

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Bảy, 18 tháng 5 2024, 10:18 AM
Kết thúc lúc	Thứ Bảy, 18 tháng 5 2024, 10:47 AM
Thời gian thực hiện	28 phút 46 giây



Câu hỏi 1

Đúng

Đạt điểm 1,00

[Tiếng Việt]

Hoàn thành hàm `recursiveSum` sử dụng đệ quy để tính và trả về tổng các phần tử trong một mảng

Đầu vào:

- `int arr[]`: Mảng chứa các số nguyên cần tính tổng.
- `int size`: Số lượng phần tử trong mảng.

Đầu ra: `int` - Kết quả là tổng các phần tử trong mảng cho trước.

*Lưu ý: Sinh viên không được sử dụng các từ khoá sau, kể cả trong tên biến và comment: **include, for, while, goto***

[English]

Complete the function `recursiveSum` using recursion to calculate and return the sum of all elements in an array.

Inputs:

- `int arr[]`: An array containing integers to be summed.
- `int size`: The size of the array.

Output: `int` - the result is the sum of the elements in the given array.

*Note: Students are not allowed to use the following keywords, including in variable names or comments: **include, for, while, goto**.*

For example:

Test	Result
<pre>int arr[] = {1,2,3,4,5}; int size = 5; cout << recursiveSum(arr, size);</pre>	15



Test	Result
<pre>int arr[] = {10, -10, 20, -20, 30, -30}; int size = sizeof(arr) / sizeof(int); cout << recursiveSum(arr, size);</pre>	0

Answer: (penalty regime: 0 %)

Reset answer

```
1 int recursiveSum(int arr[], int size) {
2     // TODO
3     if(size == 0) return 0;
4     return arr[size - 1] + recursiveSum(arr, size - 1);
5 }
```



Passed all tests! ✓

Câu hỏi 2

Đúng

Đạt điểm 1,00

[Tiếng Việt]

Cho một số thập phân dương làm đầu vào, chúng ta cần triển khai hàm

```
long int decimalToBinary(int decimal_number){}
```

để chuyển đổi số thập phân dương đã cho thành số nhị phân tương đương.

Xin lưu ý rằng bạn không thể sử dụng từ khóa for, while, goto (ngay cả trong tên biến, comment).

Đối với bài tập này, chúng ta có #include <iostream> và sử dụng namespace std;

[English]

Given a positive decimal number as input, we need to implement function

```
long int decimalToBinary(int decimal_number){}
```

to convert the given positive decimal number into equivalent binary number.

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:



Test	Result
cout << decimalToBinary(20);	10100

Answer: (penalty regime: 0 %)

Reset answer

```
1 long int decimalToBinary(int decimal_number)
2 {
3     if( decimal_number == 0 || decimal_number == 1)
4         return decimal_number;
5     return decimalToBinary(decimal_number / 2) * 10 + decimal_number % 2;
6
7 }
```



	Test	Expected	Got	
✓	cout << decimalToBinary(20);	10100	10100	✓

	Test	Expected	Got	
✓	cout << decimalToBinary(192);	11000000	11000000	✓

Passed all tests! ✓



Câu hỏi 3

Đúng

Đạt điểm 1,00

[Tiếng Việt]

Palindrome là một chuỗi (string) mà nếu đảo ngược thứ tự các ký tự trong chuỗi ta vẫn có được chuỗi cũ. Ví dụ : "ABCBA", "RADAR", "otto", "i am ma i", "C".

Hàm **palindrome** kiểm tra một chuỗi có là Palindrome đã được cho phía dưới.

a/ Viết một chương trình C++ sử dụng vòng lặp while đọc vào nhiều chuỗi và gọi hàm trên để kiểm tra các chuỗi đọc vào có phải là palindrome hay không. Vòng lặp sẽ chấm dứt khi người sử dụng đọc vào một chuỗi bắt đầu bằng *.

b/ Viết hàm palindromeRecursion sử dụng kỹ thuật gọi đệ quy để kiểm tra một chuỗi có phải là palindrome hay không.

Đầu vào:

Các chuỗi ký tự s có độ dài không quá 1000 ký tự

Đầu ra:

Mỗi dòng trả về giá trị của hàm palindrome và palindromRecursion (xem ví dụ để biết thêm chi tiết)

[English]

Palindrome is a string (string) that if you reverse the order of characters in the string, we still get the old string. For example: "ABCBA", "RADAR", "otto", "i am ma i", "C".

The **palindrome** function checks if a string is the given palindrome.

a/ Write a C++ program that uses a while loop to read multiple strings and call the above function to check whether the input strings are palindrome or not. The loop will terminate when the user reads into a string starting with *.

b/ Write a function palindromeRecursion using recursive calling technique to check a string is palindrome or not.

Input:

Character strings s with a length of no more than 1000 characters

Output:

Each line returns the value of the `palindrome` and `palindromeRecursion` functions (see example for more details)

Template of full code:

```
#include<iostream>
#include<string>
using namespace std;
/* END of library */
bool palindrome(string strg);
bool palindromeRecursion(string s)
{
    // BEGIN YOUR IMPLEMENTATION [1]
    // TODO

    // END YOUR IMPLEMENTATION [1]
}
int main()
{
    hiddenCheck();
    // BEGIN YOUR IMPLEMENTATION [2]
    // TODO

    // END YOUR IMPLEMENTATION [2]
    return 0;
}
```

**For example:**

Test	Input	Result
1	abccba abc a *	true true false false true true

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

Reset answer

```

1
2 string boolToString(bool value) {
3     return value ? "true" : "false";
4 }
5 // Hàm kiểm tra chuỗi Palindrome sử dụng đệ quy
6 bool palindromeRecursion(string s, int left, int right) {
7     // Trường hợp cơ sở
8     if (left >= right) return true;
9     // Nếu ký tự đầu và cuối không giống nhau, không phải là Palindrome
10    if (s[left] != s[right]) return false;
11    // Gọi đệ quy cho chuỗi con
12    return palindromeRecursion(s, left + 1, right - 1);
13 }
14
15 // Hàm gọi đệ quy với chỉ số đầu và cuối
16 bool palindromeRecursion(string s) {
17     return palindromeRecursion(s, 0, s.length() - 1);
18 }
19
20 int main() {
21     string str;
22     while (true) {
23         cin >> str;
24         if (str[0] == '*') break;
25         cout << boolToString(palindrome(str)) << " ";
26         cout << boolToString(palindromeRecursion(str)) << endl;
27     }
28     return 0;

```

29 }
30



	Test	Input	Expected	Got	
✓	1	abccba abc a *	true true false false true true	true true false false true true	✓
✓	8	1 *	true true	true true	✓

Passed all tests! ✓

