

| | |
|----------------------------|----------------------------------|
| Trạng thái | Đã xong |
| Bắt đầu vào lúc | Thứ Sáu, 3 tháng 5 2024, 1:16 PM |
| Kết thúc lúc | Thứ Sáu, 3 tháng 5 2024, 5:00 PM |
| Thời gian thực hiện | 3 giờ 43 phút |



Câu hỏi 1

Đúng

Đạt điểm 1,00

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 << find(str, substr);</pre> | 6 |

Answer: (penalty regime: 0 %)

Reset answer

```
1 // Student may implement another function as need  
2  
3 int find(char str[], char substr[]) {  
4     // TODO  
5     char* found = strstr(str, substr); // Tìm chuỗi con trong chuỗi ban đầu  
6  
7     if (found != nullptr) {  
8         return found - str; // Trả về vị trí đầu tiên của chuỗi con  
9     } else {  
10        return -1; // Nếu không tìm thấy, trả về -1  
11    }  
12 }  
13
```



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

Passed all tests! ✓



Câu hỏi 2

Đúng

Đạt điểm 1,00

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 khoá 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     char *token = strtok(str, " ");
3     int count = 0;
4     char a[100][50];
5     while(token != NULL){
6         strcpy(a[count++], token);
7         token = strtok(NULL, " ");
8     }
9     for (int i = 1; i < count; i++) {
10        for (int j = i - 1; j >= 0; j--) {
11            if (strcmp(a[i], a[j]) == 0) {
12                cout << a[i];
13                return;
14            }
15        }
16    }
17    cout << "No Repetition";
18 }
```



| | Test | Expected | Got | |
|---|--|----------|------|---|
| ✓ | <pre>char str[] = "car taxi bike bike car taxi" ; printFirstRepeatedWord(str);</pre> | bike | bike | ✓ |

Passed all tests! ✓



Câu hỏi 3

Đúng

Đạt điểm 1,00

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>, <cstring>

- **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>, <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>const int MAX_SIZE = 100; char str[] = " abc def ghi "; char outstr[MAX_SIZE]; process(str, outstr); cout << outstr;</pre> | abc def ghi |

Answer: (penalty regime: 0 %)

Reset answer

```
1 // Student may implement another function as need
2
3 void process(char str[], char outstr[]) {
4     outstr[0] = '\0';
5     char *token = strtok(str, " ");
6
7     while(token != NULL){
8         strcat(outstr, token);
9         strcat(outstr, " ");
10        token = strtok(NULL, " ");
11    }
12 }
```



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

Passed all tests! ✓

