**HW 1: To Do List (10 pts)**

Create an application To Do List. The program should input a list of to-do tasks ordered by priority from an input text file, and allow the user to choose from a menu of options:

* 1. Display a task at the location (selected by the user) (1 pt)
  2. Display all the tasks (1 pt)
  3. Replace a task (at the location selected by the user) (1 pt)
  4. Remove a task (at the location selected by the user) (1 pt)
  5. Add the new task (to the location selected by the user) (1 pt)
  6. Move a task to another location (ask the user from where to where) (1 pt)
  7. Quit (1 pt)

After Quit the updated list should be saved to the output text file. (1 pt)

**Implementation. (For each of the listed requirements – if not done the following penalties will apply).**

In main read tasks from an input text file into a dynamic array of strings (-0.5pt), such that the number of tasks is determined by the number of lines in the text file. (-0.5pt)

Store the array capacity and the number of tasks as two different variables. (-1 pt)

Implement a function for each of the menu options (except Quit), which gets called from main responding to the user’s choice of options, selecting Quit gets you out of the menu loop. (-2 pt) Each of these functions should work with the array of tasks but not the text file. (-2 pt)

After Quit is selected, the array should be copied onto the output text file and the program terminated. (-1 pt)

When a task is removed from the array, all the subsequent tasks should be shifted back and the number of tasks should be decreased. (-1 pt) If there are no tasks to be removed – the user should be informed that a removal could not be done. (-0.5pt)

When a task is added, all the subsequent tasks should be shifted forward and the number of tasks increased. (-1 pt) If you need to add a task above the previously allocated capacity, the capacity of the array should be increased using function *realloc*. (-0.5pt)

**Testing.** Test all the menu options. After selecting options 2-6 display all the tasks and the menu of options. (-0.5 for each not tested function).

**Submit to Blackboard page->Assignments->HW1:**

* (5 pts) source code: ToDoList.c,
* (3 pts) program output testing work of each function: test.pdf/test.jpg/test.doc/test.txt,
* (1 pt) input text file: original\_list.txt
* (1 pt) output text file: final\_list.txt.