Course : COMP203 – Data Structures and Algorithms 06/12/2020
TA : Gökhan Göy
LAB6

Questions

1. Implement a nonrecursive Java method for enumerating all permutations of the numbers {1, 2, …, N} using a stack. N is passed as an argument to your method, so your code should be generic.
   Hint: You can represent the permutations as Strings and store the numbers to permute in an array. You can consider a bottom-up approach and start from the empty String. Then in each iteration pop a permutation from the stack, add a number that is not in the String, repeat these for all permutations in the stack and push those extended permutations back to stack. Report a permutation when its length reaches N. (40 pts.)

   Input : N=3
   Output:

```
Main
"C:\Program Files\Java\jdk1.8.0_251\bin\java.exe" ...
321
312
231
213
132
123

Process finished with exit code 0
```

   Input : N=4
   Output:

```
Main
"C:\Program Files\Java\jdk1.8.0_251\bin\java.exe" ...
4321
4312
4231
4213
4132
4123
3421
3412
3241
3214
3142
3124
2431
2413
2341
2314
2143
2134
1432
1423
1342
1324
1243
1234

Process finished with exit code 0
```

2. Stacks are often used to provide “undo” support in applications like a Web browser or text editor. While support for undo can be implemented with an unbounded stack, many applications provide only limited support for such an undo history, with a fixed-capacity stack. When push is invoked with the stack at full capacity, rather than throwing an exception, a more typical semantic is to accept the pushed element at the top while “leaking” the oldest element from the bottom of the stack to make room. Modify *push* and *pop* methods in the ArrayStack class provided fort his lab using a circular array. (30 pts.)

3. Implement a Java method with signature transfer(S, T) that transfers all elements from stack S onto stack T, so that the element that starts at the top of S is the first to be pushed onto T, and the element at the bottom of S ends up at the top of T. Implement a driver application with main method that demonstrates the transfer operation. You can use the ArrayStack code provided for this lab. (30 pts)

HINT: You need to use provided classes for his lab assignment.

IMPORTANT NOTE : Your code will be check via using a software similarity tool. If there will be similarity between people then they will graded as 0 without any objection !!!

Submission : You need to upload a Q1.java file for the 1st question and Q3. java file for the 3rd question and the changed version of ArrayStack.java for this lab assignment. You need to zip it those files first then you can upload it to canvas.

Deadline : 04.12.2020 – 4:00 pm.