Create a Linked list by making the last node's next pointer points to the fifth node on the list.

Now write a function to find, whether there is a loop in the LL and if it there what is the node value of the start of the loop node(5th node). (Hint: use slow ptr, fast ptr)

Examplle : L = { 4 7 8 6 2 1 9 5 3 } , here the last node's (data 3) next ptr should be linked to 5th node (data value 2)

Input : 4 7 8 6 2 1 9 5 3 -1 (read with cin >> till -1)

Output : 1 (as there is loop) , in no loop 0 : but here as the LL is constructed with loop you will get output 1.

: 2 (at the node (data value 2) the loop starts, i,e 5th node)

CCC - T2 - 22 - 9 - 2020

Write a function for deleting from list LL, nodes occupying positions **indicated in list LL itself.**

For instance, if L= (1 3 5 7 8) , then after deletion, L= (3 7).

Explanation: The positions of nodes at 1 , 3, 5 are nodes of 1 , 5, 8 of given original LL are deleted. There is no node at positions 7 and 8 in the original LL. so The original LL will now become as L = (3 7)

You should not use another linked lists or arrays, but you have to re-adjust the existing list nodes by using few variables.

Input : 1 3 5 7 8 -1 (cin >> till -1)

output : 3 7 (print the LL)

CCC - T3 - 22-9-2020

Create a Generic Queue where each element is having varying number of items.

Example Q = { (3 , 'N' , 9.8 , 6) , ('N' , 8, 'C' , 5, 'L', 8.1) , (9.7 , 5.4, 'B') , (7, 2, 45, 4.5, 9.3,72, 81, 36) , ('A' ,6, 'C', 7.5,'D') }

each of 🟩 🟥 🟦 🟧 🟩 are elements of queue.

Implement enqueue() , Dequeue() operations on the queue.

Print the contents of the queue after creation and a dequeue operation.

Note : **You should not use Pointers.**