LAB-05

Name: K V Jaya Harsha

Roll no: CS23B1034 Date: 28-08-2024

Q1. Single Linked List. (in cpp)

```cpp
//CS23B1034
//K V Jaya Harsha
#include <iostream>
using namespace std;
struct Node{
    int data;
    Node *next;
};
struct Node *create_l_l(int n){
    Node *head = nullptr;
    Node *temp = nullptr;
    int value;
    for (int i = 0; i < n; i++){
        cout << "Element " << (i + 1) << " : ";
        cin >> value;
        Node *newNode = new Node;
        newNode->data = value;
        newNode->next = nullptr;
        if (head == nullptr){
            head = newNode;
        }
        else{
            temp = head;
            while (temp->next != nullptr){
                temp = temp->next;
            }
            temp->next = newNode;
        }
    }
    return head;
}
void display_l_l(struct Node *head){
    Node *temp = head;
    while (temp != nullptr){
        cout << temp->data << " -> ";
        temp = temp->next;
    }
    cout << "NULL" << endl;
}
int main(){
    cout << "no of nodes: ";
    int n;
    cin >> n;
    Node *head = create_l_l(n);
    cout << "Linked List: ";
    display_l_l(head);
    return 0;
}
```

```
no of nodes: 3
Element 1 : 2
Element 2 : 3
Element 3 : 4
Linked List: 2 -> 3 -> 4 -> NULL
```

Q2. Sum of odd terms in linked list. (in cpp)

```cpp
// CS23B1034
// K V Jaya Harsha
#include <iostream>
using namespace std;
struct Node{
    int data;
    Node *next;
};
struct Node *create_l_l(int n){
    Node *head = nullptr;
    Node *temp = nullptr;
    int value;
    for (int i = 0; i < n; i++){
        cout << "Element " << (i + 1) << " : ";
        cin >> value;
        Node *newNode = new Node;
        newNode->data = value;
        newNode->next = nullptr;
        if (head == nullptr){
            head = newNode;
        }
        else{
            temp = head;
            while (temp->next != nullptr){
                temp = temp->next;
            }
            temp->next = newNode;
        }
    }
    return head;
}
void display_l_l(Node *head){
    Node *temp = head;
    while (temp != nullptr){
        cout << temp->data << " -> ";
        temp = temp->next;
    }
    cout << "NULL" << endl;
}
int sum_of_odd(Node *head){
    Node *temp = head;
    int result = 0;
    int position = 1;
    while (temp != nullptr){
        if (position % 2 != 0){
            result += temp->data;
        }
        temp = temp->next;
        position++;
    }
    return result;
}
int main(){
    cout << "Number of nodes: ";
    int n;
    cin >> n;
    Node *head = create_l_l(n);
    cout << "Linked List: ";
    display_l_l(head);
    cout << "Sum: " << sum_of_odd(head) << endl;
    return 0;
}
```

```
Number of nodes: 6
Element 1 : 12
Element 2 : 43
Element 3 : 46
Element 4 : 562
Element 5 : 4321
Element 6 : 1
Linked List: 12 -> 43 -> 46 -> 562 -> 4321 -> 1 -> NULL
Sum: 4379
```

3. Swapping elements in linked list. (in cpp)

linked_list_swap.cpp

```cpp
//CS23B1034
//K V Jaya Harsha
#include <iostream>
using namespace std;
struct Node{
    int data;
    Node *next;
};
Node *create_l_l(int n){
    Node *head = nullptr;
    Node *temp = nullptr;
    int value;
    for (int i = 0; i < n; i++){
        cout << "Element " << (i + 1) << " : ";
        cin >> value;
        Node *newNode = new Node;
        newNode->data = value;
        newNode->next = nullptr;
        if (head == nullptr){
            head = newNode;
        }
        else{
            temp = head;
            while (temp->next != nullptr){
                temp = temp->next;
            }
            temp->next = newNode;
        }
    }
    return head;
}
void swap_adjacent(Node *head){
    Node *temp = head;
    while (temp != nullptr && temp->next != nullptr){
        int swap_data = temp->data;
        temp->data = temp->next->data;
        temp->next->data = swap_data;
        temp = temp->next->next;
    }
}
void display_l_l(Node *head){
    Node *temp = head;
    while (temp != nullptr){
        cout << temp->data << " -> ";
        temp = temp->next;
    }
    cout << "NULL" << endl;
}
int main(){
    cout << "Number of nodes: ";
    int n;
    cin >> n;
    Node *head = create_l_l(n);
    cout << "Linked List: ";
    display_l_l(head);
    swap_adjacent(head);
    cout << "Swapped Linked List: ";
    display_l_l(head);
    return 0;
}
```

```
Number of nodes: 5
Element 1 : 1
Element 2 : 2
Element 3 : 3
Element 4 : 4
Element 5 : 5
Linked List: 1 -> 2 -> 3 -> 4 -> 5 -> NULL
Swapped Linked List: 2 -> 1 -> 4 -> 3 -> 5 -> NULL
```