|  |  |
| --- | --- |
| Name | Taaha Hussain Khan |
| Roll No | L1F21BSCS0917 |
| Section | D21 |

Question: 01

//Taaha Hussain Khan

//L1F21BSCS0917

//D12

#include <iostream>

using namespace std;

class Node {

    public:

    int data;

    Node\* next;

};

class LinkedList {

public:

    LinkedList() {

        head = nullptr;

    }

    ~LinkedList() {

        Node\* current = head;

        Node\* next = nullptr;

        while (current != nullptr) {

            next = current->next;

            delete current;

            current = next;

        }

    }

    void insertAtBeginning(int data) {

        Node\* newNode = new Node;

        newNode->data = data;

        newNode->next = head;

        head = newNode;

    }

    void deleteMiddle() {

        if (head == nullptr) {

            return;

        }

        Node\* slowPtr = head;

        Node\* fastPtr = head;

        Node\* prev = nullptr;

        while (fastPtr != nullptr && fastPtr->next != nullptr) {

            fastPtr = fastPtr->next->next;

            prev = slowPtr;

            slowPtr = slowPtr->next;

        }

        if (fastPtr == nullptr) {

            prev->next = slowPtr->next;

            delete slowPtr;

        } else {

            Node\* next = slowPtr->next;

            slowPtr->data = next->data;

            slowPtr->next = next->next;

            delete next;

        }

    }

    void print() {

        Node\* current = head;

        while (current != nullptr) {

            cout << current->data << " ";

            current = current->next;

        }

        cout << endl;

    }

private:

    Node\* head;

};

int main() {

    LinkedList list;

    int n=0;

    cout << "Number of inputs: ";

    cin >> n;

    for (int i=0 ; i < n ; i++)

    {

        int num;

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

        cin >> num;

        list.insertAtBeginning(num);

    }

    cout << "Original list: ";

    list.print();

    list.deleteMiddle();

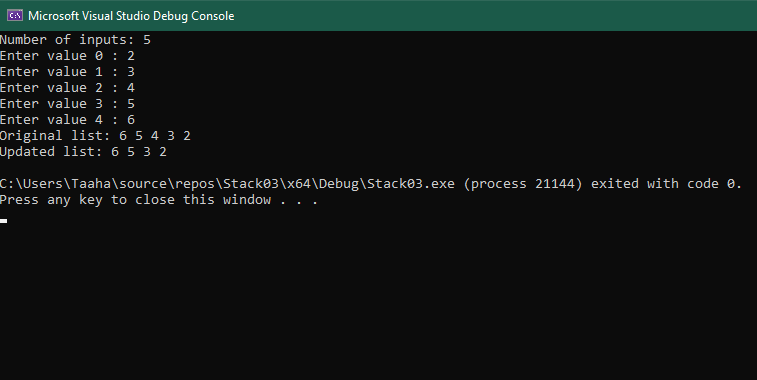
    cout << "Updated list: ";

    list.print();

    return 0;

}

Output:



Question 02

//Taaha Hussain Khan

//L1F21BSCS0917

//D12

#include <iostream>

#include <string>

using namespace std;

struct Node

{

    string name;

    int id;

    int year\_of\_joining;

    Node \*next;

};

class LinkedList

{

private:

    Node \*head;

public:

    LinkedList()

    {

        head = NULL;

    }

    void addEmployee(string name, int id, int year\_of\_joining)

    {

        Node \*newNode = new Node();

        newNode->name = name;

        newNode->id = id;

        newNode->year\_of\_joining = year\_of\_joining;

        newNode->next = head;

        head = newNode;

    }

  void removeDuplicates()

    {

        Node \*current = head;

        while (current != NULL)

        {

            Node \*runner = current;

            while (runner->next != NULL)

            {

                if (current->id == runner->next->id)

                {

                    Node \*duplicate = runner->next;

                    runner->next = runner->next->next;

                    delete duplicate;

                }

                else

                {

                    runner = runner->next;

                }

            }

            current = current->next;

        }

    }

    void sort()

    {

        Node \*current = head;

        Node \*runner = NULL;

        while (current != NULL)

        {

            runner = current->next;

            while (runner != NULL)

            {

                if (current->year\_of\_joining > runner->year\_of\_joining)

                {

                    string temp\_name = current->name;

                    int temp\_id = current->id;

                    int temp\_year\_of\_joining = current->year\_of\_joining;

                    current->name = runner->name;

                    current->id = runner->id;

                    current->year\_of\_joining = runner->year\_of\_joining;

                    runner->name = temp\_name;

                    runner->id = temp\_id;

                    runner->year\_of\_joining = temp\_year\_of\_joining;

                }

                runner = runner->next;

            }

            current = current->next;

        }

    }

    void display()

    {

        Node \*current = head;

        while (current != NULL)

        {

            cout << "Name: " << current->name << endl;

            cout << "ID: " << current->id << endl;

            cout << "Year of joining: " << current->year\_of\_joining << endl;

            cout << endl;

            current = current->next;

        }

    }

};

int main()

{

    LinkedList list;

    int choice;

    char opt;

    do{

    cout << "1. Insert a new Employee. " << endl;

    cout << "2. Find and remove duplicate employee record. " << endl;

    cout << "3. Sort the linked list with respect to the year of joining" << endl;

    cout << "4. Display the employee records" << endl;

    cin >> choice ;

    if (choice == 1)

    {

        string name;

        int id, year, n, i = 0;

        cout << "Number of Employee :";

        cin >> n;

        while (i < n)

        {

            cout << "Name: ";

            cin >> name;

            cout << "ID: ";

            cin >> id;

            cout << "Year of Joining: ";

            cin >> year;

            list.addEmployee(name, id, year);

            i++;

        }

    }

    else if( choice == 2)

       { list.removeDuplicates();

       cout << "Value Removed Successfully !" << endl;

       list.display();

       }

    else if( choice == 3)

        {

            list.sort();

            cout << "List sorted successfully! "<<endl;

                    }

    else if( choice == 4){

        list.display();

    }

    cout << "Y/y OR N/n" << endl;

    cin >> opt;

    if (opt == 'y' || opt == 'Y')

        continue;

    else

        break;

    }while(true);

    return 0;

}

