EXAMINATION QUESTION PAPER

e-DAC, May 2021

EXAM Type : Main **Module Name : Algorithm and DS (Using Java)** **SET: A**

DATE: 30 /8/2021 **Duration: 2hrs** **Max. Marks: 40**

INSTRUCTIONS:

- All Questions are compulsory.
- Create one folder on your machine rename it with your PRN (Pls use complete PRN)
- Copy all the solution files of all questions in the above created folder
- After copying all the files right click on main folder select option "Add to Achieve" and click on radio button in front of zip option.
- It will create zip folder of your PRN. This will be your final folder to upload.
- **Submissions : Code Only**
- Please note before uploading zip folder make sure you have copied all your relevant files in the respective folders. Once you upload the final folder you will not be able to re upload it.
- Kindly ensure before creating zip file, make sure all the required files should be closed to avoid any corruption of the zip file.
- Zip File should be renamed with FULL PRN No. of the candidate. (For example: 210540181001.zip)

Q. Implement a doubly linked list class that implements following interface.

```
public interface LinkedListIntf {
  // The function adds the given element as the first element in the linked list.
  public void AddAtFront(int element);
```

6 marks

```
  // The function adds the given element at the specified position (position is 1 based, i.e., first position in list is 1) in the linked list.
  // For example: If list contain elements - 10 5 20
  // Function call AddElementAtPosition(30, 1) will add element 30 at position 1 (first position). So, the resultant list will be - 30 10 5 20
  // If list contain elements - 10 5 20
  // Function call AddElementAtPosition(30, 5) will add element 30 at last position (as given position is bigger than size of list). So, the resultant list will be - 10 5 20 30
  // If list contain elements - 10 5 20
  // Function call AddElementAtPosition(30, 2) will add element 30 at position 2. So, the resultant list will be - 10 30 5 20
  // If list is empty, function call AddElementAtPosition(30, 2) will add element 30 at position 1. So, the resultant list will be - 30
  public void AddElementAtPosition(int element, int position);
```

12 marks

```
  // The function will return all elements in the list by doing forward traversal.
  public int[] GetAllElements();
```

5 marks

```
    // The function will return all elements in the list by doing backward/reverse traversal.
    public int[] GetAllElementsInReverse();                                    5 marks

    // The function will delete all occurance of elements in the list.
    public void DeleteAll(int element);                                        12 marks
}
```