

Longest Substring without Repeating Characters

Program:

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

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

```
#define MAX_CHAR 256
```

```
int lengthOfLongestSubstring(char* s) {
```

```
    int n = strlen(s);
```

```
    int maxLen = 0; // Maximum length of substring without repeating characters
```

```
    int lastIndex[MAX_CHAR]; // Array to store the last index of each character
```

```
    int start = 0; // Start index of the current substring
```

```
    // Initialize the last index array with -1
```

```
    for (int i = 0; i < MAX_CHAR; i++) {
```

```
        lastIndex[i] = -1;
```

```
    }
```

```
    for (int i = 0; i < n; i++) {
```

```
        // If the character is found again, move the start to the right of the last occurrence
```

```
        if (lastIndex[(int)s[i]] >= start) {
```

```
            start = lastIndex[(int)s[i]] + 1;
```

```
        }
```

```
        // Update the last index of the current character
```

```
        lastIndex[(int)s[i]] = i;
```

```
        // Update the maximum length of the substring
```

```
        int currentLen = i - start + 1;
```

```
        if (currentLen > maxLen) {  
            maxLen = currentLen;  
        }  
    }  
  
    return maxLen;  
}  
  
int main() {  
    char s[] = "abcabcbb";  
    int result = lengthOfLongestSubstring(s);  
    printf("The length of the longest substring without repeating characters is: %d\n", result);  
  
    return 0;  
}
```