CSE102 – Computer Programming (Spring 2021)

Homework #8

Handed out: 7:55pm April 22, 2021.

Due: 11:55pm May 4, 2021.

Hand-in Policy: Via Moodle. No late submissions will be accepted.

Collaboration Policy: No collaboration is permitted.

Grading: This homework will be graded on a scale of 100.

Description: In this homework, you will write a complete C program that implements several functions as described below. You are expected to reflect on what you have learned in class up to this point. Therefore you are not allowed to utilize any library or its function except for those mentioned in this document. You can use helper functions you coded as needed.

You are provided with four separate files (in HW8_Src.rar):

- **hw8_main.c:** Contains the main function. You are not expected to modify this file in your submission except for the test functions. You may modify it for your testing and debugging needs.
- **hw8_lib.h:** Contains the declarations of functions for this homework. You are not expected to modify this file in your submission except for defining enum types. Any additional functions you might write can be defined and used in hw8_lib.c file.
- **hw8_lib.c:** This file will contain your implementation of the functions declared in the associated header file. The details of the behavior of these functions are provided below.
- **makefile:** This is a makefile provided for you to use for compiling and testing your code.

The following provides the details of the functions to be implemented:

- **void clean_file(char* infile, char * outfile, char* words_to_delete[WORDSIZE], int number_of_words)**: This function reads a .txt file (name given in infile) and calls the following recursive function to delete words from this file and writes the result in another .txt file (name given in outfile).
    - **void delete_words (FILE * infid, FILE * outfid, char * words_to_delete[WORDSIZE], int number_of_words):** This function will receive handles (pointers) to the input and output text files. It will read the input file one line at a time, remove the occurrences of the words in the given list (words_todelete) and write the resulting new string as a line in the output file.

  You are not allowed to use any functions other than what is provided in stdio.h and stdlib.h. Furthermore, you are not allowed to use any loops in the implementation of these two functions or any helper functions you might write as part of your solution.

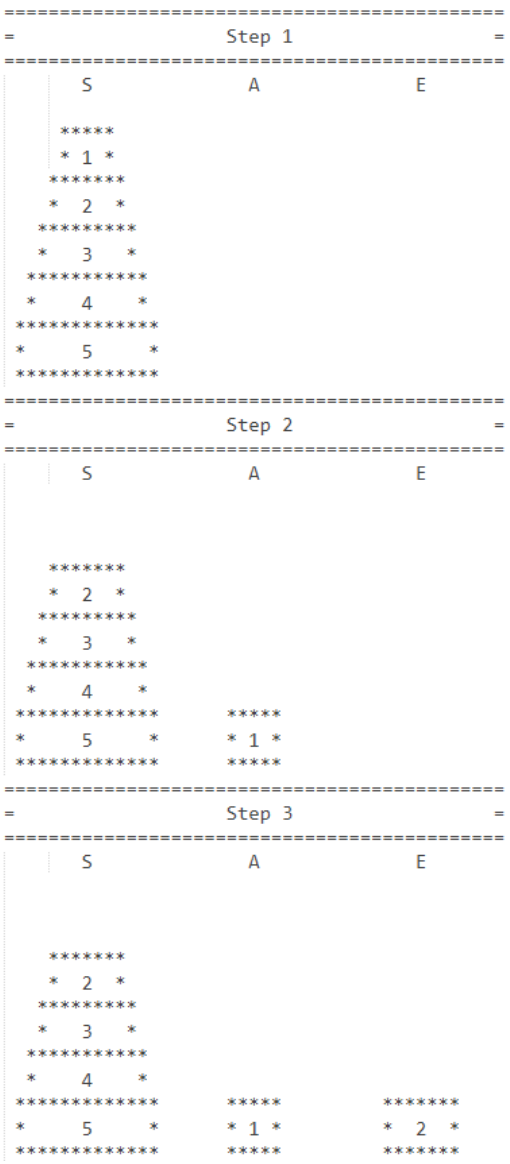
- Let us consider an improved version of the maze problem we discussed in class a couple of weeks ago. A maze board (8x8 in this case) is represented as a matrix of characters. Each cell can have one of the following enumerated values of cell_type:
    - cell_wall: wall – the player cannot move to this cell.
    - cell_free: free cell to move.
    - cell_target: target cell when reached the maze is solved.
    - cell_p1: the current (also the start) position of the first player in the maze.
    - cell_p2: the current (also the start) position of the second player in the maze.

  A player in the maze can make the following moves (enumerated values of move_type):
    - move_left: Move from the current position to the left if that position is free.
    - move_right: Move from the current position to the right if that position is free.
    - move_up: Move from the current position to the up if that position is free.
    - move_down: Move from the current position to the down if that position is free.

  int maze_move(cell_type maze[][8], cell_type player, move_type move): Implement this function using only recursions (no for, while or do-while loops). This function will check if the given move for the given player is valid. If valid, it will move the player. It will further check if the move has resulted in the player reaching the target. If so, the function will return 1, otherwise 0. You will probably need a set of helper recursive functions for this implementation.

- **void towers_of_hanoi(char start_peg, char end_peg, char aux_peg, int n)**: Enhance the implementation of the function given during the lecture on April 21st, 2021. Instead of printing the moves suggested by the algorithm, you visualize it as the following.

  A couple moves are shown for n=5, in the following example (this example is given only to show the output format, the moves are not necessarily what the algorithm would suggest):



Your solution should generalize for any values of n within the range 1 to 7. You may use loops in this implementation. You are also allowed to use functions from stdio.h and stdlib.h and nothing else.

Useful Hints: Here are some things that might make your development a bit easier.

- For testing your code use files for inputting data and getting the output. For example:

  $ hw8 < input.txt > output.txt

  will get the input from the file “input.txt” and will write the output to the file “output.txt”. This way you can easily make a lot of entries to test your code without using the keyboard again and again.

- Use the makefile to compile your code. You can add a run case to your makefile to do the compilation and testing with one simple make command.

What to hand in: You are expected to hand in a .zip or .rar file including all files you are provided regardless of modified or not.

- **HW8_lastname_firstname_studentno.rar / HW8_lastname_firstname_studentno.zip**