

6) [10 points] Implement a queue of ints using two stacks, st1 and st2. Assume you have the following stack functions.

```
typedef struct {
    int *contents;
    int maxSize;
    int top;
    int count;
} stackT;

void StackInit(stackT *stackP, int maxSize);
void StackPush(stackT *stackP, int element);
int StackPop(stackT *stackP);
bool StackIsEmpty(stackT *stackP);
```

Here is a sample main() with output.

```
// queuetest.c
stackT st1;
stackT st2;

int main(int argc, char **argv){

    debugflag = true;
    int stacksize = 10;
    StackInit(&st1, stacksize);
    StackInit(&st2, stacksize);
    int val;

    for (int i=1; i < argc; i++) {
        val = strtod(argv[i], NULL);
        if (val > 0) {
            enqueue(val);
        } else {
            dequeue();
        }
        StackPrint(&st1);
        StackPrint(&st2);
    }
    return(0);
}
```

Here is sample output:

```
% queuetest 3 4 -3 6
Stack: size: 1 :[3]
Stack: size: 0 :
Stack: size: 2 :[4] [3]
Stack: size: 0 :
Stack: size: 0 :
Stack: size: 1 :[4]
Stack: size: 1 :[6]
Stack: size: 1 :[4]
-bash-4.3$ ./queuetest 3 -1 -3
Stack: size: 1 :[3]
```

```
Stack: size: 0 :  
Stack: size: 0 :  
Stack: size: 0 :  
queue is empty  
Stack: size: 0 :  
Stack: size: 0 :
```

Define the following two functions and give the respective runtime complexities:

```
// pushes an int on the queue  
// what is the average and worst case runtime complexity?  
void enqueue(int element);
```

```
// removes next element from queue (first in first out)  
// what is the average and worst case runtime complexity?  
int dequeue();
```