2021 Winter CIS200 – Lab 6

Release date: March 1, 2021

Due date: March 8, 2021

Student Name:

# **Question 1**

Write a recursive function defined by the following recursive formula:

foo (Y, X) =

Y if X = 1

1 if X = Y

(foo ( Y-1, X-1) + 4\* foo ( Y-1, X)) if Y > X > 1

Write a driver to print out the value for foo (5, 3) and foo (6, 5).

In addition, print out the total number of recursive function calls in each case.

**Question 2**

Implement a recursive function that conducts a binary search with the following interface:

bool BinarySearch (int info[ ], int x, int fromLoc, int toLoc, int &step);

// info[fromLoc, toLoc] stores an ordered list

// step: the number of search step of this binary search

// x -- a search item

// if x is in the list, return true; otherwise, return false

Test cases:

#define LEN 10000

int info[LEN];

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

info[i] = i;

int step;

cout << BinarySearch(info, 997, 0, LEN-1, step) << endl;

cout << “Binary search steps: “ << step << endl;

cout << BinarySearch(info, 20000, 0, LEN-1, step) << endl;

cout << “Binary search steps: “ << step << endl;

**Question 3**

Implement a function that conducts a linear search with the following interface:

bool LinearSearch (int info[ ], int x, int fromLoc, int toLoc, int &step);

// info[fromLoc, toLoc] stores an ordered list

// step: the number of search step of this binary search

// x -- a search item

// if x is in the list, return true; otherwise, return false

Test cases:

#define LEN 10000

int info[LEN];

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

info[i] = i;

int step;

cout << LinearSearch(info, 997, 0, LEN-1, step) << endl;

cout << “Linear search steps: “ << step << endl;

cout << LinearSearch(info, 20000, 0, LEN-1, step) << endl;

cout << “Linear search steps: “ << step << endl;

**Submission of Your Work:**

The Word document should contain the following information

* Your name
* Machine type
* Compiler type
* Description of your code design and implementation
* Inclusion of your source
* Test case design and screen shots