Rajalakshmi Engineering College

Name: tharunika R
Email: 241801296@rajalakshmi.edu.in
Roll no: 241801296
Phone: 6369646218
Branch: REC
Department: I AI & DS FD
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Bob is tasked with developing a company's employee record management system. The system needs to maintain a list of employee records using a doubly linked list. Each employee is represented by a unique integer ID.

Help Bob to complete a program that adds employee records at the front, traverses the list, and prints the same for each addition of employees to the list.

Input Format

The first line of input consists of an integer N, representing the number of employees.

The second line consists of N space-separated integers, representing the employee IDs.

Output Format

For each employee ID, the program prints "Node Inserted" followed by the current state of the doubly linked list in the next line, with the data values of each node separated by spaces.

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 4
101 102 103 104
Output: Node Inserted
101
Node Inserted
102 101
Node Inserted
103 102 101
Node Inserted
104 103 102 101
```

Answer

```
#include <iostream>
using namespace std;

struct node {
   int info;
   struct node* prev, * next;
};

struct node* start = NULL;

# You are using Python
class Node:
   def __init__(self, data):
      self.data = data
      self.next = None
      self.prev = None

class DoublyLinkedList:
   def __init__(self):
      self.head = None

   # Insert at the front of the list
   def insert_at_front(self, data):
      new_node = Node(data)
      if not self.head:  # If the list is empty
          self.head = new_node
      else:
          new_node.next = self.head
          self.head.prev = new_node
          self.head = new_node

   # Display the list from front to back
   def display(self):
      current = self.head
      while current:
          print(current.data, end=" ")
          current = current.next
      print()

# Main function to handle input/output
def main():
   # Read the number of employees
   N = int(input())

   # Read the employee IDs
   employee_ids = list(map(int, input().split()))

   # Create an empty doubly linked list
   employee_list = DoublyLinkedList()

   # Insert each employee ID at the front and display the list
   for emp_id in employee_ids:
      employee_list.insert_at_front(emp_id)
      print("Node Inserted")
      employee_list.display()

if __name__ == "__main__":
   main()
int main() {
   int n, data;
   cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> data;
        insertAtFront(data);
        traverse();
    }
    return 0;
}
```

Status : Correct ***Marks : 10/10***