

<b>Trạng thái</b>	Đã xong
<b>Bắt đầu vào lúc</b>	Thứ Sáu, 18 tháng 4 2025, 7:30 AM
<b>Kết thúc lúc</b>	Thứ Sáu, 18 tháng 4 2025, 7:34 AM
<b>Thời gian thực hiện</b>	4 phút 41 giây



## Câu hỏi 1

Đúng

Viết hàm `int find(char str[], char substr[])` để tìm vị trí của chuỗi con trong một chuỗi cho trước. Hàm trả về vị trí đầu tiên của chuỗi con trong chuỗi ban đầu. Nếu không tìm thấy chuỗi con, hàm trả về -1.

- Mô tả hàm: Hàm này thực hiện việc tìm kiếm chuỗi con trong chuỗi ban đầu và trả về vị trí đầu tiên của chuỗi con nếu được tìm thấy. Nếu không tìm thấy, hàm trả về -1.

- Đầu vào:

`char str[]`: Mảng kí tự chứa chuỗi ban đầu.

`char substr[]`: Mảng kí tự chứa chuỗi con cần tìm.

- Đầu ra:

`int`: Vị trí đầu tiên của chuỗi con trong chuỗi ban đầu. Nếu không tìm thấy, trả về -1.

- Các thư viện có thể dùng: `<iostream>`, `<cstring>`

- Note: Sinh viên không được sử dụng các từ khoá sau trong đoạn code nộp bài, kể cả trong comment: "include", "string"  
Tham số truyền vào:

Implement the `int find(char str[], char substr[])` function to find the position of a substring in a given string. The function returns the position of the first occurrence of the substring in the original string. If the substring is not found, the function returns -1.

- Function Description: This function searches for the substring in the original string and returns the position of the first occurrence of the substring if found. If not found, it returns -1.

- Input:

`char str[]`: Character array containing the original string.

`char substr[]`: Character array containing the substring to search for.

- Output:

`int`: The position of the first occurrence of the substring in the original string. If not found, returns -1.

- Available library: `<iostream>`, `<cstring>`

- Note: Your code MUST NOT contain any of the words in the list {"include", "string"} even in the comment

**For example:**

Test	Result
<pre>char str[] = "hello world"; char substr[] = "world"; cout &lt;&lt; find(str, substr);</pre>	6

**Answer:** (penalty regime: 0 %)

Reset answer

```
1 int find(char str[], char substr[]) {
2     // Tính độ dài của chuỗi ban đầu và chuỗi con
3     int strLen = 0;
4     while (str[strLen] != '\0') {
5         strLen++;
6     }
7
8     int substrLen = 0;
9     while (substr[substrLen] != '\0') {
10        substrLen++;
11    }
12
13    // Nếu chuỗi con rỗng hoặc dài hơn chuỗi ban đầu, không thể tìm thấy
14    if (substrLen == 0 || substrLen > strLen) {
15        return -1;
16    }
17
18    // Duyệt qua chuỗi ban đầu để tìm vị trí chuỗi con
19    for (int i = 0; i <= strLen - substrLen; i++) {
20        int j;
21    }
```

```

22
23 // Kiểm tra từng ký tự của chuỗi con với chuỗi ban đầu bắt đầu từ vị trí i
24 for (j = 0; j < substrLen; j++) {
25     if (str[i + j] != substr[j]) {
26         break; // Nếu có ký tự không khớp, thoát vòng lặp
27     }
28 }
29
30 // Nếu j bằng độ dài chuỗi con, tức là đã tìm thấy chuỗi con
31 if (j == substrLen) {
32     return i; // Trả về vị trí bắt đầu
33 }
34 }
35
36 // Không tìm thấy chuỗi con trong chuỗi ban đầu
37 return -1;
38 }

```

	Test	Expected	Got	
✓	char str[] = "hello world"; char substr[] = "world"; cout << find(str, substr);	6	6	✓

Passed all tests! ✓



## Câu hỏi 2

Đúng

Viết hàm void printFirstRepeatedWord(char str[]) in ra từ đầu tiên trong chuỗi bị lặp lại. Từ bị lặp lại trong một chuỗi là từ mà từ vị trí của nó trở về phía trước có xuất hiện một từ giống với nó.

- Đầu vào:

Mảng kí tự chứa chuỗi

- Đầu ra:

In ra từ đầu tiên trong chuỗi có lặp lại. Nếu không có từ nào lặp lại thì in ra "No Repetition"

- Các thư viện có thể dùng: <iostream>, <cstring>

- **Note:** Sinh viên không được sử dụng các từ khóa sau trong đoạn code nộp bài, kể cả trong comment: "include", "string". Mỗi từ trong chuỗi có không quá 30 ký tự.

Implement void printFirstRepeatedWord(char str[]) function which print the first word that repeats. In a string, a repeated word is a word that, from its position to the left, has a similar word appearing before it.

- Input:

Char array contains string

- Output:

Print the first word that has repetition. If there is no repeated word, print "No Repetition"

- Available library: <iostream>, <cstring>

- **Note:** Your code MUST NOT contain any of the words in the list {"include", "string"} even in the comment. Each word in the string has no more than 30 characters.

For example:

Test	Result
char str[] = "car taxi bike bike car taxi" ; printFirstRepeatedWord(str);	bike
char str[] = "football volleyball badminton football" ; printFirstRepeatedWord(str);	football

Answer: (penalty regime: 0 %)

Reset answer

```

1 void printFirstRepeatedWord(char str[]) {
2     // Mảng để lưu các từ đã gặp
3     char words[1000][31]; // Mỗi từ tối đa 30 ký tự + 1 ký tự kết thúc chuỗi
4     int wordCount = 0;
5
6     // Biến lưu trữ từ hiện tại đang xét
7     char currentWord[31];
8     int currentLength = 0;
9
10    // Duyệt qua từng ký tự trong chuỗi đầu vào
11    int i = 0;
12    while (str[i] != '\0') {

```



```

13 // Nếu là khoảng trắng hoặc dấu xuống dòng hoặc hết chuỗi
14 if (str[i] == ' ' || str[i] == '\n' || str[i] == '\t' || str[i] == '\0') {
15     // Kết thúc từ hiện tại nếu có
16     if (currentLength > 0) {
17         // Thêm ký tự kết thúc chuỗi
18         currentWord[currentLength] = '\0';
19
20         // Kiểm tra xem từ hiện tại có xuất hiện trước đó chưa
21         for (int j = 0; j < wordCount; j++) {
22             // So sánh từng ký tự của hai từ
23             int k = 0;
24             while (words[j][k] != '\0' && currentWord[k] != '\0' && words[j][k] == currentWord[k]) {
25                 k++;
26             }
27
28             // Nếu cả hai từ đều kết thúc cùng lúc, tức là hai từ giống nhau
29             if (words[j][k] == '\0' && currentWord[k] == '\0') {
30                 // In ra từ bị lặp lại và kết thúc hàm
31                 for (int l = 0; currentWord[l] != '\0'; l++) {
32                     cout << currentWord[l];
33                 }
34                 return;
35             }
36         }
37
38         // Nếu từ chưa xuất hiện trước đó, thêm vào danh sách các từ đã gặp
39         for (int j = 0; j <= currentLength; j++) {
40             words[wordCount][j] = currentWord[j];
41         }
42         wordCount++;
43
44         // Reset cho từ mới
45         currentLength = 0;
46     }
47 }
48 // Nếu là ký tự thường (không phải khoảng trắng)
49 else {
50     // Thêm ký tự vào từ hiện tại
51     currentWord[currentLength] = str[i];
52     currentLength++;

```

	Test	Expected	Got	
✓	char str[] = "car taxi bike bike car taxi" ; printFirstRepeatedWord(str);	bike	bike	✓

Passed all tests! ✓

## Câu hỏi 3

Đúng

Viết hàm void process(const char str[], char outstr[]) loại bỏ các khoảng trắng thừa trong chuỗi sao cho không còn hai khoảng trắng liền kề nhau và không có khoảng trắng nào ở đầu và cuối chuỗi.

- Đầu vào:

Mảng kí tự chứa chuỗi cần loại bỏ khoảng trắng

- Đầu ra:

Kết quả của bài toán được ghi vào chuỗi outstr

- Các thư viện có thể dùng: <iostream>, <string>

- **Note:** Sinh viên không được sử dụng các từ khoá sau trong đoạn code nộp bài, kể cả trong comment: "include", "string"

Implement void process(const char str[], char outstr[]) function which eliminates redundant space characters so that there are no consecutive space characters and there is no space character at the beginning/end of string.

- Input:

Char array contains the string

- Output:

The result is written to outstr

- Available library: <iostream>, <string>

- **Note:** Your code MUST NOT contain any of the words in the list {"include", "string"} even in the comment

For example:

Test	Result
<pre>const int MAX_SIZE = 100; char str[] = " abc def ghi "; char outstr[MAX_SIZE]; process(str, outstr); cout &lt;&lt; outstr;</pre>	<pre>abc def ghi</pre>

**Answer:** (penalty regime: 0 %)

Reset answer

```
1 void process(const char str[], char outstr[]) {
2     int len = 0;
3     while (str[len] != '\0') {
4         len++;
5     }
6
7     int i = 0; // Chỉ số trong chuỗi gốc
8     int j = 0; // Chỉ số trong chuỗi kết quả
9
10    // Bỏ qua khoảng trắng ở đầu chuỗi
11    while (i < len && str[i] == ' ') {
12        i++;
13    }
14
15    bool prevSpace = false; // Biến kiểm tra ký tự trước có phải là khoảng trắng không
16
17    // Xử lý chuỗi
18    while (i < len) {
19        if (str[i] == ' ') {
20            if (!prevSpace) {
21                // Nếu đây là khoảng trắng đầu tiên, thêm vào kết quả
22                outstr[j++] = ' ';
23                prevSpace = true;
24            }
25            // Bỏ qua các khoảng trắng liên tiếp
26        } else {
27            outstr[j++] = str[i];
28            prevSpace = false;
29        }
30    }
31    outstr[j] = '\0';
32}
```

```

28
29         // Thêm ký tự không phải khoảng trắng vào kết quả
30         outstr[j++] = str[i];
31         prevSpace = false;
32     }
33     i++;
34 }
35
36 // Xóa khoảng trắng ở cuối chuỗi (nếu có)
37 if (j > 0 && outstr[j-1] == ' ') {
38     j--;
39 }
40
41 // Thêm ký tự kết thúc chuỗi
42 outstr[j] = '\0';
43 }

```

	Test	Expected	Got	
✓	<pre>const int MAX_SIZE = 100; char str[] = " abc def ghi "; char outstr[MAX_SIZE]; process(str, outstr); cout &lt;&lt; outstr;</pre>	abc def ghi	abc def ghi	✓

Passed all tests! ✓