ĐẠI HỌC BÁCH KHOA HÀ NỘI



TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**\*\*\*\*\*\*\*\*\*\***

A picture containing logo

Description automatically generated

**BÁO CÁO GIỮA KÌ**

**HỌC PHẦN: THỰC HÀNH KIẾN TRÚC MÁY TÍNH**

**Giáo viên hướng dẫn: Nguyễn Thị Thanh Nga**

**Nhóm số :3**

**Mã lớp:130997**

**Họ và tên:Phạm Minh Quang**

**MSSV:20205119**

***Năm học 2021 - 2022***

Contents

[1. Đề bài: 3](#_Toc108985995)

[2. Phân tích cách thực hiện: 3](#_Toc108985996)

[3. Khai báo biến(.data) 3](#_Toc108985997)

[4. Ý nghĩa các thanh ghi 3](#_Toc108985998)

## Đề bài:

- Create a program to input a text line from the keyboard and test if it is a palindrome. For example: “abc121cba” is a palindrome. Store all palindromes which the user typed into the memory, to make sure that the user doesn’t duplicate palindromes.

## Phân tích cách thực hiện:

-Đầu tiên cấp phát 1 địa chỉ để lưu xâu kí tự mà người dung nhập vào. Tiếp đó mở hộp thoại dialog cho phép người dung nhập vào xâu kí tự rồi chuyển tới hàm **isValidString** để kiểm tra xem chuỗi nhập vào có hợp lệ hay không dựa vào status của dialog inputString. Sau đó chuyển tới hàm **getLength**, để tìm độ dài của xâu kí tự vừa nhập và chỉ số cuối cùng của xâu.( Ví dụ xâu có độ dài là 20, thì kí tự cuối cùng sẽ có chỉ số là 9 do chỉ số bắt đầu là 0).

-Hàm **handleCheckIsStored** kiểm tra xem xâu vừa nhập có nằm trong mảng listString không ( mảng listString lưu trữ các chuỗi palindrome mà người dùng nhập vào trước đó). Nếu có thì hiển thị dialog đây là chuỗi palindrome, và hỏi xem người dung có muốn nhập tiếp hay không. Nếu không nằm trong **listString** thì đi tới hàm checkPalindrome.

Hàm **checkPalindrome** kiểm tra xem chuỗi vừa nhập có đối xứng hay không. Nếu không đối xứng thì hiện ra dialog thông báo kết quả và hỏi người dung có muốn nhập tiếp không. Còn nếu đây là chuỗi Palindrome thì chuyển tới hàm **handleStoreString**.

Hàm **handleStoreString** làm nhiệm vụ lưu chuỗi palindrome người dung vừa nhập vào bộ nhớ. Nếu bộ nhớ đầy( **listString** đã dung hết) thì không lưu vào nữa. Cuối cùng hỏi người dung muốn nhập tiếp hay không

## Khai báo biến(.data)

**inputString**: chuỗi kí tự người dung nhập vào

**listString**: mảng lưu các chuỗi palindrome mà người dung nhập vào.

RESULT, INPUT\_STRING\_MESSAGE, NOT\_IS\_PALINDROME\_MESSAGE, IS\_PALINDROME\_MESSAGE, STRING\_IS\_STORED\_MESSAGE, FULL\_MEMORY\_MESSAGE, CONFIRM\_MESSAGE, FINISH\_STORING\_STRING\_MESSAGE, IS\_INVALID\_STRING\_MESSAGE, ERROR là các message thông báo.

## Ý nghĩa các thanh ghi

$s0: Địa chỉ của xâu nhập vào ( **inputString**).

$s1:Độ dài của xâu nhập vào.

$s2: Địa chỉ của mảng lưu các xâu palindrome mà người dung đã nhập( recentString)

Ngoài 3 thanh ghi trên, các thanh ghi còn lại không mang ý nghĩa cố định nào trong cả chương trình. Trong mỗi hàm chúng sẽ có ý nghĩa khác nhau.

Hàm **getLength**:

$t1: chỉ số i, dùng để chạy đến cuối chuỗi inputString.

$t2: địa chỉ của phần tử thứ I trong xâu inputString.

$t3: giá trị của phần tử thứ I trong xâu inputString.

$s7: giá trị của kí tự ‘\n’ trong bảng ASCII(10).

Hàm **isValidString:**

$a1:là trạng thái của dialog inputString. Dùng để kiểm tra chuỗi có hợp lệ không.

Hàm **handleCheckIsStored:**

$t1: chỉ số của kí tự cuối cùng của inputString (chỉ số được tính từ 0)

$t2: chỉ số j chạy từ đầu đến cuối mảng listString cho đến hết chuỗi cuối cùng.

$t3: chỉ số k, chạy từ kí tự đầu đến cuối cùng của inputString

$t5: dòng 77 : địa chỉ của phần tử thứ j trong listString

Dòng 78: giá trị của phần tử thứ j trong listString

$t6: giá trị của phần tử thứ j trong listString( listString[j])

$t7: giá trị của phần tử thứ k trong inputString (inputString[k])

Hàm **checkPalindrome**:

$t1: chỉ số i chạy từ đầu chuỗi inputString xuống cuối.

$t2: chỉ số j chạy từ cuối chuỗi inputString lên đầu.

$t3: giá trị của phần tử thứ i trong chuỗi inputString( inputString[i])

$t4: giá trị của phần tử thứ j trong chuỗi inputString( inputString[j])

Hàm **handleStoreString:**

$t1: chỉ số cuối cùng của xâu inputString (chỉ số này bắt đầu từ 0)

$t2: chỉ số j, chạy từ đầu đến cuối mảng listString (bước nhảy 50)

$t6: chỉ số k, chạy từ đầu đến cuối xâu inputString

$t3: giá trị của phần tử thứ j trong listString

$t7: địa chỉ của phần tử thứ j+k trong listString

$t8: giá trị của phần tử thứ k trong inputString