

CS1020 Take-home Lab #4

Exercise #1: Cargo Optimization

http://www.comp.nus.edu.sg/~cs1020/3_ca/takehomelabs.html

Objectives:

1. Usage of Java Stack and other data structures
(<http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html>)
2. Usage of Generic
3. Problem solving

Task statement:

(Note that unless otherwise stated, you may assume that all input data are valid and hence there is no need for you to perform input data validation.)

A cargo terminal stores large containers that arrive by road. These containers are stacked at the terminal as they arrive. After all the containers for the day arrive, they are loaded onto ships for transport abroad.

Cargo ships carry large numbers of containers. Each ship is uniquely identified by an uppercase letter 'A' through 'Z'. The time to load a ship depends in part on the locations of its containers. The loading time increases when the containers are not on the top of the stacks, but can be fetched only after removing other containers that are on top of them.

The cargo terminal needs a plan for stacking containers in order to decrease loading time and it has to unload all containers as soon as they arrive. The plan must allow each ship to be loaded by accessing only topmost containers on the stacks, and minimizing the total number of stacks needed.

You are to complete the program **CargoOptimization.java** to read in the arrival sequence of the containers in a line. Each character in the line represents a container which is loaded onto a particular ship indicated by that character. Ships will arrive at the terminal in alphabetical order. There are no limits on the number of containers that can be placed in a single stack. The program will output a single integer indicating the minimum number of stacks required for efficient loading.

The program contains both the classes **Container** and **CargoOptimization**.

Note:

- Please use the **Stack** class in the Java API.
- Please use generics wherever applicable.

Refer to the sample runs for the output format.

Number of submissions:

You are given **12** submissions. Only the final submission will be graded.

Sample run #1:

KADD
2

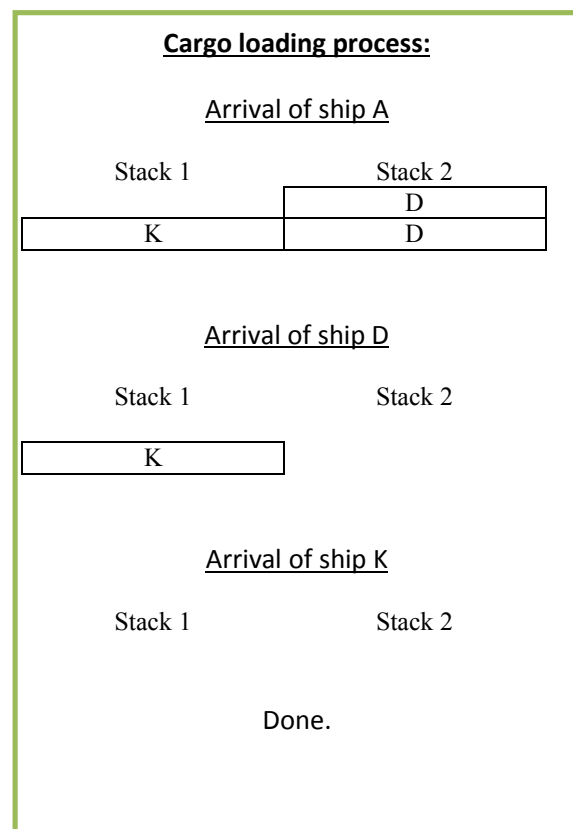
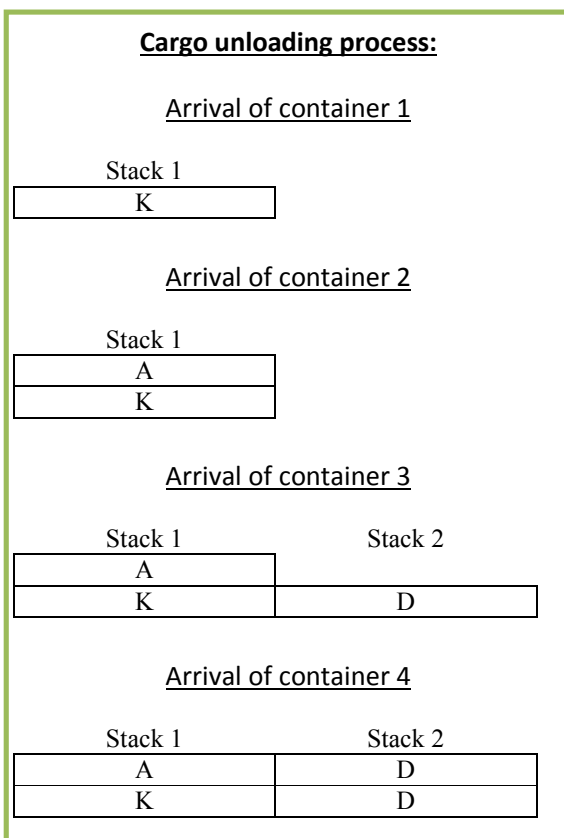
Sample run #2:

SPHSPHSPHSPHSPH
3

Sample run #3:

CCCCBBBBBAAAA
1

The following is the process for sample run #1:

**Credits:**

All credits to problem setter on UVa

(http://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=628&page=show_problem&problem=3503)