

Q1:

There are Exceptions for situations like calling POP() method when there is no element left in the list.

After reading test.csv file and filling the stacks if we try to write another csv file by using pop() method of each stack we get the following error

133530 for stack A

75062 for stack B

69925 for stack C

607210 for stack D

Nano second performance of stacks when pop() ing and writing to file.

Stack C is fastest and Stack D is Slowest.

Q2:

We use iterator of Kwnlist to reverse the stack by changing their datas.

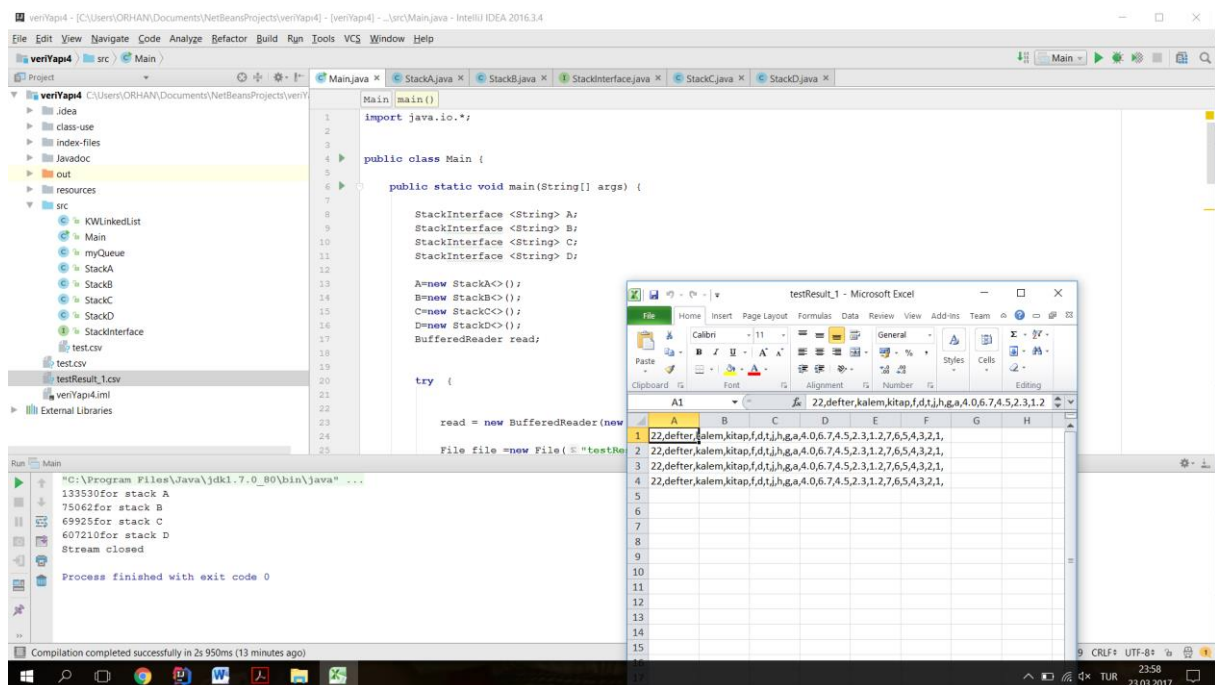
So we use set() method to do that. We swap datas of nodes. Like last one and first one.

For recursive method, we simply pop() elements until nothing is left, then we start adding them. Last one popped will be first one to add.

Q3:

We use `toString.compareTo()` method to compare elements. And remove accordingly. `priorityA` inherits `LinkedList` class there is no need to write methods like `size()` `isEmpty()` but `priorityB` uses a `LinkedList` type list so we have to write those methods.

Q1:



Q2:

The screenshot shows the IntelliJ IDEA IDE with a Java project named 'veri4soru2'. The main class is 'Main.java', which contains a linked list implementation. The code uses a `LinkedList` and a `LinkedListIterator` to traverse the list and print its elements. The output is displayed in the 'Run' console, showing the sequence of numbers: 1, 2, 7, 6, 4, 5, 3, 2, 1. A Microsoft Excel window titled 'testResult_2' is also open, showing the same sequence of numbers in a single cell.

```

1.2
7
6
4
5
3
2
1
Stream closed
Process finished with exit code 0

```

Q3:

The screenshot shows the IntelliJ IDEA IDE with a Java project named 'VeriYapiHw4Soru3'. The main class is 'Main.java', which contains a program that reads two input files, 'testResult_1.csv' and 'testResult_2.csv', and writes their contents to a new file, 'testResult_3.csv'. The output is displayed in the 'Run' console, showing the sequence of numbers: 1, 1, 2, 2, 2, 3, 4, 4, 0, 4, 5, 5, 6, 6, 7, 7, a, d, d, e, f, e, r, f, g, h, i, j, k, a, l, m, k, i, t, a, p, t. A Microsoft Excel window titled 'testResult_3' is also open, showing the same sequence of numbers in a single cell.

```

1.1
2.2
2.2
2.3
3.4
4.0
4.5
5.5
6.6
7.7
a.d
d.e
f.e
g.h
i.j
k.a
l.m
k.i
t.a
p.t
Stream closed
Process finished with exit code 0

```

