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

Lab 4 (Evaluation 1) : 9th Feb 2017

Time: 170 minutes **Marks: 8 + 22 = 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: 50 minutes **Marks: 8**

Problem Statement

A matrix of dimensions M * N can be represented by an array of M arrays of length N each. A vector of dimension L can be represented by an array of length L.

All memory allocations for matrices and vectors should be dynamic only.

Input format

Each line will start with a one of the following key values (1, 2, 3, 4, 5, 7, or -1). Each key corresponds to a function and has a pattern. Implement following function according to given pattern and write a driver function which can call the functions according to given key value.

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.
- Structure *myPtr (void*, int)* : holds the actual pointer (pointing to the allocated block) and the size allocated
- Wrapper Functions
  - *myPtr* myMalloc(unsigned int size)*
    - Wrapper function over malloc(). It should be called instead of malloc() in this problem. Should return pointer to a *myPtr* object which should contain base address of allocated memory and its size. It should also update above global variables accordingly.
  - *void myFree(myPtr *ptr)*
    - Wrapper function over free(). It should be called instead of free() in this problem. Should update above global variables with *size* information available in *ptr*.

Key	Function to call	Format	Description
1	readMatrix	1 matID M N val_{11} val_{12} … val_{1N} val_{21} val_{22} … val_{2N} …… val_{M1} val_{M2} … val_{MN}	“1” shows start of a new matrix. “matID” represents Matrix ID, which will be unique integer among matrices. “M” represents number of rows. “N” represents number of columns. Next M lines will have N space separated integer values. Call printMatrix function in the end.
2	rcMul	2 $matID_1$ R $matID_2$ C	“2” shows call to rcMul (Row Column Multiplication) function. “$matID_1$” represents Matrix ID of first matrix M1. “R” represents row number (starting with 1) of first matrix. “$matID_2$” represents Matrix ID of second matrix M2. “C” represents column number (starting with 1) of second matrix. rcMul should return summation of a vector V such that: *V[i] = M1 [R, i] x M2 [i, C]*, where 1 <= i <= M1.columns Print sum on a new line.

3	rmMul	3 matID$_1$ R matID$_2$ F	“3” shows call to rmMul (Row Matrix Multiplication) function. “matID$_1$” represents Matrix ID of first matrix M1. “R” represents row number (starting with 1) of first matrix. “matID$_2$” represents Matrix ID of second matrix M2. “F” is a flag (0 or 1). 0 means do not print resultant vector, and 1 means print resultant vector on a new line (space separated values). rmMul should return a vector W such that: W[i] = rcMul (M1, R, M2, i), where 1 <= i <= M2.columns.
4	mmMul	4 matID$_1$ matID$_2$ matID$_3$	“4” shows call to mmMul (Matrix Matrix Multiplication) function. “matID$_1$” represents Matrix ID of first matrix M1. “matID$_2$” represents Matrix ID of second matrix M2. “matID$_3$” represents Matrix ID of output product matrix M3. mmMul should create M3 such that: M3 [i] = rmMul (M1, i, M2), where 1 <= i <= M1.rows Do not print anything.
5	printMatrix	5 matID	Print matrix with given matID such that each Row is in a new line and values are tab separated on each line.
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				
1 **1** 3 4	1	2	0	3	
1 2 0 3	2	0	0	1	
2 0 0 1	1	0	0	0	
1 0 0 0	2	0	3	1	0
1 **2** 4 5	0	0	1	2	0
2 0 3 1 0	1	1	0	0	1
0 0 1 2 0	2	0	1	0	5
1 1 0 0 1	7				
2 0 1 0 5	2	0	3	1	0
2 **1** 2 **2** 3	8	0	8	5	15
3 **1** 3 **2** 1	6	0	7	2	5
4 **1 2 101**	2	0	3	1	0
5 **101**	1	2			
1 20 5 2	3	0			
1 2	0	6			
3 0	0	0			
0 6	0	0			
0 0	8	64			
0 0	6	54			
4 101 20 102	2	22			
5 102					
-1					