CSCI 104L Review: Streams and Recursion

Example usage of file streams; you’ll need to include `fstream`.

```cpp
ofstream myFile;
myFile.open("games.txt");
myFile << "1: Minecraft" << endl;
myFile.close();
ifstream myFile2;
string line;
myFile2.open("games.txt");
getline(myFile2, line);
cout << line;
myFile2.close();
```

Example usage of string streams; you’ll need to include `sstream`

```cpp
int first;
string second;
char third;
string fourth;
stringstream ss;
ss << "1: Minecraft";
ss >> first;
ss >> second;
ss >> third;
ss >> fourth;
cout << first << endl << second << endl << third << endl << fourth;
```

**Recursive** means “defined in terms of itself”.

```cpp
int iterativeFactorial(int n) {
    int p=1;
    for (int i=1; i <= n; i++)
        p *= i;
    return p;
}
```

```cpp
int recursiveFactorial (int n) {
    if (n==1) return 1;
    else return n*recursiveFactorial(n-1);
}
```

What will happen if you run the following function?

```cpp
int UCLAfactorial (int n) {
    if (n==1) return 1;
    else return UCLAfactorial(n);
}
```
What will happen if you run the following function?

```c
int BruinFactorial(int n) {
    return n*BruinFactorial(n-1);
}
```

A recursive Binary Search function:
//t = target element. b = array.
//lo = index of first element in array (pass in 0 when you call this function).
//hi = index of last element in array (initially array length - 1).
int binarySearch(int t, int *b, int lo, int hi) {
    if (hi < lo) return -1; // nothing to search, it's not in the array.
    else {
        int mid = (hi+lo)/2; // the middle of the array, rounded down.
        if (t == b[mid]) return mid; // found it!
        else if (t < b[mid]) return binarySearch(t, b, lo, mid-1); // search left.
        else return binarySearch(t, b, mid+1, hi); // search right.
    }
}

An iterative Binary Search function:
//t = target element. b = array. len = length of array.
int iterativeBinarySearch(int t, int *b, int len) {
    int lo = 0, hi = len - 1, mid;
    while (lo <= hi) {
        mid = (hi+lo)/2;
        if (b[mid] == t) return mid;
        else if (t < b[mid]) hi = mid - 1;
        else lo = mid + 1;
    }
    return -1;
}

**Recursive Definitions**

You can define other things recursively, not just functions.

- A string of lower-case letters is either: (1) the empty string, or (2) a letter ‘a’-‘z’ followed by a string of lower-case letters.
- A non-negative integer is either: (1) the number 0, or (2) n+1, where n is a non-negative integer.
- A palindrome is either: (1) the empty string, or (2) a single letter ‘a’-‘z’, or (3) a string xPx, where x is a single letter ‘a’-‘z’, and P is a palindrome.
- A simple algebraic expression is either:
  1. A number.
  2. A variable.
  3. (A+B), where A and B are simple algebraic expressions.
  4. (A*B), where A and B are simple algebraic expressions.