|  |  |
| --- | --- |
| Name: | Taaha Hussain Khan |
| Roll No: | L1F21BSCS0917 |
| Subject : | D12 |

Task 01

//Taaha Hussain Khan

//L1F21BSCS0917

//D12

#include <iostream>

using namespace std;

class Node

{

public:

    int data;

    Node \*next;

    Node(int data)

    {

        this->data = data;

        next = NULL;

    }

};

Node \*nthNodeFromEnd(Node \*head, int n);

void display(Node \*head);

void insertAtTail(Node \*&head, int data);

int main()

{

    Node \*head = NULL;

    int key, n, node;

    cout << "Enter the number of nodes: ";

    cin >> n;

    for (int i = 0; i < n; i++)

    {

        cin >> key;

        insertAtTail(head, key);

    }

    cout << "Linked List : " << endl;

    display(head);

    cout << "Enter the value of n: ";

    cin >> node;

    cout << "N: " << node << endl;

    if (node <= n)

    {

        Node \*result = nthNodeFromEnd(head, node);

        cout << "Output : "<< result->data << endl;

    }

    else

    cout << "Nothing there! " << endl ;

}

void insertAtTail(Node \*&head, int data)

{

    Node \*n = new Node(data);

    if (head == NULL)

    {

        head = n;

        return;

    }

    Node \*temp = head;

    while (temp->next != NULL)

    {

        temp = temp->next;

    }

    temp->next = n;

}

void display(Node \*head)

{

    Node \*temp = head;

    while (temp != NULL)

    {

        cout << temp->data << " -> ";

        temp = temp->next;

    }

    cout << "NULL" << endl;

}

Node \*nthNodeFromEnd(Node \*head, int n)

{

    Node \*s = head, \*fast = head;

    int count = 0;

    while (fast != NULL)

    {

        fast = fast->next;

        count++;

        if (count > n)

        {

            s = s->next;

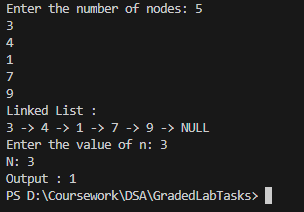
        }

    }

    return s;

}

Output:



Task 02

//Taaha Hussain Khan

//L1F21BSCS0917

//D12

#include <iostream>

using namespace std;

class Node

{

public:

    int data;

    Node \*next;

    Node(int data)

    {

        this->data = data;

        next = NULL;

    }

};

void insertAtTail(Node \*&head, int data);

void display(Node \*head);

Node \*reverse(Node \*&head);

Node \*reverseRecursive(Node \*&head);

int main()

{

    Node \*head = NULL;

    int num, n;

    cout << "Enter the number of nodes: ";

    cin >> n;

    for (int i = 0; i < n; i++)

    {

        /\* code \*/

         cout << "Enter data for node "<< i <<" : ";

        cin >> num;

        insertAtTail(head, num);

    }

    cout << "Original Linked List : " << endl;

    display(head);

    Node \*newhead = reverseRecursive(head);

    cout << "Reversed Linked List : " << endl;

    display(newhead);

    return 0;

}

void insertAtTail(Node \*&head, int data)

{

    Node \*n = new Node(data);

    if (head == NULL)

    {

        head = n;

        return;

    }

    Node \*temp = head;

    while (temp->next != NULL)

    {

        temp = temp->next;

    }

    temp->next = n;

}

void display(Node \*head)

{

    Node \*temp = head;

    while (temp != NULL)

    {

        cout << temp->data << " -> ";

        temp = temp->next;

    }

    cout << "NULL" << endl;

}

Node \*reverse(Node \*&head)

{

    Node \*prevptr = NULL;

    Node \*currptr = head;

    Node \*nextptr;

    while (currptr != NULL)

    {

        nextptr = currptr->next;

        currptr->next = prevptr;

        prevptr = currptr;

        currptr = nextptr;

    }

    return prevptr;

}

Node \*reverseRecursive(Node \*&head)

{

    if (head == NULL || head->next == NULL)

    {

        return head;

    }

    Node \*newhead = reverseRecursive(head->next);

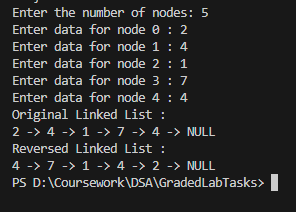
    head->next->next = head;

    head->next = NULL;

    return newhead;

}

Output:



Task 03

//Taaha Hussain Khan

//L1F21BSCS0917

//D12

#include <iostream>

using namespace std;

class Node {

public:

    int data;

    Node\* next;

};

void removeDuplicates(Node\* head)

{

    Node\* current = head;

    Node\* next\_next;

    if (current == NULL)

        return;

    while (current->next != NULL) {

        if (current->data == current->next->data) {

            next\_next = current->next->next;

            free(current->next);

            current->next = next\_next;

        }

        else

        {

            current = current->next;

        }

    }

}

void insert(Node\*\* head\_ref, int new\_data)

{

    Node\* new\_node = new Node();

    new\_node->data = new\_data;

    new\_node->next = (\*head\_ref);

    (\*head\_ref) = new\_node;

}

void printList(Node\* node)

{

    while (node != NULL) {

        cout << " " << node->data;

        node = node->next;

    }

}

int main()

{

    Node\* head = NULL;

    int n ,num ;

    cout << "Enter the number of nodes:" ;

    cin >> n;

    for (int i = 0; i < n; i++)

    {

        /\* code \*/

        cout << "Enter value " << i << " : " ;

        cin >> num ;

    insert(&head, num);

    }

    cout << "Original Linked list :" << endl;

    printList(head);

    removeDuplicates(head);

    cout << endl ;

    cout << "Linked List without duplicates :" << endl;

    printList(head);

}

