BİL 102 – Computer Programming
HW 08

Last Submission Date: April 30, 2014 – 09:00

Notes: For all recursive functions in this homework, unless otherwise stated, you are not allowed

- to use any helper function
- to use any extra parameter in a function
- to use loops in any functions.

1. **(25 Pts)** Write a recursive function which returns the number of the increasing subsequences in an integer array. This function should have the following prototype::

   int getNumOfIncreasingSubSeq(int arr[], int size)

   Some examples of increasing sub-sequences are given below:

Array	Number of increasing subsequence
1 2 5 0 2	2
5 2 3 1 7 5 3 1	6
8	1
Empty Array	0

2. **(25 Pts)** Write two recursive functions to print a given number in reversed and straight order. In each recursive call, these functions should print only one digit of the number. These functions should return the sum of all digits of the number. They should have the following prototype:

   int printNumberRevesedOrder(int number)
   int printNumber(int number)

3. (**25 Pts**) Write a recursive function "evalF" to evaluate the following mathematical function.

$$F: NxN \rightarrow N$$

$$F(x,y) = \begin{cases} F(x-1,y) + 2F(x,y-1) + x + y & ,if\ x,y > 0 \\ 0 & ,otherwise \end{cases}$$

Write down all function calls with return values for F(4, 3) (handwritten, to be delivered with printouts).

4. (**50 Pts)** (25 Pts if the length of the longest path is calculated only) In this part you will find the longest path in a given map starting from the lowest leftmost point (point (0,0)). Possible movement directions are **left, right and up**. The map is represented by a string array with maximum 20 rows and 20 columns. In the map, only the cells labeled with a '0' are permissible. The cells labeled with '1' represent obstacles and the cells are not permissible. A path is represented by a sequence of character '0's starting from the starting point to any point such that all movements are at a possible direction. An example of a map and the longest path are given in table 1 and table 2, respectively. Your implementation will take the map from a text file, "Map.txt", and print the longest path (as in table 2) and the length of it on console.

1	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	0	1	1	1
1	1	0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1
0	0	0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	1	1	1
0	0	0	0	0	1	0	1	1	1

Table 1 A representation of a map the longest path in it

.	.	.	.	.	.	.	.	.	.
.	.	x	.	.	.	.	.	.	.
.	.	x	.	.	.	.	.	.	.
.	.	x	x	.	.	.	.	.	.
.	.	.	x	.	.	.	.	.	.
.	x	x	x	.	.	.	.	.	.
.	x	.	.	.	.	.	.	.	.
x	x	.	.	.	.	.	.	.	.

Table 2 Representation of the longest path

Implementation Details:

Define an enumeration type "direction_t", which refers all possible directions and write your code by implementing at least the following functions:

getData: gets the map from a given file as a string array.

printTable: prints a string array on the console, used for printing map and the longest path.

findLongestPathRec: a recursive function to find the longest path. Takes the following arguments.

map: (input) string array representing the given map
mapRowC: (input) number of rows in the map
mapColC: (input) number of columns in the map
route: (input) string array representing the current route
routeLen: (input/output) length of the current route
longestRoute: (input/output) string array representing the longest route
longestRouteLen: (input/output) length of the longest route
yPos: (input) integer representing y coordinate of the current position
xPos: (input) integer representing x coordinate of the current position

"route" represents the current route. It is updated at the beginning and end of each recursive call. If the length of the "route" is bigger than the length of the longest route, longest route is

updated.

findLongestPath: a wrapper function for findLongestPathRec

5. **BONUS (25 Pts)**:

Implement part3 with an iterative function. Use a table to hold all previous data required. For example, for calculating F(n, m), you should calculate and store all F(x,y) with $0 \leq x < n$, $0 \leq y < m$.. Starting from F(0, y) and F(x, 0), calculate and store all these values. Compare by simple observation the performance of this method and the recursive method in part3.

General:

1. Obey honor code principles.
2. **Read your homework <u>carefully</u>** and follow the directives about the I/O format (data file names, file formats, etc.) and submission format **<u>strictly</u>**. Violating any of these directives will be penalized.
3. Obey coding convention.
4. Do not forget to put the required **tags** in the main function.
5. Your submission should include the following file **<u>and NOTHING MORE</u>** (no data files, object files, etc):

   HW08_<StudentName>_<StudentSirname>_<student number>.c

   Do **NOT** compress the file you submit.
6. Do not use non-English characters in any part of your homework (in body, **file name**, etc.).
7. Deliver the printout of your work **until the last submission date**.