OPERATING SYSTEM FINAL LAB EXAM SPRING 2016

Name:_____________ **Time Allowed: 2hrs**
Roll#:_____________ **Total Marks:100**

IMPORTANT INSTRUCTIONS:

1. Your code must be well **indented, aligned, commented** and properly written.
2. Total time of the paper includes submission time. **No extra time will be provided for submission.**
3. Submission MUST be on **XEON** mentioned at the end of the document where folder is created. Email or any other way of submission will not be accepted.
4. Manage your time wisely after reading question paper carefully.
5. Submit only these files **Question.c. Do not submit .rar file.**
6. Use of internet, cell phones, USB or any other helping material will award you **F-Grade** in Lab.

   Good Luck

Question: **[100]**

Write a C program to construct a **Stack using Shared Memory Segment**.
The details of the program are as follows:-
The parent process takes two file names as input from user. First file has integers while second file has alphabets (Both files have a special character to specify the end of file). Parent process creates three child processes.

1. First child reads the integers from the file , pushes it on stack and output the position of stack with the pushed integer till it reaches end of file character.
2. Second child reads the alphabets from the second file , pushes it on stack and output the position of stack with the pushed alphabet till it reaches end of file character.
3. The third child pops the stack , output the popped element with its position on stack till there is no more element on stack left.

All the output of the stack that the child processes displayed to user is passed to parent process using a shared memory segment, the **parent process** then writes the output from the three child processes onto an output file with proper order.

Note: You also have to synchronize these processes to avoid undesirable situations.

The Output must be:

```
Input file names:  'file1.txt  file2.txt'
Child 1: Integer  10 pushed on stack position 0
Child 2: Alphabet 'A' pushed on stack position 1
Child 3: Popped 'A' at position 1 of stack
Child 2: End of file reached, all elements of file has been pushed on
   stack.
Child 3: Popped 10 at position 0 of stack.
Child 1: End of file reached, all elements of file has been pushed on
   stack.
Child 3: Stack is completely  empty
Parent: All output written to 'outputfile.txt'
```

Submission: The code should be submitted to the following folder with your roll number...

```
//sandata/XEON/Spring2016/OS LAB Section-B/Mid/Submissions/XX-XXXX/,
   where X represents your Roll Number e.g. 13-4045
```

Further Information: For helping material and the syntax of the related APIs, refer to the provided material of shared memory, message queue and forked processses pdf file “API Help.pdf”.