Lab 2

Given two `.o` files containing all functions related to stack and queue along with their header files, solve the following problems.
Remember the allowed operations for queues and stacks (for eg, you cannot access q->back or q->front (it is a pointer, queue only allows access to elements) since it is not allowed)

- TLDR, you are only allowed to interact with the stack/queue using functions
- You can compile/link them using normal gcc command `gcc objectfile.o source_code.c -o output`

Figure out the various time complexities.

Example: Implement a stack using queue

- Solve using the .o files provided

Q1: Implement a queue using stack (30 mins coding time)

- Solve using the .o files provided

Q2: Infix to postfix using stack (30 mins coding time)

- Solve using the .o files provided
- Remember the order of precedence for brackets and the arithemetic operations
- `A * B - (C + D) + E` becomes `A B * C D + - E +`

Homework:

Evaluate a postfix expression using a stack without converting it into a infix expression

---

Implement a deque (45 mins coding time)

Deque (double-ended queues) are generalized queues.
They support the following operations, all in O(1) time:

- push_back(d, x) -> Adds an elemen x at the back of the deque d
- push_front(d, x) -> Adds an element x at the front of the deque d
- pop_front(d) -> Removes the first element in deque d and returns it
- pop_back(d) -> Removes the last element in deque d and returns it
- empty(d) -> Checks if a deque d is empty
- size(d) -> Returns the number of elements in deque d

- front(d) -> Returns the first element in deque d
- back(d) -> Returns the last element in deque d
  Implement a deque from scratch
