//MATEO S PEREZ OLMEDO

//CECS 282 LAB 4

#include <string>

#include <iostream>

#include <bits/stdc++.h>

#include <cctype>

#include <cstring>

using namespace std;

//declaring our functiona

void transform(char \*raw, char \*testStr);

bool testPal(char \*str);

int main(){

bool check = true;

char yesOrNo;

string word;

int size = 0;

cout << "Please enter a sentence to check if it is a palindrome:" << endl;

//allows us to grab the whole input rather

//than just one word

getline(cin, word);

//store the input into an cstring

const char \*str1 = word.c\_str();

//find the size of the cstring and create a new

//cstring that is not constant to modify and

//another to put the final transformed cstring into

size = strlen(str1);

char \*str2 = new char[size];

char \*strFinal = new char[size];

strcpy(str2, str1);

transform(str2,strFinal);

//check if the string is a palindrome and act accordingly

if(testPal(strFinal)){

cout << "\"" << word << "\" is a palindrome" << endl;

}else{

cout << "\"" << word << "\" is not a palindrome" << endl;

}

//if the user says they want to try another phrase use recursion

//to

cout << "Would you like to try another sentence? [Y] or [N]" << endl;

cin >> yesOrNo;

cin.ignore();

if(yesOrNo == 'Y' || yesOrNo == 'y')

{

main();

}

return 0;

}

void transform(char \*raw, char \*testStr){

//keep track of the final test string pointer location

int testCounter = 0;

for(int i = 0;i<strlen(raw);i++){

//if the char is a letter check if it is upper

//if not change it to lower and pass it to the

//final CString

if (isalpha(\*(raw+i))){

if (isupper(\*(raw+i))){

\*(raw+i) = tolower(\*(raw+i));

\*(testStr + testCounter) = \*(raw+i);

}else{

\*(testStr + testCounter) = \*(raw+i);

}

testCounter++;

}

}

//add the null character at the end of the cString

\*(testStr + testCounter) = '\0';

}

bool testPal(char \*str){

//create pointers that show where the first and last

//chars will be

char \*first = str;

char \*last = str + strlen(str) - 1;

//if the first and last equal we keep converging towards

//the middle to see if the string is a palindrome

for(int i = 0; i < strlen(str); i++){

if (\*first == \*last){

first = first + 1;

last = last - 1;

}

//if at any point the letters don't match it is not a

//palindrome and we return a false.

else{

return false;

}

}

return true;

}