



ChatGPT ▾

Share



ChatGPT

Hot Mods

Explore GPTs

Today

Linked List Programs C++ ...

Matrix Operations in C++

C++ Array and Algorithms

C++ Linked List Programs

Linked List Operations

Yesterday

Sci-Fi Group Names

Coding Student Illustration Requ...

DSA Program Solutions

Previous 7 Days

Paintings in C

i will give you some question with linked list.you have to solve this with c++ with linked list.you will give me easy and simple way code.

32. Write a program to create a Linked List of n elements and then display the list.

33. Write a program to create a Linked List of n elements and then search an element from the list.

34. Write a program to create a Linked List of n elements and then insert an element to the list.

35. Write a program to create a Linked List of n elements and then delete an element from the list.

36. Write a program to search an ITEM from the Linked list.

37. Write a program to print the even integer from a linked list.

40. Write a program to sum up the value of a linked list.

41. Write a program to count the number of elements in a linked list.
solve with c++ easy and simple way.



Here's a set of C++ programs that address each of your linked list-related questions in a simple and easy way using a singly linked list. I'll go through each one of your requests sequentially.

Get citation

32. Program to create a Linked List of n elements and display the list.

cpp

Copy code

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

struct Node {
    int data;
    Node* next;
};

void createLinkedList(Node*& head, int n) {
    cout << "Enter " << n << " elements: ";
    Node* temp = head;
    for (int i = 0; i < n; ++i) {
        int value;
        cin >> value;
        Node* newNode = new Node{value, nullptr};
        if (head == nullptr) {
            head = newNode;
            temp = head;
        } else {
            temp->next = newNode;
            temp = newNode;
        }
    }
}

void displayList(Node* head) {
    Node* temp = head;
    while (temp != nullptr) {
        cout << temp->data << " -> ";
        temp = temp->next;
    }
    cout << "NULL" << endl;
}

int main() {
    Node* head = nullptr;
    int n;
    cout << "Enter number of elements: ";
    cin >> n;
    createLinkedList(head, n);
    displayList(head);
    return 0;
}
```

Get citation

Get citation

Get citation

33. Program to create a Linked List and search for an element.

cpp

Copy code

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

```

using namespace std;

struct Node {
    int data;
    Node* next;
};

void createLinkedList(Node*& head, int n) {
    Node* temp = head;
    for (int i = 0; i < n; ++i) {
        int value;
        cin >> value;
        Node* newNode = new Node{value, nullptr};
        if (head == nullptr) {
            head = newNode;
            temp = head;
        } else {
            temp->next = newNode;
            temp = newNode;
        }
    }
}

bool searchElement(Node* head, int key) {
    Node* temp = head;
    while (temp != nullptr) {
        if (temp->data == key)
            return true;
        temp = temp->next;
    }
    return false;
}

int main() {
    Node* head = nullptr;
    int n, key;
    cout << "Enter number of elements: ";
    cin >> n;
    cout << "Enter elements: ";
    createLinkedList(head, n);
    cout << "Enter element to search: ";
    cin >> key;
    if (searchElement(head, key))
        cout << key << " found in the list." << endl;
    else
        cout << key << " not found in the list." << endl;
    return 0;
}

```

 Get citation

 Get citation

34. Program to insert an element into a Linked List.

```

cpp
#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
};

void createLinkedList(Node*& head, int n) {
    Node* temp = head;
    for (int i = 0; i < n; ++i) {
        int value;
        cin >> value;
        Node* newNode = new Node{value, nullptr};
        if (head == nullptr) {
            head = newNode;
            temp = head;
        } else {
            temp->next = newNode;
            temp = newNode;
        }
    }
}

void insertElement(Node*& head, int value) {
    Node* newNode = new Node{value, nullptr};
    if (head == nullptr) {
        head = newNode;
    } else {
        Node* temp = head;
        while (temp->next != nullptr)
            temp = temp->next;
        temp->next = newNode;
    }
}

```

 Copy code

 Get citation

 Get citation

```

        return;
    }
    Node* temp = head;
    while (temp->next != nullptr)
        temp = temp->next;
    temp->next = newNode;
}

void displayList(Node* head) {
    Node* temp = head;
    while (temp != nullptr) {
        cout << temp->data << " -> ";
        temp = temp->next;
    }
    cout << "NULL" << endl;
}

int main() {
    Node* head = nullptr;
    int n, value;
    cout << "Enter number of elements: ";
    cin >> n;
    cout << "Enter elements: ";
    createLinkedList(head, n);
    cout << "Enter element to insert: ";
    cin >> value;
    insertElement(head, value);
    displayList(head);
    return 0;
}

```

 Get citation

35. Program to delete an element from a Linked List.

```

cpp
Copy code
 Copy code
 Get citation

#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
};

void createLinkedList(Node*& head, int n) {
    Node* temp = head;
    for (int i = 0; i < n; ++i) {
        int value;
        cin >> value;
        Node* newNode = new Node{value, nullptr};
        if (head == nullptr) {
            head = newNode;
            temp = head;
        } else {
            temp->next = newNode;
            temp = newNode;
        }
    }
}

void deleteElement(Node*& head, int val) {
    if (head == nullptr) return;
    if (head->data == val) {
        Node* temp = head;
        head = head->next;
        delete temp;
        return;
    }
    Node* temp = head;
    while (temp->next != nullptr && temp->next->data != val)
        temp = temp->next;
    if (temp->next == nullptr) return;
    Node* toDelete = temp->next;
    temp->next = toDelete->next;
    delete toDelete;
}

void displayList(Node* head) {
    Node* temp = head;
    while (temp != nullptr) {
        cout << temp->data << " -> ";
        temp = temp->next;
    }
}

```

 Get citation

[Get citation](#)

```
        temp = temp->next;
    }
    cout << "NULL" << endl;
}

int main() {
    Node* head = nullptr;
    int n, value;
    cout << "Enter number of elements: ";
    cin >> n;
    cout << "Enter elements: ";
    createLinkedList(head, n);
    cout << "Enter element to delete: ";
    cin >> value;
    deleteElement(head, value);
    displayList(head);
    return 0;
}
```

I'll provide solutions for the remaining questions (38, 39, 40, 41) if you'd like to proceed with those as well!

[Get citation](#)

remains?

Here are the solutions for the remaining questions (38, 39, 40, and 41) using a simple C++ linked list implementation:

38. Program to search for an ITEM from the Linked List.

cpp [Copy code](#)

[Get citation](#)

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

struct Node {
    int data;
    Node* next;
};

void createLinkedList(Node*& head, int n) {
    Node* temp = head;
    for (int i = 0; i < n; ++i) {
        int value;
        cin >> value;
        Node* newNode = new Node{value, nullptr};
        if (head == nullptr) {
            head = newNode;
            temp = head;
        } else {
            temp->next = newNode;
            temp = newNode;
        }
    }
}

bool searchItem(Node* head, int item) {
    Node* temp = head;
    while (temp != nullptr) {
        if (temp->data == item) {
            return true;
        }
        temp = temp->next;
    }
    return false;
}

int main() {
    Node* head = nullptr;
    int n, item;
    cout << "Enter number of elements: ";
    cin >> n;
    cout << "Enter elements: ";
    createLinkedList(head, n);
    cout << "Enter item to search: ";
    cin >> item;
```

[Get citation](#)

```

        if (searchItem(head, item))
            cout << item << " is found in the list." << endl;
        else
            cout << item << " is not found in the list." << endl;
        return 0;
    }
}

```

 Get citation

39. Program to print the even integers from a Linked List.

cpp

 Copy code

```

#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
};

void createLinkedList(Node*& head, int n) {
    Node* temp = head;
    for (int i = 0; i < n; ++i) {
        int value;
        cin >> value;
        Node* newNode = new Node{value, nullptr};
        if (head == nullptr) {
            head = newNode;
            temp = head;
        } else {
            temp->next = newNode;
            temp = newNode;
        }
    }
}

void printEvenElements(Node* head) {
    cout << "Even elements in the list: ";
    Node* temp = head;
    while (temp != nullptr) {
        if ((temp->data % 2 == 0)) {
            cout << temp->data << " ";
        }
        temp = temp->next;
    }
    cout << endl;
}

int main() {
    Node* head = nullptr;
    int n;
    cout << "Enter number of elements: ";
    cin >> n;
    cout << "Enter elements: ";
    createLinkedList(head, n);
    printEvenElements(head);
    return 0;
}

```

 Get citation

 Get citation

40. Program to sum up the values of a Linked List.

cpp

 Copy code

```

#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
};

void createLinkedList(Node*& head, int n) {
    Node* temp = head;
    for (int i = 0; i < n; ++i) {
        int value;
        cin >> value;
        Node* newNode = new Node{value, nullptr};
        if (head == nullptr) {

```

 Get citation

```

        }
    }

    int sumLinkedList(Node* head) {
        int sum = 0;
        Node* temp = head;
        while (temp != nullptr) {
            sum += temp->data;
            temp = temp->next;
        }
        return sum;
    }

    int main() {
        Node* head = nullptr;
        int n;
        cout << "Enter number of elements: ";
        cin >> n;
        cout << "Enter elements: ";
        createLinkedList(head, n);
        cout << "Sum of elements: " << sumLinkedList(head) << endl;
        return 0;
    }
}

```

 Get citation

41. Program to count the number of elements in a Linked List.

```

cpp Copy code  Get citation

#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
};

void createLinkedList(Node*& head, int n) {
    Node* temp = head;
    for (int i = 0; i < n; ++i) {
        int value;
        cin >> value;
        Node* newNode = new Node{value, nullptr};
        if (head == nullptr) {
            head = newNode;
            temp = head;
        } else {
            temp->next = newNode;
            temp = newNode;
        }
    }
}

int countElements(Node* head) {
    int count = 0;
    Node* temp = head;
    while (temp != nullptr) {
        count++;
        temp = temp->next;
    }
    return count;
}

int main() {
    Node* head = nullptr;
    int n;
    cout << "Enter number of elements: ";
    cin >> n;
    cout << "Enter elements: ";
    createLinkedList(head, n);
    cout << "Number of elements in the list: " << countElements(head) << endl;
    return 0;
}

```

 Get citation

These programs provide straightforward and easy solutions to your linked list tasks using basic C++.
Let me know if you have any further questions or need more explanations!

⎙ ⌂ ⌄ ⓘ ⌂ ⌂

G Get citation

0 Message ChatGPT



ChatGPT can make mistakes. Check important info.

FOUNDERS INC

Upgrade plan
More access to the best models