Data Structures and Algorithms (CS/IS F211)

Lab: 3 **Date: 11-02-2014**

Instructions:

- You are expected to use C/C++ language only. You will need only g++ , gedit and VI editor for today's lab, if these softwares are not there on your system, you should call the evaluators/assistants.

- Your code will be evaluated by the server. So you strictly need to follow the given Input/Output format. Any request for recheck on this basis will not be entertained later.

- You will not be allowed to upload solution after 11.00am..

- Upload your code in the format <full_id_no>.c (in a single file) on the under mentioned Server
  http://10.1.5.104/domjudge/team

- Login detail
  - ➢ Username : <Full ID No in UPPERCASE>
  - ➢ Password: Will be given to you.

- You will be allowed to use language of your choice from next lab. So check the required environment (appropriate eclipse plugin) on your system and report it to your evaluators/assistants

- ## Question:

Implement the following operations on a given linked list of elements of character type.

1. **Insert** an element at any arbitrary position

2. **Delete** an element at any arbitrary position

3. **Undo** operation to arbitrary steps
Your undo operation can go back up to **5** steps. If input steps for undo operation is **greater than 5** or the **number of Insert/Delete operations performed before undo are less than input number of steps in undo operation** than print **"no"** as the output and exit from program
(*Hint***:** you can use circular array of size 5 to store every operations)

4. **Display** all elements in linked list

5. **Exit**

Input format:

<No. of elements> <Elements of linked list> **(First Line)**

<operation> <position/no of backward steps to undo operations> <element> **(Second line onwards)**

Notations:

<1 > <Position><Element>

<2 > <Position>

<3> <Number of times undo have to be done (steps) >

<4>

<5>

1- Insert 2- Delete 3- Undo 4- Display 5- Exit

SAMPLE TESTCASE

Input case1

```
10 a b c d e f g h i j
1 2 x
1 7 y
2 3
3 2
4
5
```

Output case1

```
a x b c d e f g h i j
```

Input case2

```
10 a b c d e f g h i j
1 2 x
1 7 y
2 3
3 6
4
5
```

Output case2

```
no
```

Input case3

```
10 a b c d e f g h i j
1 2 x
2 3
3 4
4
5
```

Output case3

```
no
```