

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Tư, 16 tháng 4 2025, 4:26 PM
Kết thúc lúc	Thứ Tư, 16 tháng 4 2025, 4:33 PM
Thời gian thực hiện	6 phút 14 giây

Câu hỏi 1

Đúng

Đạt điểm 1,00

[Tiếng Việt]

Viết chương trình đọc vào chuỗi **s** từ bàn phím, tìm và trả về chuỗi con có tính chất đối xứng dài nhất có trong **s**.

Đầu vào:

Chuỗi **s**.

Đầu ra:

Chuỗi con có tính chất đối xứng dài nhất có trong **s**. Trong trường hợp có nhiều chuỗi con dài nhất có cùng độ dài, trả về chuỗi con có vị trí bắt đầu nhỏ nhất.

[English]

Write a program read an **s** string from keyboard, find and return the longest symmetrical sub-string of **s**.

Input:

String **s**.

Output:

The longest symmetrical sub-string from **s**. In case where there are more than one highest string, return the string with the smallest starting character index.

For example:

Test	Input	Result
1	babab	bab

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 // Hàm mở rộng từ tâm để tìm chuỗi đối xứng
6 void expandFromCenter(const string &s, int left, int right, int &start, int &maxLen)
7 {
8     // Mở rộng ra hai bên khi các ký tự đối xứng
9     while (left >= 0 && right < s.length() && s[left] == s[right]) {
10         left--;
11         right++;
12     }
13     // Tính độ dài chuỗi đối xứng
14     int length = right - left - 1;
15     // Cập nhật chuỗi đối xứng dài nhất nếu cần
16     if (length > maxLen) {
17         maxLen = length;
18         start = left + 1;
19     } else if (length == maxLen && left + 1 < start) {
20         // Nếu cùng độ dài, chọn chuỗi có vị trí bắt đầu nhỏ hơn
21         start = left + 1;
22     }
23 }
24
25 // Hàm tìm chuỗi con đối xứng dài nhất
26 string findLongestPalindrome(const string &s) {
27     int n = s.length();
28     if (n == 0) return "";
29     int start = 0, maxLen = 0;
30     for (int i = 0; i < n; i++) {
31         expandFromCenter(s, i, i, start, maxLen);
32         expandFromCenter(s, i, i + 1, start, maxLen);
33     }
34     return s.substr(start, maxLen);
35 }
```

```
28     int start = 0; // Vị trí bắt đầu của chuỗi đối xứng dài nhất
29     int maxLen = 1; // Độ dài của chuỗi đối xứng dài nhất
30     for (int i = 0; i < n; i++) {
31         // Xét trường hợp chuỗi đối xứng có độ dài lẻ (aba)
32         expandFromCenter(s, i, i, start, maxLen);
33         // Xét trường hợp chuỗi đối xứng có độ dài chẵn (abba)
34         expandFromCenter(s, i, i + 1, start, maxLen);
35     }
36
37     // Trả về chuỗi con đối xứng dài nhất
38     return s.substr(start, maxLen);
39 }
40
41 int main() {
42     string s;
43     // Đọc chuỗi đầu vào
44     getline(cin, s);
45     // Tìm và in ra chuỗi con đối xứng dài nhất
46     cout << findLongestPalindrome(s) << endl;
47
48     return 0;
49 }
```

	Test	Input	Expected	Got	
✓	1	babad	bab	bab	✓

Passed all tests! ✓

Câu hỏi 2

Đúng

Đạt điểm 1,00

[Tiếng Việt]

Viết chương trình tìm và trả về độ dài của chuỗi con dài nhất không có ký tự nào được lặp lại có trong **s**.

Đầu vào:

Chuỗi **s**.

Đầu ra:

Độ dài của chuỗi con dài nhất không có ký tự nào được lặp lại có trong **s**.

[English]

Write a program find and return the length of the longest sub-string that doesn't have any repeated character in **s**.

Input:

String **s**.

Output:

Length of the longest sub-string that doesn't have any repeated character in **s**.

For example:

Test	Input	Result
1	abcabcbb	3

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int longestNonRepeatSubString(string s) {
6     // Nếu chuỗi rỗng, trả về 0
7     if (s.empty()) return 0;
8
9     // Nếu chuỗi chỉ có 1 ký tự, trả về 1
10    if (s.length() == 1) return 1;
11
12    // Tạo mảng đánh dấu vị trí xuất hiện gần nhất của mỗi ký tự
13    // (256 là số ký tự ASCII)
14    int lastPosition[256];
15    for (int i = 0; i < 256; i++) {
16        lastPosition[i] = -1; // Khởi tạo với -1 (chưa xuất hiện)
17    }
18
19    int maxLength = 0; // Độ dài của chuỗi con dài nhất
20    int start = 0;     // Vị trí bắt đầu của chuỗi con hiện tại
21
22    // Duyệt qua từng ký tự trong chuỗi
23    for (int i = 0; i < s.length(); i++) {
24        // Nếu ký tự này đã xuất hiện sau điểm bắt đầu hiện tại
25        if (lastPosition[s[i]] >= start) {
26            // Cập nhật vị trí bắt đầu mới
27            start = lastPosition[s[i]] + 1;
28        }
29
30        // Cập nhật vị trí xuất hiện mới nhất của ký tự hiện tại
```

```
31         lastPosition[s[i]] = i;
32
33         // Cập nhật độ dài lớn nhất
34         maxLength = max(maxLength, i - start + 1);
35     }
36
37     return maxLength;
38 }
39
40 int main() {
41     string s;
42     cin >> s;
43     cout << longestNonRepeatSubString(s);
44     return 0;
45 }
```

	Test	Input	Expected	Got	
✓	1	abcabcbb	3	3	✓

Passed all tests! ✓

