

LINKEDLIST | C++

CODE

```
// HEADER FILES
```

```
#include <iostream>
#include <conio.h>
using namespace std;
```

```
// CLASS FOR LINKEDLIST
```

```
class List {
private:
```

```
// STRUCTURE
```

```
typedef struct node {
    int data;
    node *next;
} *nodePtr;
```

```
nodePtr head;
nodePtr curr;
nodePtr temp;
```

```
public:
```

```
// CONSTRUCTOR
```

```
List() {
    head = NULL;
    curr = NULL;
    temp = NULL;
}
```

```
// FUNCTION FOR ADDING NODE
```

```
void AddNode(int addData) {
    nodePtr n = new node;
    n->data = addData;
    n->next = NULL;

    if (head != NULL) {
        curr = head;
        while (curr->next != NULL) {
            curr = curr->next;
        }
        curr->next = n;
    }
    else {
        head = n;
    }
}
```

```
// FUNCTION FOR DELETING NODE
```

```
void DeleteNode(int delData) {
    nodePtr delPtr = NULL;
    temp = head;
    curr = head;
    while (curr != NULL && curr->data!=delData) {
```

LINKEDLIST | C++

CODE

```
        temp = curr;
        curr = curr->next;
    }
    if (curr == NULL) {
        cout << delData << " is not in the List.\n";
        delete delPtr;
    }
    else {
        delPtr = curr;
        curr = curr->next;
        temp->next = curr;
        if (delPtr == head) {
            head = head->next;
            temp = NULL;
        }
        delete delPtr;
        cout << " The value " << delData << " is deleted.\n";
    }
}
```

// FUNCTION FOR PRINTING LIST

```
void PrintList() {
    curr = head;
    while (curr != NULL) {
        cout << curr->data<<endl;
        curr = curr->next;
    }
}

};
```

// MAIN FUNCTION

```
int main() {
    List Usama;
mmenu:
    cout << "\n===== \n";
    cout << "\t\tMENU";
    cout << "\n===== \n";
    cout<<"1. INSERT\n2. DELETE\n3. EXIT\n";
    char choice;
    cout << "Choice: ";
    cin >> choice;
    if (choice == '1') {
        cout << "===== \n";
        int num1;
start:
        cout << "Enter a value to insert: ";
        cin >> num1;
        Usama.AddNode(num1);
        cout << "\n"<<num1<<" is Added!\n\nUpdated List: \n";
        Usama.PrintList();
        cout << "\n===== \n";
        cout << "1. Insert More Values\n2. Main Menu\n";
        cout<<"Choice: ";
        cin >> choice;
    }
}
```

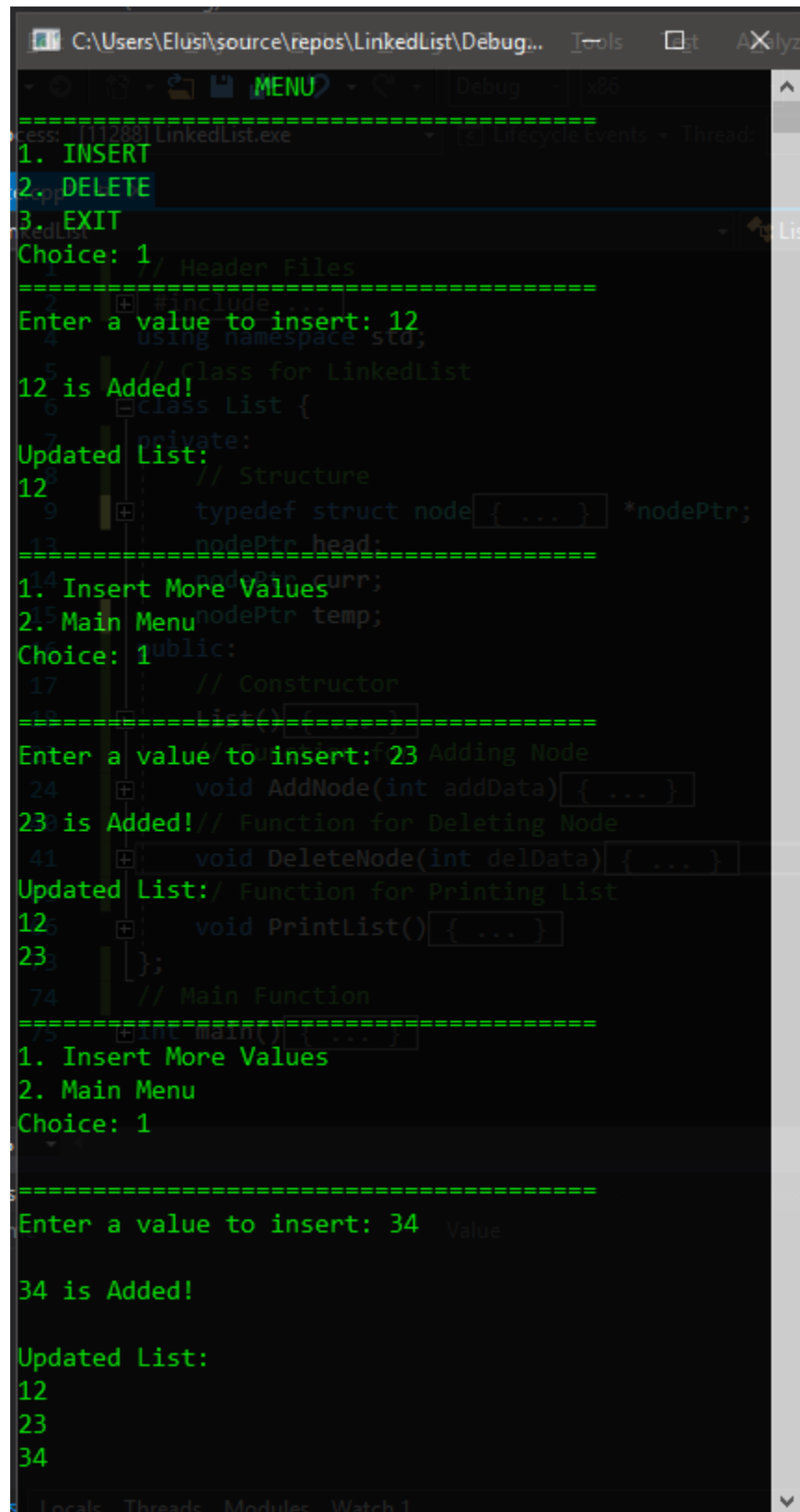
LINKEDLIST | C++

CODE

```
        cout << "\n=====\\n";
        if (choice == '1')
            goto start;
        else if (choice == '2')
            goto mmenu;
    }
    else if (choice=='2') {
        cout << "\\nList Values\\n";
        Usama.PrintList();
        int num1;
        cout << "\\nEnter a value to delete: ";
        cin >> num1;
        Usama.DeleteNode(num1);
        cout << "\\nUpdated List: \\n";
        Usama.PrintList();
        goto mmenu;
    }
    else if (choice == '3') {
        cout << "\\n=====\\n";
        cout << "    Thanks for Using our Software :)";
        cout << "\\n=====\\n";
    }
    else {
        cout << "\\nInvalid!\\n";
        goto mmenu;
    }
    _getch();
}
```

LINKEDLIST | C++

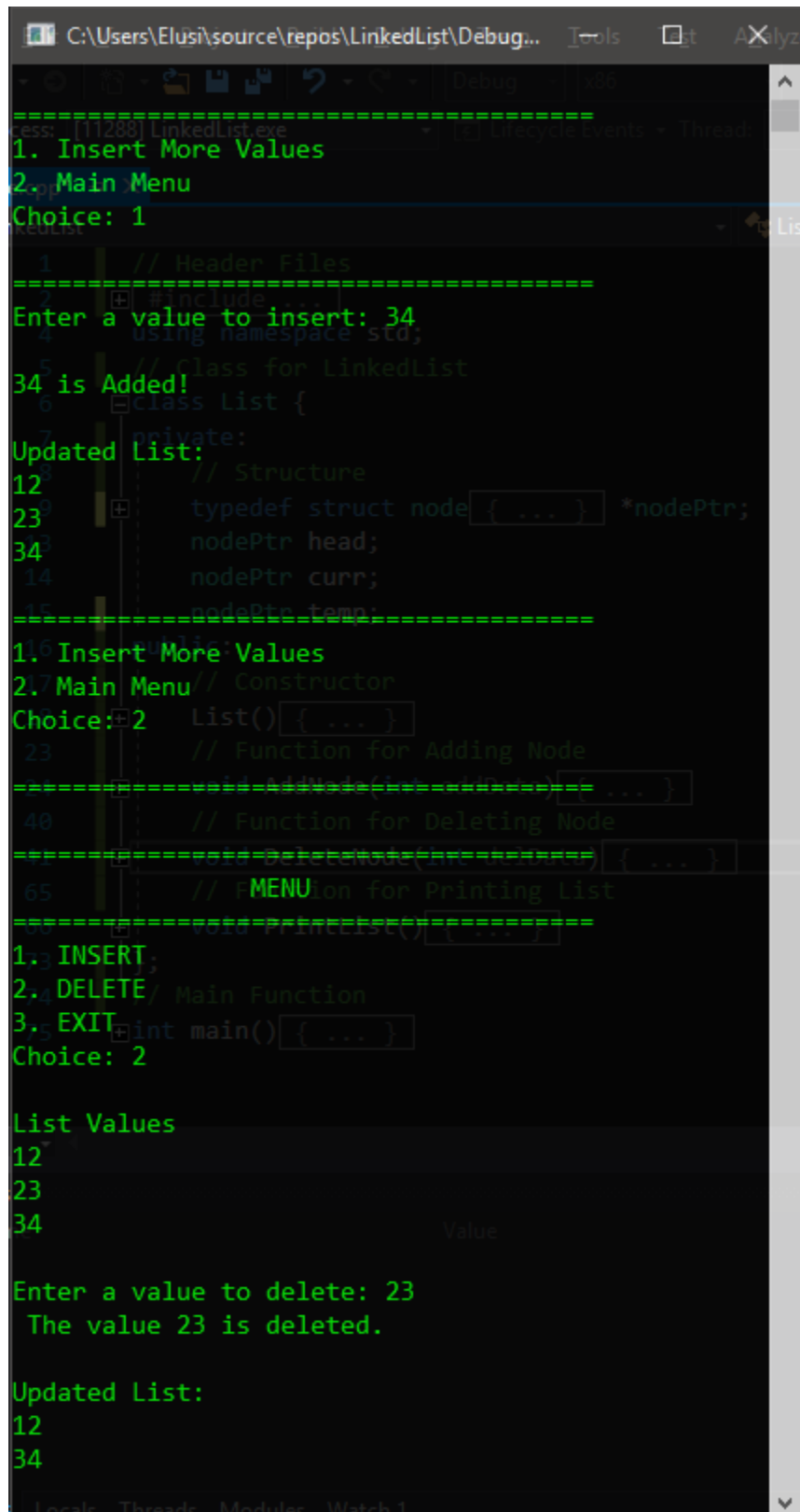
OUTPUT



```
C:\Users\Elusi\source\repos\LinkedList\Debug... Tools [X] Analyz
=====
Press: [11268] LinkedList.exe [X] Lifecycle Events + Thread:
1. INSERT
2. DELETE
3. EXIT
Choice: 1
=====
1 // Header Files
2 #include <iostream>
3 using namespace std;
4 // Class for LinkedList
5 class List {
6 private:
7 // Structure
8 typedef struct node { ... } *nodePtr;
9 nodePtr head;
10 nodePtr curr;
11 nodePtr temp;
12 public:
13 // Constructor
14 List() { ... }
15 void AddNode(int addData) { ... }
16 void DeleteNode(int delData) { ... }
17 void PrintList() { ... }
18 };
19 // Main Function
20 int main() { ... }
21
22 1. Insert More Values
23 2. Main Menu
24 Choice: 1
25
26 =====
27 Enter a value to insert: 12
28 12 is Added!
29 Updated List:
30 12
31
32 1. Insert More Values
33 2. Main Menu
34 Choice: 1
35
36 =====
37 Enter a value to insert: 23
38 23 is Added!
39 Updated List:
40 12
41 23
42
43 1. Insert More Values
44 2. Main Menu
45 Choice: 1
46
47 =====
48 Enter a value to insert: 34
49 34 is Added!
50 Updated List:
51 12
52 23
53 34
54
55 Locals Threads Modules Watch 1
```

LINKEDLIST | C++

OUTPUT



```
=====
1. Insert More Values
2. Main Menu
Choice: 1
=====
1 // Header Files
2 #include <iostream>
3 using namespace std;
4 // Class for LinkedList
5 class List {
6 private:
7 // Structure
8 typedef struct node { ... } *nodePtr;
9 nodePtr head;
10 nodePtr curr;
11 nodePtr temp;
12 =====
13 1. Insert More Values
14 2. Main Menu
15 Choice: 2
16 List() { ... }
17 // Function for Adding Node
18 void AddNode(int value) { ... }
19 // Function for Deleting Node
20 void DeleteNode(int value) { ... }
21 // MENU
22 void PrintList() { ... }
23 =====
24 1. INSERT;
25 2. DELETE
26 3. EXIT
27 Choice: 2
28 =====
29 List Values
30 12
31 23
32 34
33 Value
34
35 Enter a value to delete: 23
36 The value 23 is deleted.
37
38 Updated List:
39 12
40 34
41 =====
42 Locals: Threads: Modules: Watch:
=====
```