

## Lab 1 – due 9/7 at 11pm

### Review for stack and queue

1. Create a generic stack class with a linked list as its underlying data structure. You will instantiate the stack class for strings in your driver. You must write this on your own - do not get code from the internet. You will need to use the proper method names – push, pop. Write a driver that will push the string onto the stack if it is not a “\$”. If it is a “\$”, it will pop the head of the list and print it. Use the input:

to be \$ or not \$ \$ to be that is the question \$ \$ \$

Once you have processed the input and printed the results of the \$’s, print whatever is left on the stack.

The answer should be:

```
be not or question the is
Left on stack: that be to to
```

Write a second driver that will instantiate the stack for postfix math equation for integers:

3 4 \* 2 5 \* +

The answer should be:

```
The result of the postfix equation is 22
```

2. Create a generic queue class with an array as its underlying data structure. You will instantiate the queue class for floats in your driver. You must write this on your own - do not get code from the internet. You will need to use the proper method names – enqueue and dequeue. Initially make your queue hold 4 items. As you add items and you fill it up, double the size of your queue (the array). Write a driver that will enqueue the following numbers:

4.5 2.6 3.2 7.2 15.5 17.2 2.7 16.2 13.3

As you dequeue each number, add the number to running total and print:

Example:

4.5 dequeued, running total is 4.5

2.6 dequeued, running total is 7.1

Etc.

Submission: submit your code in gradescope and screen shots showing each run with inputs and outputs.

Rubric: Each problem is worth 50 points:

- 5 points – style – used comments and proper indentation
- 5 points – submitted screenshot of the run(s)
- 10 points – created the proper driver(s) to “drive” your data structures to get the correct answer
- 10 points – you created the proper data structures as specified in the write up (generic stack with list as underlying data attributes, generic queue with array as underlying data attributes)
- 5 points – use the proper method names for the classes (insert for push, pop for stack; enqueue, dequeue for queue)
- 10 points – prints the correct output (answers)
- 5 points – reads in the input as specified above
- You will get a 0 if you:
  - a) Do not submit the source code