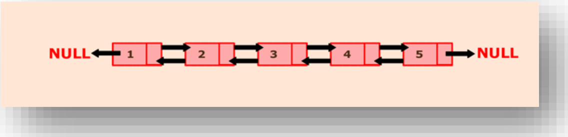
Chongqing University of Technology

Subject: Data Structure and Algorithm

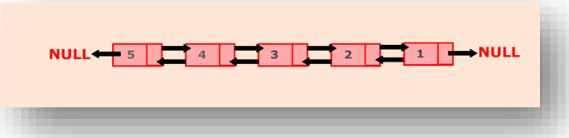
Instructor: Henry

Assignment_02

1. Java program to create a **doubly linked list** of n nodes and display it in reverse order. In this program, you create a doubly linked list, and then reverse the list by reversing the direction of the list and print out the nodes.
   For Example, consider this list



Expected output



- Define a Node class which represents a node in the list. It will have three properties: data, previous which will point to the previous node and next which will point to the next node.
- Define another class for creating a doubly linked list, and it has two nodes: head and tail. Initially, head and tail will point to null.
- addNode() will add node to the list:
  - It first checks whether the head is null, then it will insert the node as the head.
  - Both head and tail will point to a newly added node.
  - Head's previous pointer will point to null and tail's next pointer will point to null.

- If the head is not null, the new node will be inserted at the end of the list such that new node's previous pointer will point to tail.
- The new node will become the new tail. Tail's next pointer will point to null.

a. reverse () will reverse the given doubly linked list.

- Define a node current which will initially point to head.
- Traverse through the list by making current to point to current. Next in each iteration till current points to null.
- In each iteration, swap previous and next pointer of each node to reverse the direction of the list.
- In the end, swap the position of head and tail.

a. display () will show all the nodes present in the list.

- Define a new node 'current' that will point to the head.
- Print current Data till current points to null.
- Current will point to the next node in the list in each iteration.

2. Implement the information below with **linked List**

Following String Elements Liam, Noah, Ames, Oliver, Benjamin, Lucas, Mason, Jacob, Michael, Daniel, Jackson, Aiden, Matthew

The public constructors and methods required for the Array List class are listed here. The type E is the generic type of an element of the list. Implements all the method.

- int size (): Return the size (number of items) in this Array List.
- Boolean isEmpty(): Return true if this Array List has no items. (This is the same as the size equal to zero.) Return false if the size is greater than zero.
- void add (E value): Add the given element, value, to the end of the list.
- Void Traverse (): Display all the Elements
- void add (int index, E value): Add the element “William” “Elijah” “Logan” “Ethan” “Michael” “Henry” “Sebastian”, value to the list at the following position 3,7,10,12,14,16,18.
- Void Traverse (): Display all the Elements.
- E get (int index): get the position for Ames, Mason, Michael, Jackson, Matthew.

- E remove (E value): Removes the Elements Ames, Benjamin, Mason, Jacob, Michael, Jackson. And display the result.
- void clear (): Removes all the elements from this list.

** NOTICE**

1. Test your code in main method.

2. Your code won’t be same of others if it is then both of your assignment will be declined.

3. Submit it on time. make it zip and send it to akabillah4420@gmail.com don’t forget to write the subject name “Assignment_02” and student id in the mail. And be sure that your code has received by replying okay.

Deadline: 11 October 12:00 am BD time and 2:00 pm China time (Thanks)