1. Write a function AlternatingSplit() that takes one list and divides up its nodes to make two smaller lists. The sublists should be made from alternating elements in the original list. So if the original list is {a, b, a, b, a}, then one sublist should be {a, a, a} and the other should be {b, b}. The elements in the new lists may be in any order (for some implementations, it turns out to be convenient if they are in the reverse order from the original list.)

```
/* Given the source list, split its nodes into two shorter lists. If we number the elements 0, 1, 2, ...
then all the even elements should go in the first list, and all the odd elements in the second. The
elements in the new lists may be in any order. */

void AlternatingSplit(struct node* source, struct node** aRef, struct node** bRef) {

        // Your code

}
```

<u>You will have to write the code for creating the initial list as well!</u> (add-first/last) and print function.

2. What are the operations used by stacks and queues? Briefly describe their working mode (access modes) and limitations (if any).