Online Assignment 2: Stack (07/12/2021 Tuesday-Afternoon)

In your offline component you already have implemented Arr (ArrStack) and LL implementation of Stack ADT as specified in the offline specification document. You also have implemented the one array two stack (OATS) concept while simulating the Dishwasher. Now, you will have to formalize the OATS concept as follows. You will have to write a class, OATS, which provides all functionalities of the Stack ADT and can handle the situation when the 2 stacks overlap with each other (so far we ignored this). OATS will have its own array OATS_Arr for maintaining the two stacks. OATS will use the ArrStack class and particularly its 'second' constructor thereof to create two stacks in one array. OATS will provide the following methods utilizing the methods of ArrStack:

Table 1: OATS methods

Fn #	Function	Comment
7	pushA(item)	Works on Stack A
8	popA()	
9	topValueA()	
10	pushB(item)	Works on Stack B
11	popB()	
12	topValueB()	
OPTIONAL		
13	clearA()	Works on Stack A
14	clearB()	
15	lengthA()	Works on Stack A
16	lengthB()	

While working with the stack operations through the above functions, if there is an overlap between stack A and stack B, you must detect that and then double the size of the underlying array (OATS_Arr).

Submission Guidelines:

1. **Create a directory with your 7-digit student id as its name**
2. **You need to create one file for the OATS implementation code (e.g. oats.cpp/oats.py). You will also include your previous ArrStack implementations as a separate file. Create a separate file for the main function to demonstrate the OATS implementation. You cannot use any DS apart from your ArrStack implementation.**
3. **Put all the source files only into the directory created in step 1. Also create a readme.txt file briefly explaining the main purpose of the source files.**
4. **Zip the directory (compress in .zip format. Any other format like .rar, .7z etc. is not acceptable)**
5. **Upload the .zip file on Moodle in the designated assignment submission link. For example, if your student id is 1905xxx, create a directory named 1905xxx. Put only your source files (.c, .cpp, .java, .h, etc.) into 1905xxx. Compress the directory 1905xxx into 1905xxx.zip and upload the 1905xxx.zip on Moodle.**

Failure to follow the above-mentioned submission guideline may result in upto 10% penalty.