



İ.T.Ü.

ITU Computer Engineering Department
BLG233E Data Structures
2nd Homework

Due Date: Dec 9, 2013 23:00

1-) For this question, implement an **evaluator** application for the **prefix expressions** by using stack data structure. You should read the expressions line by line from the file and the results should be written to the output file. In the table below, a simple input and an output are given. An example input file and the expected output of this input file is also given as “**input1.txt**” and “**output1.txt**”.

- The application should read the expressions from the input file(“input1.txt”), evaluate them and write the evaluated outcome to the output file(“output1.txt”).
- You should implement your own **stack structure** with **doubly linked list**.
- You should successfully **deallocate** all of the allocated memory before termination of your program.
- Make sure that **GNU C++ compiler (g++)** compiles your project, and the application runs in Linux smoothly. You can use **ITU ssh server** to compile and test your application.

Sample Input File	Expected Output File
+ * 11 2 5	27
* + 4 3 3	21
- / 12 3 2	2
* 2 + 13 - 4 2	30

2-) Implement a **priority queue** application for tasks that can read from the input file and creates the appropriate output file. In the table below, there is a simple input and output for this input. An example input file and the expected output of this input file is also given as “**input2.txt**” and “**output2.txt**”. The application should read data from the input file(“input2.txt”) and interpret the given commands.

- The **priority queue** should be implemented with **linked list**.
- “**enqueue priority task_name**” command consist of integer priority value and a task name. Enqueue operation should maintain the linked list ordered by priority. The tasks with higher priority should dequeued first.
- “**dequeue**” command should dequeue task with a highest priority and write it to the output file(“output2.txt”).

- You should successfully **deallocate** all of the allocated memory before termination of your program.
- Make sure that **GNU C++ compiler (g++)** compiles your project, and the application runs in Linux smoothly. You can use **ITU ssh server** to compile and test your application.

Sample Input File	Expected Output File
enqueue 8 A enqueue 5 B enqueue 10 C dequeue enqueue 9 D dequeue dequeue dequeue	C D A B

3-) Find an **example usage scenario for the following data structures. The examples should be different than the ones given in **slides of the course or the practice session**. **Explain why** that type of data structure is used on a given example and **benefits of using** this data structure.**

a-) Stack

b-) Queue

Note: If you have any question about the homework, contact research assistant Çağatay KOÇ via e-mail (kocca@itu.edu.tr) or in person (in Research Lab 3, EEB 4313).

Submission Procedure:

1. Prepare a report consist of answers and a brief information about implementation details.
2. Make sure you write your name and number to all of the files of your project in the following format:

```
/*
 * @Author
 * Student Name: !! enter here !!
 * Student ID : !! enter here !!
 * Date:
 */
```

3. Make sure that **GNU C++ compiler (g++)** compiles your project, and the application runs in Linux smoothly. This is important because we will **evaluate** your homework in **Linux using g++**.
4. Use comments wherever necessary in your code to explain what you did.
5. After you make sure that everything is **compiled** smoothly, submit the files through www.ninova.itu.edu.tr. Ninova enables you to change your submission before the submission deadline.

Academic dishonesty -including but not limited to cheating, plagiarism and collaboration- is unacceptable and subject to disciplinary actions. Any student found guilty will receive F as his/her final grade for the course.