1: Generally speaking, sorting primitives is faster than sorting objects.

Answer: **True**

2: Sink Implementation()

Here is the sink method for a maxheap:

private void sink(int k) {

while(2\*k <= n) {

int j = 2\*k;

if (j < n && less(j, j+1)) j++;

if (!less(k, j)) break;

exch(k, j);

k = j;

}

}

Answer: C. **Because children of a node in a binary heap are not necessarily in order**

**3:** Consider the following pseudo code. Assume that IntQueue is an integer

queue. What does the function fun do?

Answer: C. **Prints first n Fibonacci numbers**

4: A max-heap is a heap where the value of each parent is greater than or equal

to the values of its children. Which of the following is a max-heap?

Answer: **Option B**

5: Please select which of the following statements are true about a binary

heap:

Answer: **All 4 options seems correct**

1. A binary heap is a complete binary tree as a complete binary tree provides the opportunity to

use a compact array representation that does not involve explicit links.

1. The binary heap does not enforce any ordering between the nodes of one level.
2. A binary heap is a complete binary tree, by which we mean that all nodes have either two or zero

children.

1. The binary heap is a data structure that can efficiently support priority-queue operations.

Problem – Reverse the Linked List

You’re given the pointer to the head node of a linked list. Change

the

next

 pointers of the nodes so that their order is reversed. The head

pointer given may be null meaning that the initial list is empty.

Input Format

You have to complete the

SinglyLinkedListNode

reverse(SinglyLinkedListNode head)

 method which takes one

argument - the head of the linked list. You should NOT read any input from

stdin/console.

The input is handled by the code in the editor and the format is as follows:

The first line contains an integer , denoting the number of test cases.

Each test case is of the following format:

The first line contains an integer , denoting the number of elements in the

linked list.

The next  lines contain an integer each, denoting the elements of the linked

list.

Constraints

, where  is the  element in the list.

Output Format

Change the

next

 pointers of the nodes such that their order is reversed

and

return

 the head of the reversed linked list. Do NOT print anything to

stdout/console.

The output is handled by the code in the editor. The output format is as

follows:

For each test case, print in a new line the elements of the linked list after

reversing it, separated by spaces.

Sample Input

1

5

1

2

3

4

5

Sample Output

5 4 3 2 1

Explanation

The initial linked list is: 1 -> 2 -> 3 -> 4 -> 5 -> NULL

The reversed linked list is: 5 -> 4 -> 3 -> 2 -> 1 -> N

Solution that I submitted:

static SinglyLinkedListNode reverse(SinglyLinkedListNode head) {

if(head == null) return null;

SinglyLinkedListNode prev = head;

SinglyLinkedListNode current = head;

current = current.next;

while(current != null){

SinglyLinkedListNode temp = current.next;

current.next = prev;

current = temp;

prev= prev.next;

}

return head;

}