

```

// Jorge Rivas
#include <iostream>
using namespace std;

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

void insertAtBeginning(struct LLNode** head, int dataToBeInserted)
{
    struct LLNode* curr = new LLNode;
    curr->data = dataToBeInserted;
    curr->next = NULL;
    if (*head == NULL)
        *head = curr;

    else
    {
        curr->next = *head;
        *head = curr;
    }
}

void display(struct LLNode** node)
{
    struct LLNode* temp = *node;
    while (temp != NULL)
    {
        if (temp->next != NULL)
            cout << temp->data << "->";
        else
            cout << temp->data;

        temp = temp->next;
    }

    cout << endl;
}

void deleteNode(struct LLNode** head_ref, int pos)
{
    if (*head_ref == NULL)
    {
        return;
    }

    struct LLNode* temp = *head_ref;

    if (pos == 0)
    {
        *head_ref = temp->next;
        free(temp);
        return;
    }
}

```

```

    for (int i = 0; temp != NULL && i < pos - 2; i++)
    {
        temp = temp->next;
    }
    if (temp == NULL || temp->next == NULL)
    {
        return;
    }

    struct LLNode* next = temp->next->next;
    free(temp->next);
    temp->next = next;
}

void SortedInsertion(struct LLNode** head, struct LLNode* ToBeInserted)
{
    struct LLNode* curr;
    if (*head == NULL || (*head)->data >= ToBeInserted->data)
    {
        ToBeInserted->next = *head;
        *head = ToBeInserted;
    }

    else
    {
        curr = *head;
        while (curr->next != NULL && curr->next->data < ToBeInserted->data)
        {
            curr = curr->next;
        }
        ToBeInserted->next = curr->next;
        curr->next = ToBeInserted;
    }
}

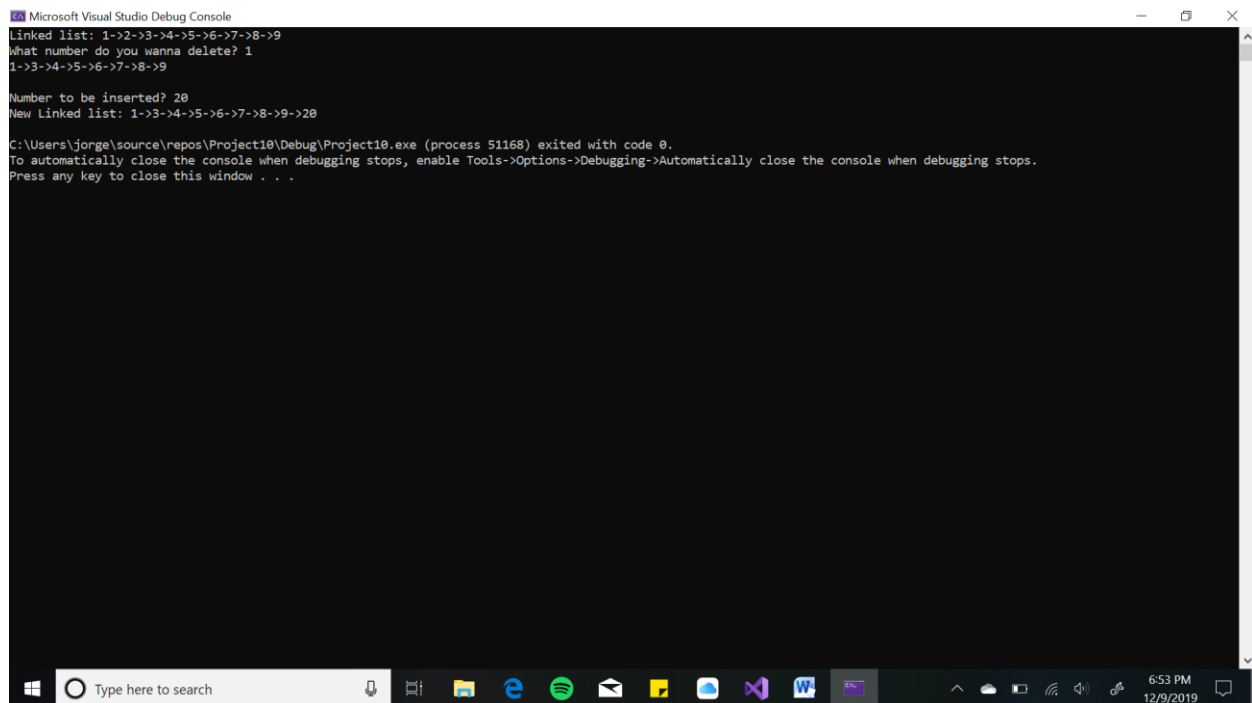
int main()
{
    struct LLNode* head = NULL;
    insertAtBeginning(&head, 9);
    insertAtBeginning(&head, 8);
    insertAtBeginning(&head, 7);
    insertAtBeginning(&head, 6);
    insertAtBeginning(&head, 5);
    insertAtBeginning(&head, 4);
    insertAtBeginning(&head, 3);
    insertAtBeginning(&head, 2);
    insertAtBeginning(&head, 1);

    int x;
    cout << "Linked list: ";
    display(&head);
    cout << "What number do you wanna delete? ";
    cin >> x;
    deleteNode(&head, x);
    display(&head);

    struct LLNode* head1 = NULL;
    insertAtBeginning(&head1, 20);
    cout << "" << endl;
}

```

```
    cout << "Number to be inserted? ";  
    display(&head1);  
    SortedInsertion(&head, head1);  
  
    cout << "New Linked list: ";  
    display(&head);  
    return 0;  
}
```



The screenshot shows the Microsoft Visual Studio Debug Console window. The output text is as follows:

```
Linked list: 1->2->3->4->5->6->7->8->9  
What number do you wanna delete? 1  
1->3->4->5->6->7->8->9  
Number to be inserted? 20  
New Linked list: 1->3->4->5->6->7->8->9->20  
C:\Users\jorge\source\repos\Project10\Debug\Project10.exe (process 51168) exited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .
```

The console window is titled "Microsoft Visual Studio Debug Console" and has standard window controls (minimize, maximize, close) in the top right corner. The bottom of the image shows the Windows taskbar with the search bar and several application icons.