

Quiz 5

Subject: Stack and Queue
Due Date: 24.04.2022 23:59

Problem 1: Display octal equivalents of decimal numbers

Write a main program that takes decimal numbers(between 1000-200000) from an input file (input.txt) as an argument and put them into any data structure, then finds and displays their octal equivalents by using a stack (output.txt), i.e convert the number with base value 10 to base value 8.

Create a Stack class with:

- One constructor
- Push
- Pop
- Top
- isFull()
- isEmpty()
- Size

You must use ONLY stack(s) for decimal-to-octal operation, don't use other data structures such as normal array, string etc.

Algorithm:

- Store the remainder when the number is divided by 8 into a stack.
- Divide the number by 8 now
- Repeat the above two steps until the number is not equal to 0.
- Print the content now.



8	2980			
8	372	—	4	← LSD
8	46	—	4	
8	5	—	6	
	0	—	5	← MSD

Figure 1: Representation of decimal-to-octal

Problem 2: Design Queue

Design a queue that supports enqueue and dequeue operations in the front, middle, and back.

Notice that when there are two middle position choices, the operation is performed on the frontmost middle position choice. For example:

- Enqueue 6 into the middle of [1, 2, 3, 4, 5] results in [1, 2, 6, 3, 4, 5].
- Dequeue the middle from [1, 2, 3, 4, 5, 6] returns 3 and results in [1, 2, 4, 5, 6].

Implement the Queue class with:

- void enqueueFront(int val) Adds val to the front of the queue.
- void enqueueMiddle(int val) Adds val to the middle of the queue.
- void enqueueBack(int val) Adds val to the back of the queue.
- int dequeueFront() Removes the front element of the queue and returns it. If the queue is empty, return -1.
- int dequeueMiddle() Removes the middle element of the queue and returns it. If the queue is empty, return -1.
- int dequeueBack() Removes the back element of the queue and returns it. If the queue is empty, return -1.

You must use ONLY queue for problem 2 operation, don't use other data structures such as normal array, string etc.

1 Grading Policy

- Your work will be graded over a maximum of 100 points.
- Your total score will be partial according to the grading policy stated below.

Stack and Queue classes	20p
Taking input files as arguments from command line and correct output	80p

2 Execution and Test

- Upload your java files to your server account (dev.cs.hacettepe.edu.tr)
- Compile your code (javac *.java)
- Run your program (java Main input.txt)
- Control your output
- **Note: In the input file provided to you, the entries of problem 1 will always be given before the entries of problem 2.**

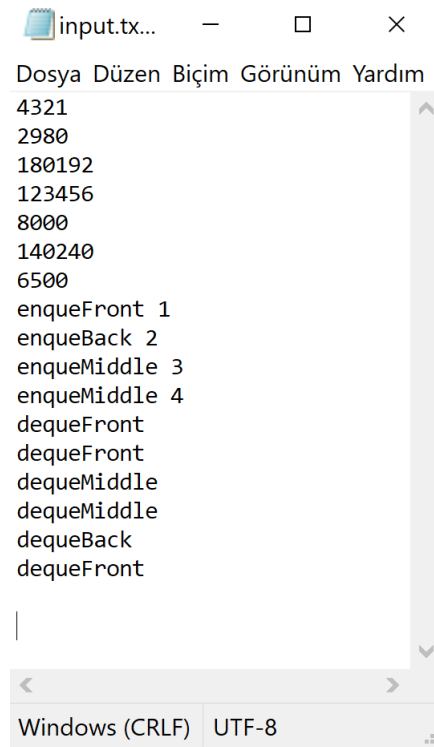


Figure 2: Example of input file(input.txt)

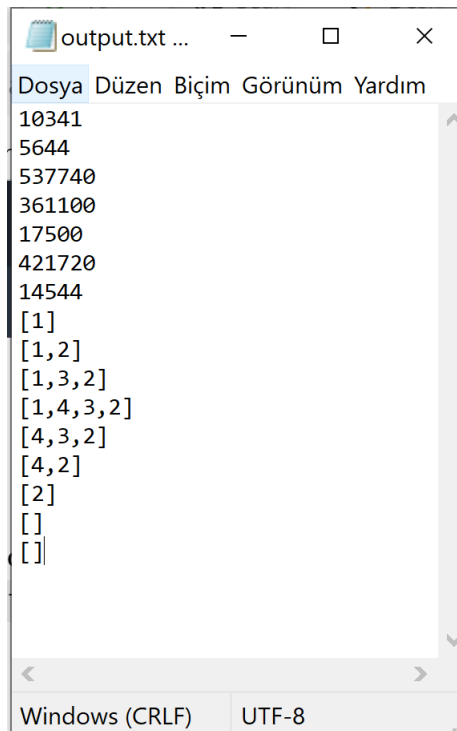


Figure 3: Example of output file (output.txt)

Notes

- **You MUST implement your own stack and queue classes.**
- The name of input file is input.txt; the name of output file is output.txt. Do not change the name of the files.
- Do not miss the submission deadline.
- Save all your work until the quiz is graded.
- You can ask your questions via Piazza and you are supposed to be aware of everything discussed on Piazza.
- You must submit your work with the file hierarchy as stated below:

- <studentid>.zip
 - <src>
 - Main.java
 - Stack.java
 - Queue.java
 - *.java