

Indian Institute of Technology (ISM), Dhanbad
Department of Computer Science and Engineering
Data Structures Lab(CSC204)

Assignment 2

Full Marks: 50

1 The Problem

Define a structure to represent a node of a doubly linked list to store one integer and one next and one previous pointers. Then create an array of n pointers to the structure you have defined to keep track of n doubly linked lists. You need to take the value of n from the user. On this data structure you need to write functions to support the following operations:

1. *insert(i, j, x)*: You need to insert a node with data element x at the j^{th} position of the i^{th} list. If there are lesser than j nodes in the i^{th} list, then insert node at the end of the i^{th} list. (10)
2. *delete(i, j)*: You need to delete the node present at the j^{th} position of the i^{th} list. If there are lesser than j nodes in the i^{th} list, then print an error message. (5)
3. *deleteX(i, x)*: You need to delete the node with data element x of the i^{th} list. If there are no such node in the i^{th} list, then print an error message. If there are multiple nodes with the same element x , delete the last occurrence of the element x . If x is not present in the i^{th} list then print error. (5)
4. *getElementPrev(i, j, x)*: You need to return the j^{th} previous element of x in the i^{th} list. If no such element exist return the 1st element in the i^{th} list. If there are multiple nodes with the same element x , then return the j^{th} previous element of the last occurrence of the element x in the i^{th} list. If x is not present in the i^{th} list then print error. (10)
5. *getElementNext(i, j, x)*: You need to return the j^{th} next element of x in the i^{th} list. If no such element exist return the last element in the i^{th} list. If there are multiple nodes with the same element x , then return the j^{th} next element of the first occurrence of the element x in the i^{th} list. If x is not present in the i^{th} list then print error. (10)
6. *printList()*: You need to print the whole collection of doubly linked lists. (5)

Overall design and organisation of the program carries 5 marks. Full marks 50.

2 Sample output

The following data corresponds to the collection of 3 doubly linked lists.

Doubly Linked List:

List 0 : 1 < - > 2 < - > 3 < - > 4

List 1 : 7 < - > 8 < - > 9

List 2 : 10 < - > 11 < - > 12 < - > 14

Operation *insert(1, 2, 50)*

Doubly Linked List:

List 0 : 1 < - > 2 < - > 3 < - > 4

List 1 : 7 < - > 8 < - > 50 < - > 9

List 2 : 10 < - > 11 < - > 12 < - > 14

Operation *insert*(2, 20, 12)

Doubly Linked List:

List 0 : 1 < - > 2 < - > 3 < - > 4

List 1 : 7 < - > 8 < - > 50 < - > 9

List 2 : 10 < - > 11 < - > 12 < - > 14 < - > 12

Operation *getElementPrev*(2, 1, 12): 14

Operation *getElementPrev*(2, 3, 12): 11

Operation *getElementPrev*(2, 10, 12): 10

Operation *getElementNext*(2, 1, 12): 14

Operation *getElementNext*(2, 2, 10): 12

Operation *getElementNext*(2, 100, 11): 12

Operation *getElementNext*(2, 1, 50): Element 50 is not found in list 2

Operation *delete*(1, 1)

Doubly Linked List:

List 0 : 1 < - > 2 < - > 3 < - > 4

List 1 : 7 < - > 50 < - > 9

List 2 : 10 < - > 11 < - > 12 < - > 14 < - > 12

Operation *deleteX*(1, 12)

Doubly Linked List:

List 0 : 1 < - > 2 < - > 3 < - > 4

List 1 : 7 < - > 50 < - > 9

List 2 : 10 < - > 11 < - > 12 < - > 14