

It is extremely important that you understand the programming assignments; please participate in the classroom discussion. You are expected to solve the programming assignments individually. You are encouraged to discuss with your friends. However, **do not copy**. Follow the submission and grading policies for full points.

Linked List, Stack, and Queue in C

The primary focus is to implementation *Linked List, Stack, and Queue* in C.

Programming Details:

The programming assignment should be discussed in the class. It is extremely important that you understand the assignments; please participate in the classroom discussion.

Q1 (mandatory): You must use Linked List (singly, doubly, or circularly) implementation of Stack or Queue to answer this question. If you plan to answer optional Q2, do not use Queue to answer Q1. The major steps, input file, and output file information is given below.

Major steps:

- 1) Enter ID. (Example: A123B456) >
NOTE: Responses should be saved in "A123B456responses.txt" output file.
- 2) Ten tests are available; select a test. (Example: 1) >
NOTE: if '*n*' is entered, then questions should be selected from "Test*n*questions.txt" input file.
 - a. A question should be displayed (randomly selected, repetition is okay).
 - b. User's answer to the question should be received.
 - c. Depending on the answer is correct or incorrect, points should be calculated.
 - d. Five questions should be asked (selected randomly from >5 questions, repetition is okay).
 - e. Total points (for the ID and for the test) should be stored in a file.NOTE: if '*n*' is entered, then scores should be saved in "Test*n*scores.txt" output file.
- 3) Go for another test? (Example: Y/N) >
 - a. For Y, repeat from 1.
 - b. For N, Exit.

Input file:

There are multiple input files – one per test (example: Test1questions.txt). Each input file contains a number of questions (and answers). An example is given below. The very first line contains the name of the test and the number of question-answer pairs. After that, each line contains the question number, the correct answer (T is for true and F is for false), and the question.

(Sample: Test1questions.txt)

```
Test1  27
1      T      A linked list can grow and shrink as a program runs.
2      F      Nodes in a linked list are stored in contiguous memory.
3      T      To create a linked list, one must first create a struct.
4      F      A basic linked list operation is: sorting a list.
5      T      To build a list initially, you can use an "append" routine.
6      F      When you delete a node from a list, you ensure that the links are broken.
7      T      Deleting an entire list does not require the use of the Delete operator.
...
27     T      The last node in a singly linked list points to NULL.
```

Output files:

Multiple output files should be generated/modified – one per test (example: Test1scores.txt) and one per student (example: A123B456responses.txt). These two types of files are explained below.

Test score files

Each test score output file should store the scores of all students who took that test. The very first line should represent the name of the test, maximum score possible, and the number of students who took the test. After that, each line should represent student ID, student score, and the date when the test was taken. An example is given below.

(Sample: Test1scores.txt)

Test1	5	4
A123B456	3	2012-09-21
P765Q678	5	2012-09-21
K767H676	5	2012-09-21
X987Y654	4	2012-09-22

Student response files

Each student response file should contain all the answers that the student provided. The very first line should represent student's ID and number of tests s/he participated. After that, each line should represent the name of the test, the question number, the answer that the student provided, and the date when the test was taken. An example is given below.

(Sample: A123B456responses.txt)

A123B456	1
Test1 23	T 2012-09-21
Test1 2	T 2012-09-21
Test1 5	F 2012-09-21
Test1 17	F 2012-09-21
Test1 11	T 2012-09-21

Sample Codes:

```
drzaman@kirk:~/CS460/Samplecodes$ pwd
/usr/users/User11/drzaman/CS460/Samplecodes
drzaman@kirk:~/CS460/Samplecodes$ ls -ltr
-rw-r--r-- 1 drzaman drzaman 2057 Sep 18 03:38 Test1questions.txt
-rw-r--r-- 1 drzaman drzaman 2057 Sep 18 03:38 Test2questions.txt
-rw-r--r-- 1 drzaman drzaman 2057 Sep 18 03:38 Test3questions.txt
```

Important Notes: There are two (2) questions. The first one is the mandatory question and the second one is the optional bonus question. Please make your file names as **Prog05q1.c (MAIN for mandatory)**, **Prog05q2.c (MAIN for bonus)**, etc. as discussed in class.

Q2 (Optional 25% Bonus Question): If you use Queue to answer Q1, you do not qualify for credits from optional Q2. In this part, you are asked to make the following changes to your program/answer to Q1.

- a) When questions are selected randomly, repetition is NOT allowed. *Ref: Major steps | 2) | a, d.*
- b) Using priority Queue concept, sort the test score files (example: Test1scoresPQ.txt) such that the record with maximum scores should be on top (i.e., can be accessed first), and so on. If there are multiple records with the same score, perform ascending sort on IDs to break the tie. An example is given below.

(Sample: Test1scores.txt) – Original file after Q1

Test1	5	4
A123B456	3	2012-09-21
P765Q678	5	2012-09-21
K767H676	5	2012-09-21
X987Y654	4	2012-09-22

(Sample: Test1scoresPQ.txt) – New file after Q2

Test1	5	4
K767H676	5	2012-09-21
P765Q678	5	2012-09-21
X987Y654	4	2012-09-22
A123B456	3	2012-09-21