

## Homework Assignment #1

Assigned: Jan 19, 2012

Due: Jan 27, 2012, by 10pm

### Theme: Recursion

**Problem #1:** Carrano (7th ed.), problem 2.14

**Problem #2:** Carrano (7th ed.), problem 2.20

**Problem #3:**

(a) Carrano (7th ed.), problem 2.25

(b)\* Is there a guarantee that Ackermann's function will always yield a result – regardless of the values of  $m$  and  $n$  – after a finite number of calculations? In other words, are there values of  $m$  and  $n$ , for which the recursion will never end, hence the function will never reach a final result, even if we assume the availability of infinite memory? Briefly explain your thinking.

**Problem #4:**

Consider a sequence of single-digit natural numbers written out on a line, e.g.,

3, 4, 4, 2

A subsequent line of numbers is defined in terms of its immediately preceding line by the following recursive rule:

“Each line  $n$  (for  $n > 1$ ) describes the previous line ( $n-1$ ) as a sequence of pairs of numbers, as follows. Line  $n$  starts with a pair of numbers  $x, y$  describing how many times ( $x$ ) the leading number ( $y$ ) from line ( $n-1$ ) occurs at the start of line ( $n-1$ ). Then, the same procedure is applied on the remainder of line ( $n-1$ ) – i.e., after the first  $x$  characters are ignored/removed from it – and the result is appended to line  $n$ .”

For example, the very next line after the sample one above would be

1, 3, 2, 4, 1, 2,

representing that the former line consists of 1 digit 3, followed by 2 digits 4, followed by 1 digit 2.

(a) Does a sequence of numbers exist for a first line such that:

(a)(1) the second line consists of only different numbers?

(a)(2) some subsequent line is shorter (has fewer numbers) than its immediately preceding line?

(a)(3) some subsequent line consists of nothing but repetitions, more than 2, of one number?

(a)(4) some subsequent line is an exact copy of its preceding line?

(a)(5) the third line consists solely of odd numbers?

(a)(6)\* the fourth line is a palindrome (i.e., it reads the same left to right or right to left)

For each of the sub-questions (a)(1) – (a)(6), if such a sequence exists, please provide an example; if not, explain why not.

(b) Prove that every line (except possibly the first) consists of an even count of numbers.

(c) What is the maximum possible length (count of numbers) in line  $n$ , if the first line is of length  $k$ ?

(d) What is the largest number of equal consecutive numbers in lines after the first?

(e)\* Does an unreachable sequence of numbers exist for some line after the first?

(f) The dependence of line  $n$  on line ( $n-1$ ) is expressed through the recursive rule in the text of the problem. Express, in a similarly precise way (in English), a recursive rule that defines the opposite dependence: that of line ( $n-1$ ) on line  $n$ .

(g)\* Propose an interesting (to you, personally) question relating to the idea of this problem, and briefly sketch the solution/answer to your question.

(h) Write functions in C++ (and a brief sample program to test them) that, given one line of such a sequence of numbers (e.g., provided by the user as command-line arguments), computes the immediate next line, as well as the immediate previous line. Be sure to test your code for correctness.

For example, given the line:

1, 3, 3, 1

the next line would be

1, 1, 2, 3, 1, 1

while the previous line would be

3, 1, 1, 1

## Theme: Sorting Algorithms

### Problem #5:

(a) For each of the three simple sorting methods *SelectionSort*, *BubbleSort*, and *InsertionSort*, discussed in Chapter 9 of the Carrano textbook, briefly sketch the operations needed to sort an array of N numbers in ascending order, if the initially provided array is sorted in descending order instead.

(b) For each of these three sorting methods, count the number of comparisons and number of swaps, separately, as a function of N. Represent the results in table format, as a function of N.

## Theme: Pointers

### Problem #6

In each of the code snippets shown below (which are to be considered independent of each other), describe in one sentence, in your own words, any pointer issues you recognize.

#### (a)

```
int *pnInt = 7;
```

#### (b)

```
int *pnInt = &7;
```

#### (c)

```
int nInt = 3, *pnInt;  
*pnInt = 7 * nInt;
```

#### (d)

```
int nInt = 7;  
int *pnInt = nInt;
```

#### (e)

```
int *pnInt = new int;  
*pnInt = 7;  
pnInt = NULL;
```

#### (f)

```
int *pnInt, nInt = 4;  
while (pnInt != NULL)  
{  
    pnInt = new int;  
    *pnInt = nInt;  
    nInt++;  
}
```

#### (g)

```
void addInt(int *pnFirst, int *pnSecond, int nSum)  
{  
    nSum = *pnFirst + *pnSecond;  
}
```