Data Structure and Algorithms **Computer Engineering Degree** **University Carlos III of Madrid** **COURSE 2021-2022**	uc3m | Universidad Carlos III de Madrid

Name:

Instructions:

- Exam duration: 60'.
- Download midtermexam87.zip and unzip it .
- You must write your solution in the file **midtermexam87.py**. Add a python comment at the beginning of the file with your first and last name.
- DO NOT MODIFY the test.py file (it helps you test your solution).
- Only your Python framework (Spyder, Pycharm) can be open on your computer. You are not allowed to consult any other material (code, notes, books, etc).
- When there are 5 minutes left, the teacher will notify you to upload your solution. You will be then allowed to open Aula Global and upload the midtermexam87.py file (only this file) to the “MidTerExam” activity in Aula Global.
- It is your responsibility to check that you have uploaded the correct version of the file.
- Cell phones must be turned off and can never be on the table.
- You cannot leave the classroom until the end of the exam.
- It is not allowed to go to the bathroom.

Problem:

Implement a new method in the abstract data type SList (Singly Linked List), ***reverseDup()***. The method must reverse the list and eliminate all consecutive duplicated elements. The complexity of the solution must be O(n).

Note: consider that in the list there are maximum 2 consecutive repeated elements.

Some examples:

List before operation	**Operation**	**List after operation**
1<->2<->3<->4<->5<->6<->7	l. *reverseDup()*	7<->6<->5<->4<->3<->2<->1
1<->2<->3<->4<->5<->6<->6	l. *reverseDup()*	6<->5<->4<->3<->2<->1
1<->2<->3<->4<->4<->5<->6	l. *reverseDup()*	6<->5<->4<->3<->2<->1
1<->1<->2<->2<->3<->3	l. *reverseDup()*	3<->2<->1
1<->1	l. *reverseDup()*	1
(Empty list)	l. *reverseDup()*	(Empty list)

You cannot add new attributes or functions to classes DNode and MyDList .

For the solution to be considered correct, the proposed function must be **correct** (it solves the problem), **robust** (has no errors and works for any input) and **efficient** in terms of temporal and spatial complexity (avoid the use of auxiliary structures). Also, the code should be easy to understand and maintain.

The use of Python structures such as dictionaries or lists is not allowed.