A. Consider the following operations applied in the indicated order to a List ADT implemented using Singly Linked List (forward only pointer) with head and tail pointers.

1. [3 points] Define the current state of the linked list after performing each operation. The state can be defined using list notation [] where the first elements points to head and last element points to the tail. For example, [1,2,3] shows a linked list of three elements connected using singly linked list.

add(0, x)	[x] (answer is provided as an example)
add(0, y)	
add(1, z)	
remove(0)	

2. [1 points] what is the cost of add(i) for the above data structure?

3. [2 points] What is the cost of remove(i) for the above data structure?

B. Consider reverse() operation for a List ADT. This operation reverse the order of indices such as the first element should become the last element and last element should become the first element.

a. [7 points] What would be the most efficient way to implement the reverse() operation if the ADT is implemented using **double linked list**? Also provide the cost of your method and justify your answer briefly.

b. [7 points] What would be the most efficient way to implement the reverse() operation if the ADT is implemented using **arrays**? Also provide the cost of your method and justify your answer briefly.

A. Consider the following operations applied in the indicated order to an initially empty ArrayQueue implementation with an initial array of size 1. The implementation is based on circular structure and provide add(x) and remove() operations.

1. [7 points] Define the current state of the array for each operation,

add(x)	0 x
add(y)	
add(z)	
remove()	
add(p)	
add(q)	
add(r)	
remove()	

2. [2 points] How many times the resize() operation has been performed?

3. [1 points] What is the cost of add(x) if ignore the cost of resize() operation.

B. [4 points] Consider RandomQueue ADT provides add(x) and remove() operations. It is similar to Queue ADT except for the remove() operation removes an element that is chosen randomly. Can we implement the remove() operation in O(1) time complexity if the ADT is implemented using an array (if we ignore the cost of resize())? Provide your arguments briefly.

C. [6 points] If we implement a List ADT using singly linked list (a linked list with only forward only pointer), what would be the cost of the following operations. Provide brief justification/proof.

a. add_first(x): add an element at the start of the list

b. remove(): remove an element from the list using FIFO