Workshop 2 Exercises

August 2, 2019

In this week, the following exercises are to be coded, compiled and executed. Skeleton files are available on the LMS.

1. Implement the function `insert_after(list_t* l,node_t* n,data_t d)` which inserts element `d` after node `node` in the list. Write some test code to make sure your implementation is correct. Use `WS2_list_insert_skeleton.c`.

2. Stacks and queues can also be implemented using an array of type `data_t` and pointers. Write a type `stack_t` which uses arrays instead of linked lists and provide functions for `make_empty_stack()` and `push()` and `pop()` in this representation. Use `WS2_stacks_skeleton.c`

3. Write a program that can read a $n \times m$ matrix of integers from stdin with the following format:

   ```
   n = 2
   m = 4
   4 812 94 24
   42 43 31 5
   ```

   (a) Use dynamic memory allocation to store the matrix in memory

   (b) Output the largest and smallest value in each column of the matrix

   The file to be used is available to you on the LMS as: `matrix.txt`. Use `WS2_dyn_alloc_skeleton.c`