|  |  |
| --- | --- |
| **Mudterm Skill Test** | |
| **Course Code:** CPE 201 - L | **Program:** Computer Enginnering |
| **Course Title:** Data Structure and Algorithm | **Date Performed:** 09/06/2025 |
| **Section:** 2B | **Date Submitted:** 09/06/2025 |
| **Name:** Elpedes, Glen Jorge A. | **Instructor:** Engr. Maria Rizette H. Sayo |
| 1. **Objectives** | |
| * to demonstrate dynamic memory allocation and pointer-based linking by implementing a singly-linked list data structure to store and manage an integer sequence (1 to 30). * To comprehend traversal, insertion, and reversal logic in linear data structures, one must be able to execute the fundamental linked list operations of displaying data, adding a new node, and reversing the list. | |
| 2. Discussion | |
| * This test emphasis on important operations like display, append, and reverse, this task shows how to manage integers dynamically using a singly-linked list. It reinforces fundamental data structure concepts and aids in understanding how nodes are connected and altered via pointers. | |
| **3. Materials and Equipment** | |
| 1. Google Colab  - Used to create python program and show the output.  2. Desktop computer  - Use to run the program and for typing the codes.  3. Operating system (e.g Windows 10 or 11)  - most python softwares requires the lastest version of operating system | |
| **4. Procedure** | |
| * Provide data and next attributes to a Node class or structure. * Use the append method to create a LinkedList and initialize it with integers ranging from 1 to 30. * Explore and show every node value. * Add a new node at the end with a specified value, like 31. * Change each node's next pointers to reverse the list. * To verify the operation, show the list in reverse. | |
| **5. Output** | |
| A.) Display all data    B.) Append a Node    C.) Reverese all No. | |
| **6. Conclusion**   * This test carrying out fundamental operations like display, append, and reverse, this exercise reinforced important ideas in dynamic data handling and pointer manipulation while assisting in the understanding of singly-linked lists. | |
|  | |