

Some exercises on data structures

1. Exercises on **singly linked lists**. For these exercises, give a pseudo-code or a sequence of steps described in English to perform the requested exercise on linked list.

- (a) Given a linked list of integers sorted from smallest to largest, and a pointer to a single node containing an integer, insert the node in the linked list so that it remains sorted.
- (b) Given a nonempty list, delete the last node and set the new last link to null.
- (c) Given a null terminated linked list, rearrange its nodes into two lists: <first node, third node, fifth node, ...> and <second node, fourth node, sixth node, ...>. Do not allocate any new nodes.
- (d) Given a null-terminated linked list, reverse the order of its nodes

2. Exercises on stacks:

- (a) A letter means push and an asterisk means pop in the following sequence. Give the sequence of values returned by the pop operations when this sequence of operations is performed on an initially empty stack.

E A S * Y * Q U E * * * S T * * * I O * N * * *

- (b) Suppose that an intermixed sequence of push and pop operations are performed. The pushes push the integers 0 through 9 in order; the pops print out the return value. Which of the following sequences could not occur?

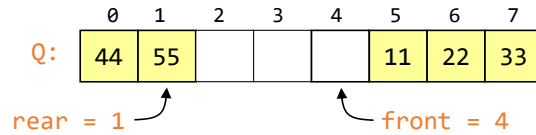
- (a) 4 3 2 1 0 9 8 7 6 5
- (b) 4 6 8 7 5 3 2 9 0 1
- (c) 2 5 6 7 4 8 9 3 1 0
- (d) 4 3 2 1 0 5 6 7 8 9

- (c) Give a pseudo-code that reads in a positive integer and prints the binary representation of that integer. Hint: divide the integer by 2.
- (d) Consider the following sequence of stack commands: push(a), push(b), push(c), pop(), push(d), push(e), pop(), pop(), pop(), pop().
Change the position of the pop() commands in the above sequence so that the items are popped in the following order: b,d,c,a,e. You are not allowed to change the ordering of the push commands

3. Exercises on queues:

- (a) Consider the following sequence of stack operations: push(d), push(h), pop(), push(f), push(s), pop(), pop(), push(m). Suppose we replace the push and pop operations with enqueue and dequeue respectively. What would be the sequence of dequeued values, and what would be the final state of the queue?
- (b) Show how to implement a stack using two queues. Describe how the operations push and pop will be implemented using the queue operations enqueue() and dequeue(). Give the running time of each stack operation.

(c) Consider the queue Q below:



Execute the sequence of operations on Q as dictated by the loop below. If an Enqueue() operation is attempted when the queue is full, skip such operation until the loop asks for a Dequeue() operation. If an Dequeue() operation is attempted when the queue is empty, skip such operation until the loop asks for an Enqueue() operation: $A = [a,b,c,d,e,f,g,h]$

```
for ( $i = 0$ ;  $i < 8$ ;  $i++$ ){  
    if ( $A[i] \leq A[i+1]$ ) Dequeue(Q);  
    else Enqueue(Q,A[i]);  
}
```

- i. Write on your answer sheet the value of each entry of Q after the execution of all the above operations. Note, each time Dequeue(Q) is performed, replace the corresponding entry in queue Q by -1.
- ii. What is the final value of the pointers $rear$ and $front$?