//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// Programmer: Chris Dang Class: CSCI 1107

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <iostream>

#include <string>

#include <cctype>

#include <cstdlib>

using namespace std;

string strToLower(string str);

string strToUpper(string str);

string replace\_all(string str, string substring, string newSubstring);

int stringToInt (string str) ;

int main() {

string str;

string testNumber = "-12045" ;

string test = "HelloMikeMyNameIsMikeMike" ;

cout << "Please enter a word to conver to lowercase: ";

cin >> str;

cout << strToLower(str) << endl << endl ;

cout << "Please enter a word to conver to uppercase: ";

cin >> str;

cout << strToUpper(str) << endl << endl ;

cout << test << " is a string that will be entered into Replace\_all.\n\n"

<< "Replace\_all will now replace all cases of Mike with Riley:\n" ;

cout << endl << replace\_all(test, "Mike" , "Riley") << endl << endl ;

cout << testNumber << " is a string that will be converted to an int.\n" ;

cout << stringToInt(testNumber) << endl << endl ;

return 0 ;

}// end main

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function: strToLower

//

// Pre-Condition: C++ string class must be included and passed in as argument.

//

// Description: Takes all characters and switches them to lower case.

//

// Post: Returns a string with all lower case.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

string strToLower(string str){

int strLength = str.length();

for (int i = 0; i < strLength; i++)

str[i] = tolower(str.at(i));

return str;

}// end strToLower

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function: strToUpper

//

// Pre-Condition: C++ string class must be included and passed in as argument.

//

// Description: Takes all characters and switches them to upper case.

//

// Post: Returns a string with all upper case.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

string strToUpper(string str){

int strLength = str.length();

for (int i = 0; i < strLength; i++)

str[i] = toupper(str.at(i));

return str;

}// end strToLower

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function: replace\_all

//

// Pre-Condition: C++ string class must be included and passed in as arguments.

// Argument str is string that is to searched and modified.

// Argument subString is to be searched for and will be replaced by newSubstring

//

// Description: Replaces all cases of subString (that are found within str)

// with newSubstring

//

// Post: Returns a copy of str that is modified with newSubstring

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

string replace\_all(string str, string subString, string newSubstring){

int length = str.length() ;

int sub = subString.length() ;

for (int i = 0, position; i < length; i++){

position = str.find(subString, 0) ;

while (position > 0) { //cut subString out and replace with newSubstring

str.erase(position, sub) ;

str.insert(position, newSubstring) ;

position -= position ;

}// end while

}// end for

return str ;

} // end replace\_all

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function: stringToInt

//

// Pre-Condition: C++ string class must be included and passed in as arguments.

//

// Description: Takes in a string, which may only include

// characters '+', '-', and integer values and converts string to int.

//

// Post: Returns an integer value of the string that was entered.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int stringToInt (string str) {

bool flag = true; // used for error checking

int newInt ; // if string has been checked and cleared,

// the string will be converted and assigned to this int

for (int i = 0 ; flag == true && i < str.length() ; str[i++]){

flag = isdigit(str[i]) ;

//allows + or - to pass as they may be a negative/positive indicator

if (str[i] == '+' || str[i] == '-' )

flag = true ;

//Error checking. If a character is not supposed to be part of an integer,

//the function will end and alert the user.

if (flag == false){

cout << "\n\nString contains a character that is not part of an integer.\n\n" ;

return 0;

}// end if

}// end for

newInt = atoi(str.c\_str()) ;

return newInt ;

}// end stringToInt

