Birla Institute of Technology & Science, Pilani
2nd Semester 2016-17 - CS F211 – Data Structures and Algorithms

---

Lab 4 (Evaluation 1) : 7th Feb 2017

Time: 170 minutes **Marks: 12 + 18 = 30**

Instructions:

- *This test consists of two problems (Problem 1 and Problem 2) specified in two different files.*
- All input expressions should be read from stdin (scanf) and output should be printed on stdout (printf).
- For first 150 minutes, only a subset of test cases will be visible to students after submitting the code on the portal. Only in last 20 minutes, all test cases will be made visible.
- At the end of 170 minute period, the online system will stop evaluating the submissions but it will accept it for additional 10 minutes. At the end of 180 minute period, it will stop accepting the submissions.
- Only the last submission by the student for each problem will be considered for evaluation, irrespective of earlier correct submission.
- Assuming that a problem contains M marks, in case of (Run-error/Compiler-error/Timelimit-error), evaluation will be done for M/2 marks only.
- Total marks of each problem contains some marks for modularity and proper structuring of code.
- All submitted source code will be later checked manually by the instructor and final marks will be awarded. Any case of plagiarism and/or hard coding of test cases will fetch 0 marks for the problem/evaluation component.
- Make sure to return 0 from the main() function in case of normal termination.

Problem 1 of 2

Expected Time: 70 minutes **Marks: 12**

Problem Statement

Implement a FIFO list (i.e. a Queue) ADT with operations create, add, delete, and isEmpty using a linked list with a first pointer as well as a last pointer (pointing to the first and the last nodes respectively). Create a FIFO list *fls1*.

Input format

Each line will start with a one of the following key values (0, 1, 2, 3, or 4). Each key corresponds to a function and has a pattern. Implement functions create, add,

delete, isEmpty, and traverse according to given pattern and write a driver function which can call the functions according to given key value.
Note: This problem will call create function only once, in the beginning i.e. there will be a single list.
For memory profiling, create following additional constructs:

- Global variables:
  - *curHeapSize* : To store currently allocated amount of heap memory (in bytes) in the program
  - *maxHeapSize* : To store maximum amount of heap memory (in bytes) which was allocated during execution of the program.
- Wrapper Functions
  - *void* myMalloc(unsigned int size)*
    - Wrapper function over malloc(). It should be called instead of malloc() in this problem. Should update above global variables to update memory status.
  - *void myFree(void *ptr)*
    - Wrapper function over free(). It should be called instead of free() in this problem. Should update above global variables to update memory status.

Key	**Function to call**	**Format**	**Description**
0	create	0 N val_1, val_2 , … , val_N	Shows start of a new list. All operations till next "0" key value should be on the current list. Read N as an integer following "0". Then, read next N numbers on the same line and insert them in sequence to *fls1*. Call traverse function in the end.
1	add	1 val_1, val_2 , … , val_M ,-1	Keep reading numbers after "1" key value, until a "-1" is encountered and insert all numbers in sequence to *fls1*. Call traverse function in the end.
2	delete	2 N	Delete N numbers from *fls1*. Call traverse function in the end. Assume that N will be smaller than number of nodes in the list.
3	isEmpty	3	Should return 1 if the list is empty, otherwise 0.
4	traverse	4	Print all numbers in current list on a single line separated by a tab. If the list is empty, print -1.
7	memProf	7	Print *curHeapSize* and *maxHeapSize*

			variables on a line, separated by a tab.
-1		-1	stop the program.

Test Case 1:

Input	Output
0 3 45 56 87	45 56 87
1 23 19 -1	45 56 87 23 19
2 2	87 23 19
1 -2 999 -165 -98 -1	87 23 19 -2 999 -165 -98
2 4	999 -165 -98
3	0
2 3	-1
3	1
-1	