    return 0;
}
```



10. Write a program to convert a string in uppercase.

```
#include <iostream>
using namespace std;

void toUppercase(char* str) {
    while (*str != '\0') {
        if (*str >= 'a' && *str <= 'z') {
            *str = *str - 32;
        }
        str++;
    }
}

int main() {
    char str[100];

    cout << "Enter a string: ";
    cin >> str;

    char* ptrToUpper = str;
    toUppercase(ptrToUpper);

    cout << "String in uppercase: " << str << endl;
```

```
    return 0;

}
```


Enter a string: divyamohanverma
String in uppercase: DIVYAMOHANVERMA
...Program finished with exit code 0
Press ENTER to exit console.