CSC 150 LAB 4-2 RECURSIVE FUNCTIONS

PROF GODFREY C. MUGANDA
DEPT. OF COMPUTER SCIENCE
NORTH CENTRAL COLLEGE

1. Overview

The purpose of this lab is give you some practice with recursive problem-solving techniques.

Your are going to write a number of recursive methods that will be called in the `main` function.

You will need

(1) A function

```
void output(vector<int> & vec)
```

to output a vector.

(2) A recursive function

```
int freq(string str, int last_pos, char ch)
```

that returns the number of times the character `ch` occurs in positions 0. . . `last_pos` of the string `str`.

(3) A recursive function

```
bool ispalindrome(string str, int lower, int upper)
```

that determines whether the portion of the string `str` in positions `lower` . . . `upper` is a palindrome.

(4) A recursive function

```
void factors(int number, int max_factor,
             vector<int>& divisors)
```

that adds all factors of `number` that are less or equal to `max_factor` to a vector `divisors`.

2. How the Program Will Work

We will put these functions to use in a program that asks the user to enter a string, a character, and a number. The program will call `freq` to determine the frequency of the character in the string, then it will call `ispalindrome` to determine whether the string is a palindrome, and then it will call `factors` to compute a vector of all factors of the number.

Here is a sample run:

```
Enter a sentence: North Central College
Enter a character: e
Enter a number: 38

The number of times e occurs in North Central College is 3

North Central College is not a palindrome

The factors of 38 are [38, 19, 2, 1]
```

Here is a second sample run:

```
Enter a sentence: madam
Enter a character: a
Enter a number: 109

The number of times a occurs in madam is 2

madam is a palindrome

The factors of 109 are [109, 1]
```

3. Project Shell

Start with the following shell, where the code in main has already been written.

```
#include <iostream>
#include <string>
#include <vector>

using namespace std;

// Output a vector as in previous lab,
// Does not have to be recursive
void output(vector<int> & vec)
{

}


// Recursive frequency of characters
// This function returns the number of times ch occurs in the
// in positions 0..last (inclusive on both ends) in the string str
int freq(string str, int last_pos, char ch)
{
return 0;
}

// Recursive palindrome checking
//This function checks whether the  characters in positions
// lower ... upper in the string str form a palindrome.
// Should return true if they form a palindrome and false otherwise.
bool ispalindrome(string str, int lower, int upper)
```

```
{
return false;
}


// Recursive function
// Adds all factors of number that are less or equal to max,
// to the vector divisors
void factors(int number, int max_factor, vector<int>& divisors)
{

}

int main()
{
// Read in a string, a character, and a number
string str;
char ch;
int number;
cout << "Enter a sentence: ";
getline(cin, str);
cout << "Enter a character: ";
cin >> ch;
cout << "Enter a number: ";
cin >> number;

// Test the recursive freq function
cout << "\nThe number of times " << ch << " occurs in "
 << str << " is " << freq(str, str.length() - 1, ch);
cout << endl << endl;

// Test the recursive palindrome function
if (ispalindrome(str, 0, str.length() - 1))
{
cout << str << " is a palindrome\n";
}
else
{
cout << str << " is not a palindrome\n";
}

// Test the recursive factors function to
// Add the factors of the number to the vector

// Create vector to hold factors of number and call factors
vector<int> number_factors;
factors(number, number, number_factors);
cout << "\nThe factors of " << number << " are ";
output(number_factors);

cout << endl;
}
```

4. Complete The Assignment

Write all of the above functions one at a time, testing each function as you complete it.

This assignment is due Tuesday of Week 5.