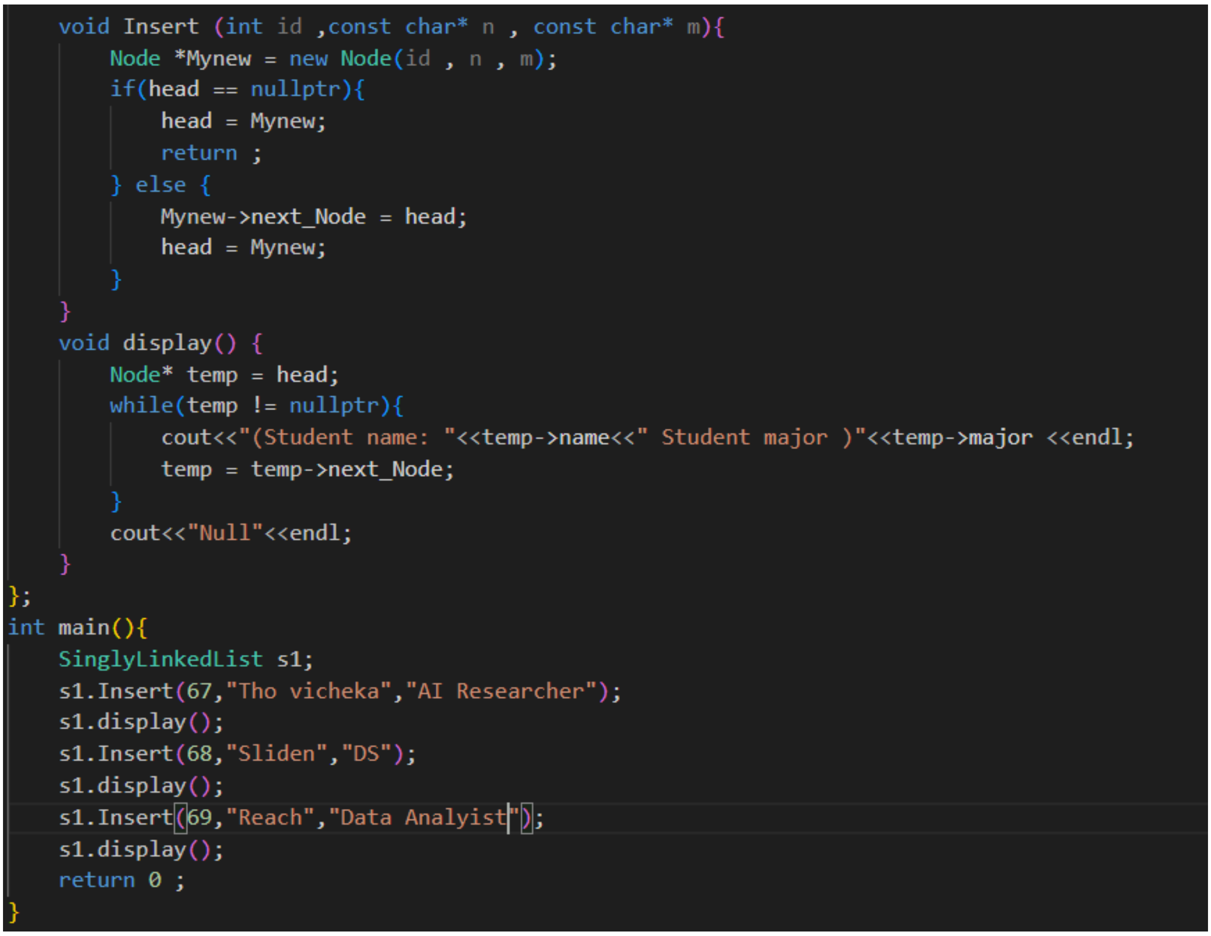
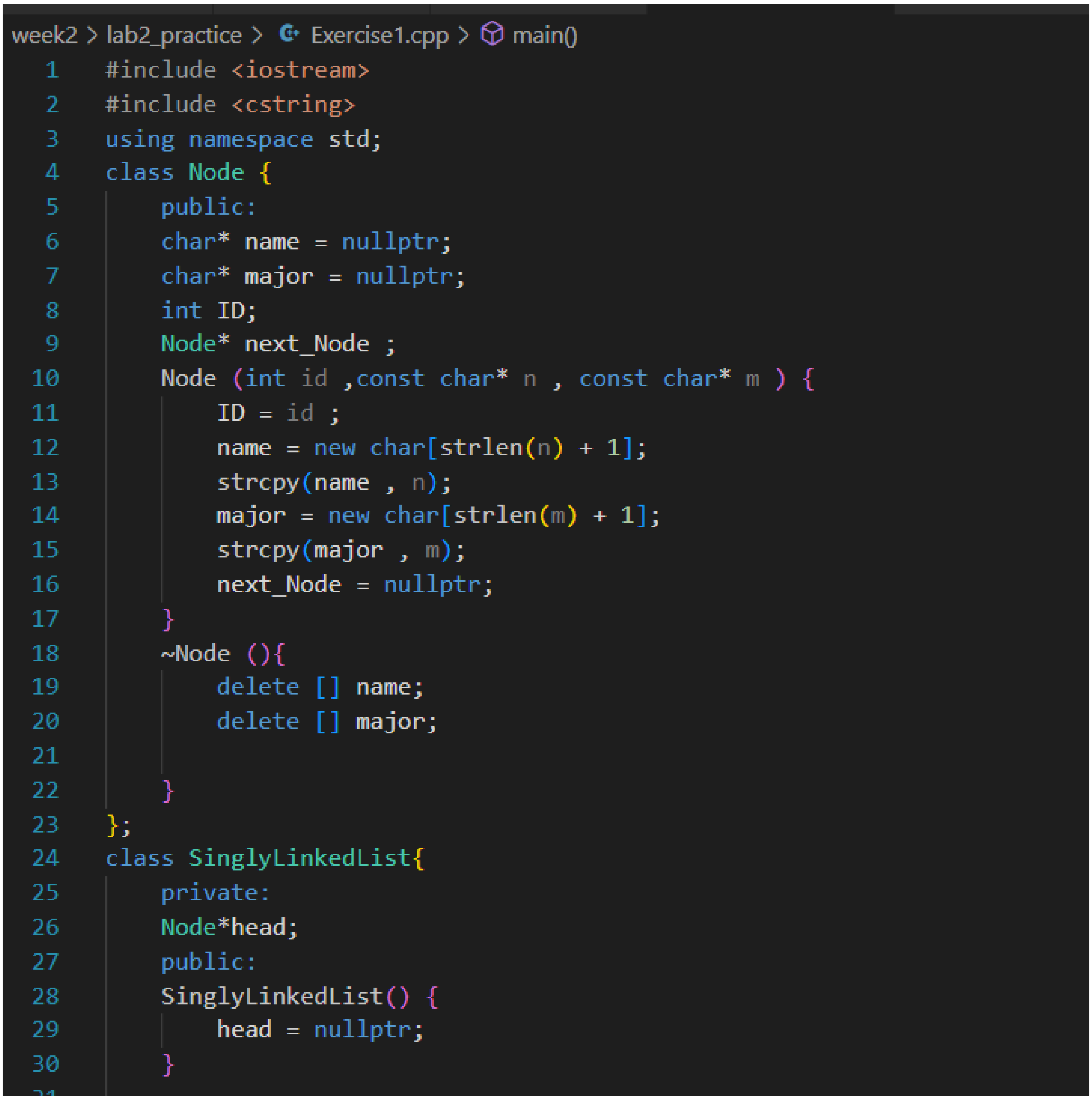
# Lab Practice Week 2

Name : Heng SovannReach

Exercise 1 :

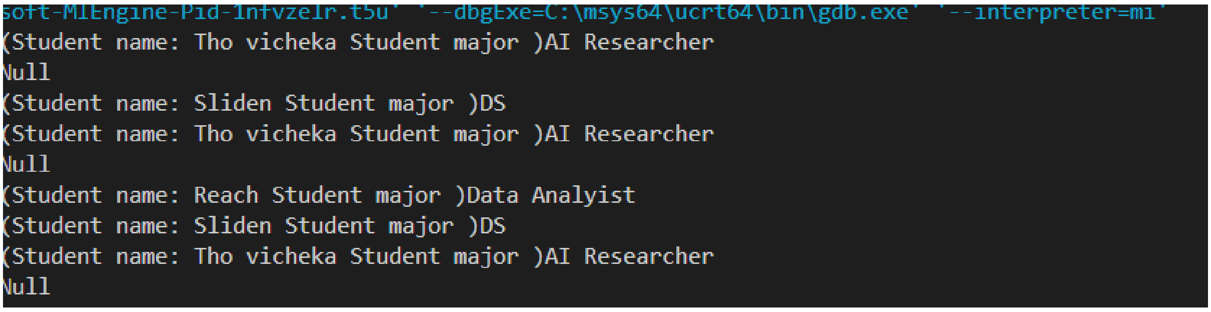


```
    void Insert (int id ,const char* n , const char* m){
        Node *Mynew = new Node(id , n , m);
        if(head == nullptr){
            head = Mynew;
            return ;
        } else {
            Mynew->next_Node = head;
            head = Mynew;
        }
    }
    void display() {
        Node* temp = head;
        while(temp != nullptr){
            cout<<"(Student name: "<<temp->name<<" Student major )"<<temp->major <<endl;
            temp = temp->next_Node;
        }
        cout<<"Null"<<endl;
    }
};
int main(){
    SinglyLinkedList s1;
    s1.Insert(67,"Tho vicheka","AI Researcher");
    s1.display();
    s1.Insert(68,"Sliden","DS");
    s1.display();
    s1.Insert(69,"Reach","Data Analyist");
    s1.display();
    return 0 ;
}
```



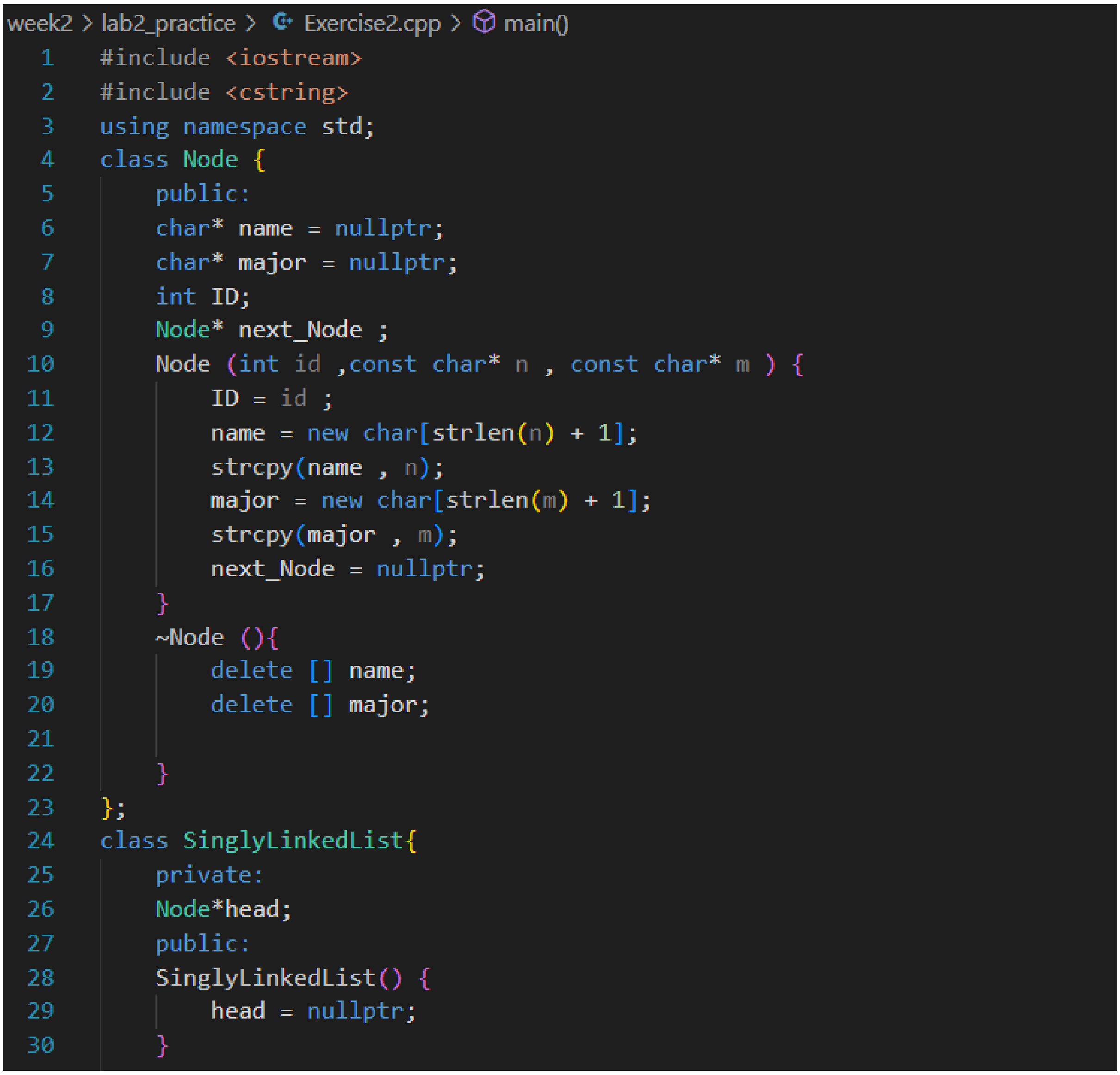
```cpp
#include <iostream>
#include <cstring>
using namespace std;
class Node {
    public:
    char* name = nullptr;
    char* major = nullptr;
    int ID;
    Node* next_Node ;
    Node (int id ,const char* n , const char* m ) {
        ID = id ;
        name = new char[strlen(n) + 1];
        strcpy(name , n);
        major = new char[strlen(m) + 1];
        strcpy(major , m);
        next_Node = nullptr;
    }
    ~Node (){
        delete [] name;
        delete [] major;

    }
};
class SinglyLinkedList{
    private:
    Node*head;
    public:
    SinglyLinkedList() {
        head = nullptr;
    }
```

Output :

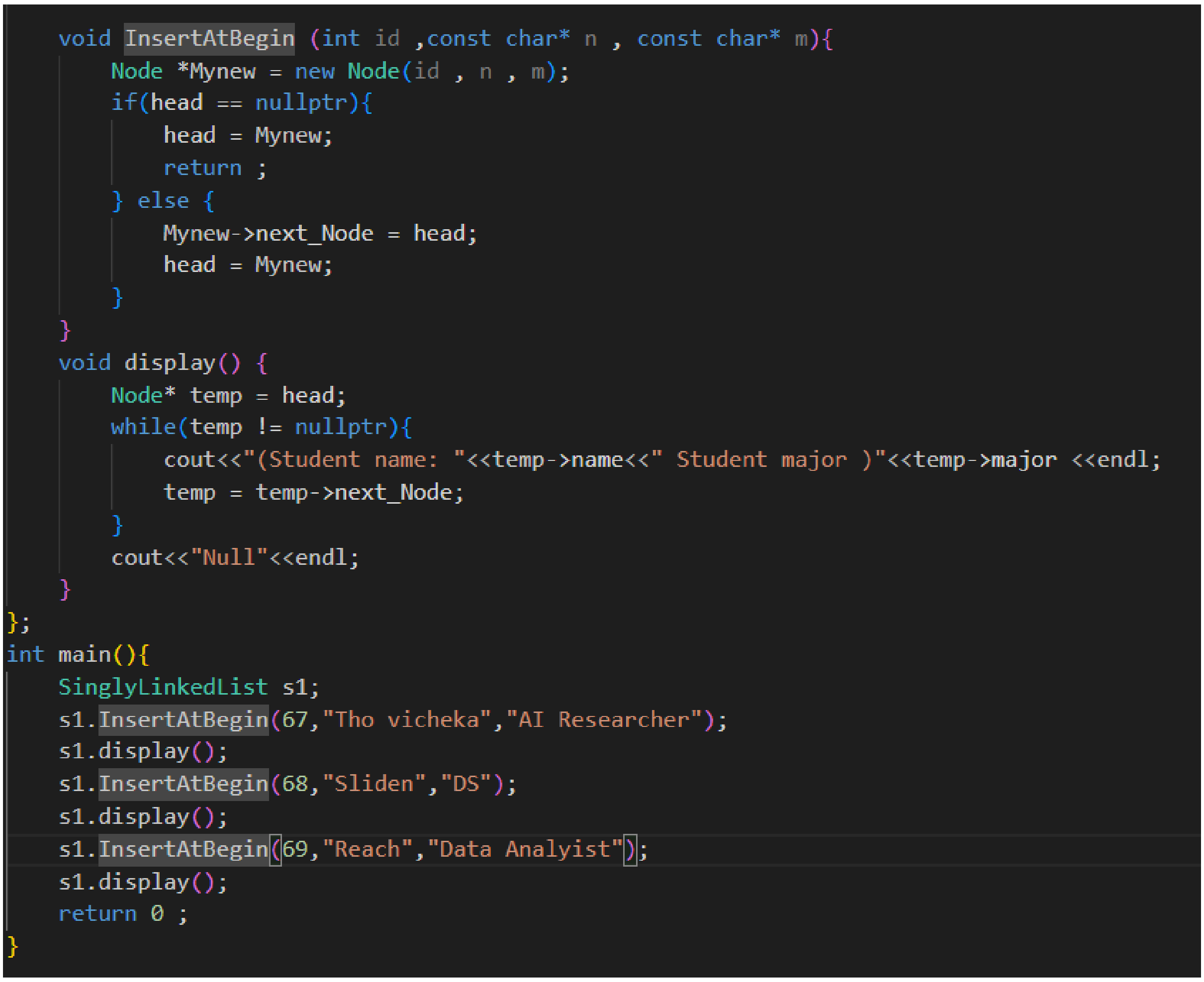


```
soft-MIEngine-Pid-1nfvzelr.t5u' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
(Student name: Tho vicheka Student major )AI Researcher
Null
(Student name: Sliden Student major )DS
(Student name: Tho vicheka Student major )AI Researcher
Null
(Student name: Reach Student major )Data Analyist
(Student name: Sliden Student major )DS
(Student name: Tho vicheka Student major )AI Researcher
Null
```

Exercise 2 :

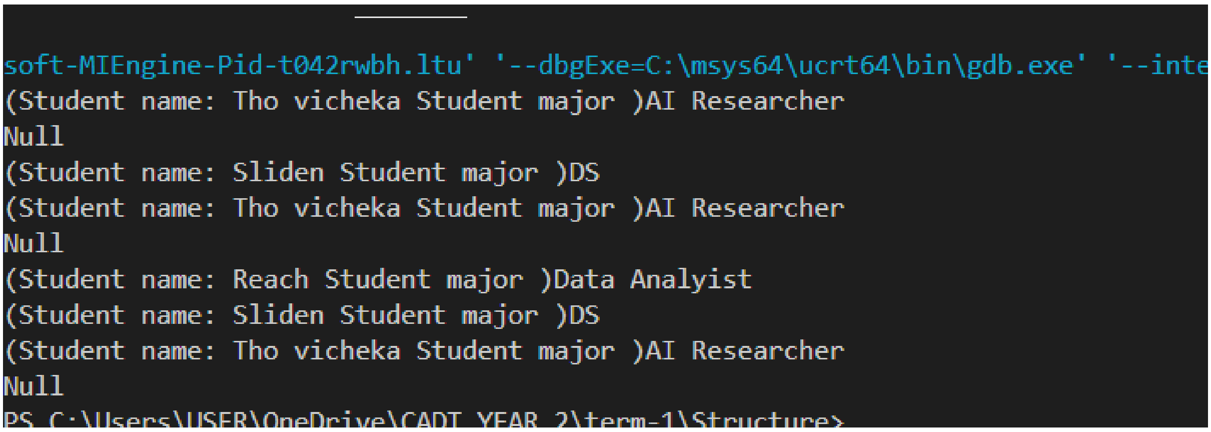


```cpp
#include <iostream>
#include <cstring>
using namespace std;
class Node {
    public:
    char* name = nullptr;
    char* major = nullptr;
    int ID;
    Node* next_Node ;
    Node (int id ,const char* n , const char* m ) {
        ID = id ;
        name = new char[strlen(n) + 1];
        strcpy(name , n);
        major = new char[strlen(m) + 1];
        strcpy(major , m);
        next_Node = nullptr;
    }
    ~Node (){
        delete [] name;
        delete [] major;

    }
};
class SinglyLinkedList{
    private:
    Node*head;
    public:
    SinglyLinkedList() {
        head = nullptr;
    }
```



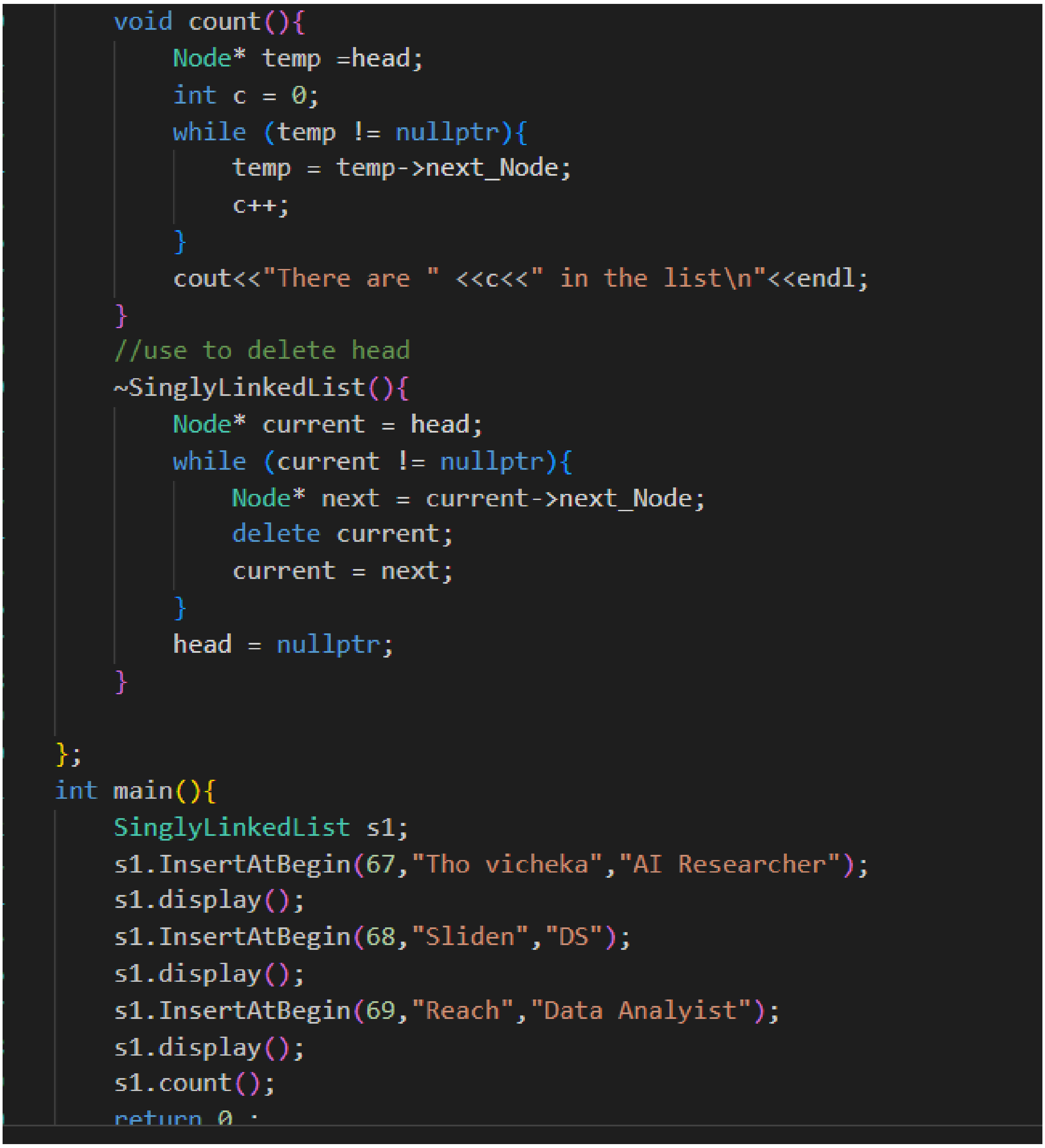
```
    void InsertAtBegin (int id ,const char* n , const char* m){
        Node *Mynew = new Node(id , n , m);
        if(head == nullptr){
            head = Mynew;
            return ;
        } else {
            Mynew->next_Node = head;
            head = Mynew;
        }
    }
    void display() {
        Node* temp = head;
        while(temp != nullptr){
            cout<<"(Student name: "<<temp->name<<" Student major )"<<temp->major <<endl;
            temp = temp->next_Node;
        }
        cout<<"Null"<<endl;
    }
};
int main(){
    SinglyLinkedList s1;
    s1.InsertAtBegin(67,"Tho vicheka","AI Researcher");
    s1.display();
    s1.InsertAtBegin(68,"Sliden","DS");
    s1.display();
    s1.InsertAtBegin(69,"Reach","Data Analyist");
    s1.display();
    return 0 ;
}
```

Output :



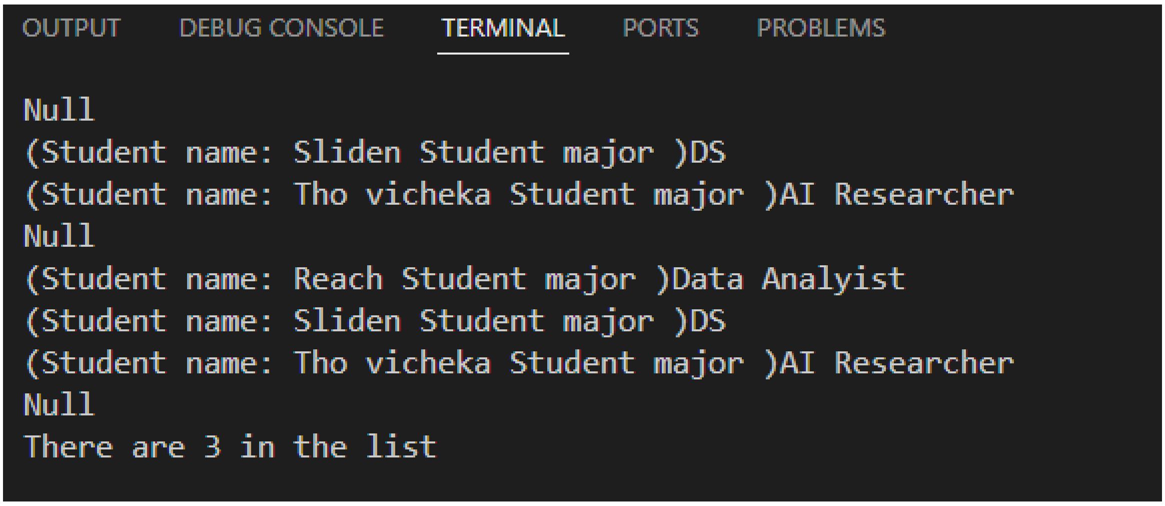
```
soft-MIEngine-Pid-t042rwbh.ltu' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--inte
(Student name: Tho vicheka Student major )AI Researcher
Null
(Student name: Sliden Student major )DS
(Student name: Tho vicheka Student major )AI Researcher
Null
(Student name: Reach Student major )Data Analyist
(Student name: Sliden Student major )DS
(Student name: Tho vicheka Student major )AI Researcher
Null
PS C:\Users\USER\OneDrive\CADT YEAR 2\term-1\Structure>
```

Exercise 3 :



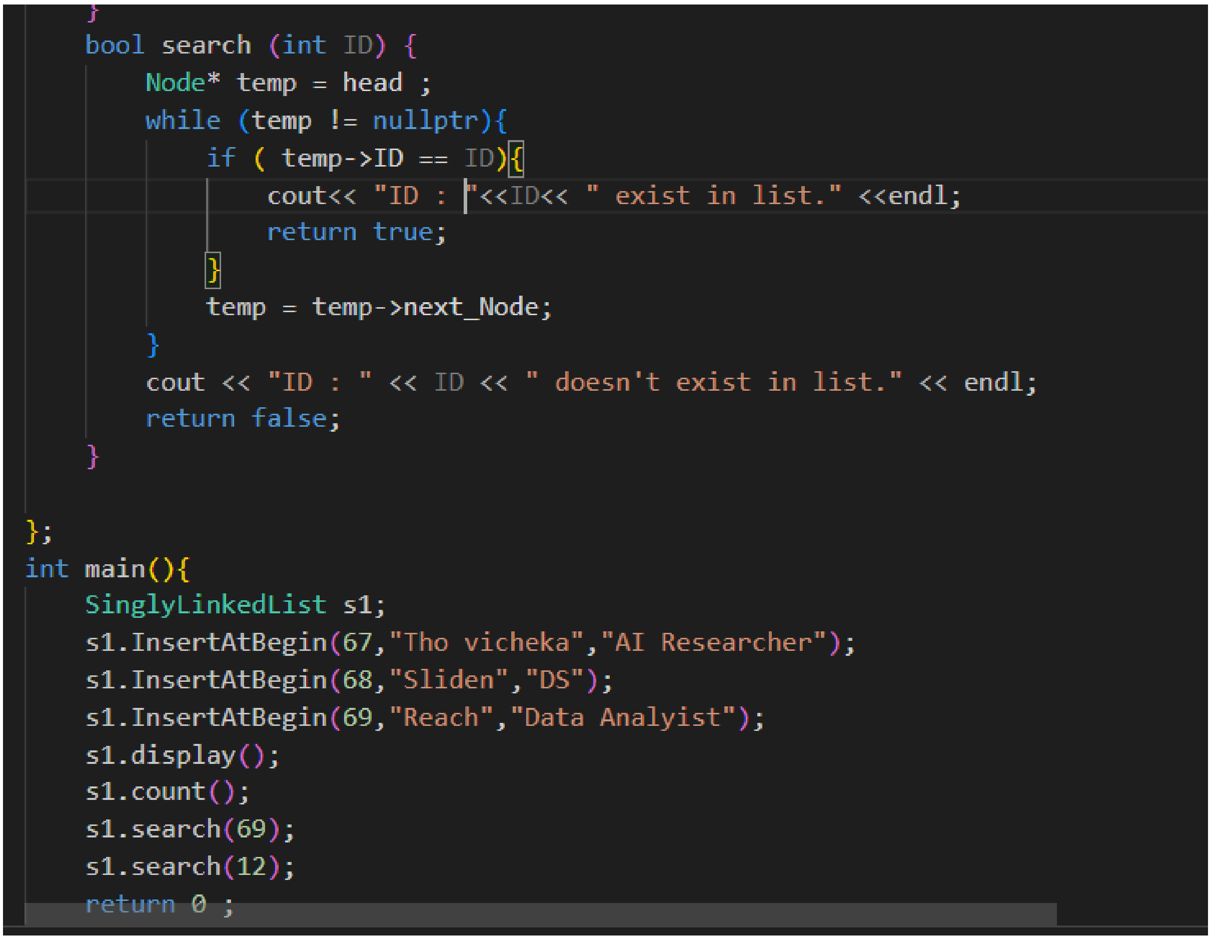
```
    void count(){
        Node* temp =head;
        int c = 0;
        while (temp != nullptr){
            temp = temp->next_Node;
            c++;
        }
        cout<<"There are " <<c<<" in the list\n"<<endl;
    }
    //use to delete head
    ~SinglyLinkedList(){
        Node* current = head;
        while (current != nullptr){
            Node* next = current->next_Node;
            delete current;
            current = next;
        }
        head = nullptr;
    }

};
int main(){
    SinglyLinkedList s1;
    s1.InsertAtBegin(67,"Tho vicheka","AI Researcher");
    s1.display();
    s1.InsertAtBegin(68,"Sliden","DS");
    s1.display();
    s1.InsertAtBegin(69,"Reach","Data Analyist");
    s1.display();
    s1.count();
    return 0 ;
```

Output :



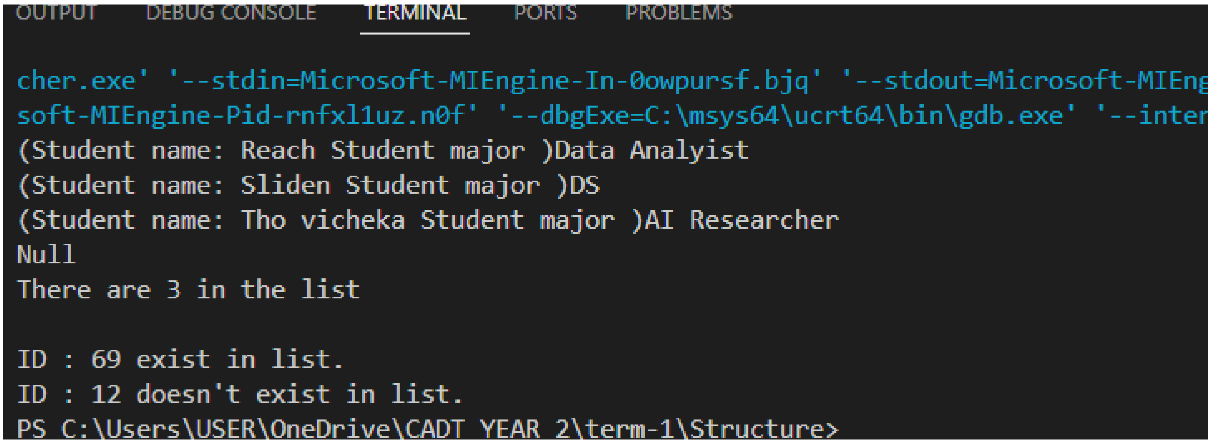
```
Null
(Student name: Sliden Student major )DS
(Student name: Tho vicheka Student major )AI Researcher
Null
(Student name: Reach Student major )Data Analyist
(Student name: Sliden Student major )DS
(Student name: Tho vicheka Student major )AI Researcher
Null
There are 3 in the list
```

Exercise 4:



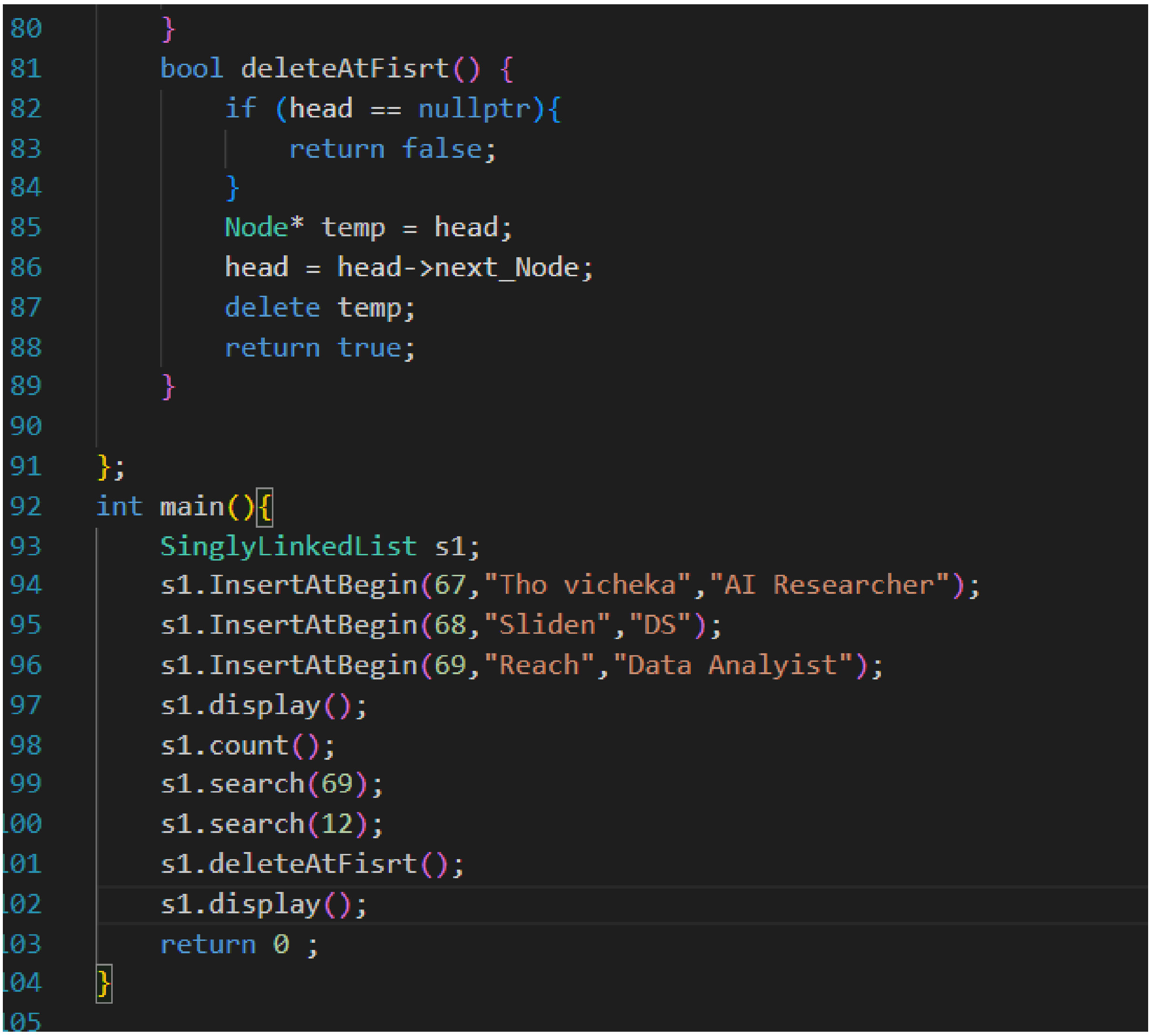
```
    }
    bool search (int ID) {
        Node* temp = head ;
        while (temp != nullptr){
            if ( temp->ID == ID){
                cout<< "ID : "<<ID<< " exist in list." <<endl;
                return true;
            }
            temp = temp->next_Node;
        }
        cout << "ID : " << ID << " doesn't exist in list." << endl;
        return false;
    }

};
int main(){
    SinglyLinkedList s1;
    s1.InsertAtBegin(67,"Tho vicheka","AI Researcher");
    s1.InsertAtBegin(68,"Sliden","DS");
    s1.InsertAtBegin(69,"Reach","Data Analyist");
    s1.display();
    s1.count();
    s1.search(69);
    s1.search(12);
    return 0 ;
```

Output :



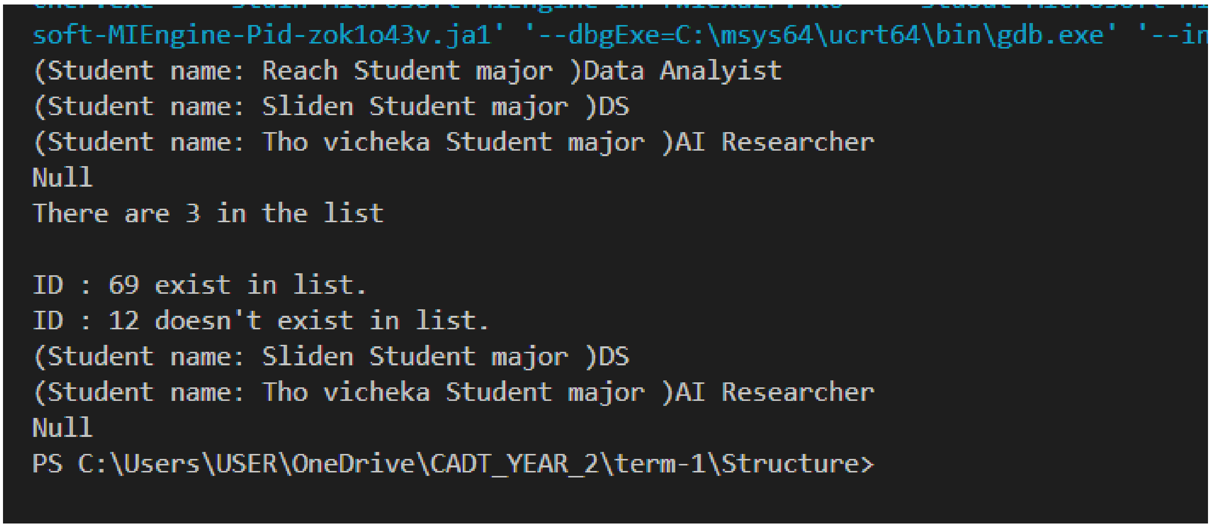
```
cher.exe' '--stdin=Microsoft-MIEngine-In-0owpursf.bjq' '--stdout=Microsoft-MIEng
soft-MIEngine-Pid-rnfxl1uz.n0f' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--inter
(Student name: Reach Student major )Data Analyist
(Student name: Sliden Student major )DS
(Student name: Tho vicheka Student major )AI Researcher
Null
There are 3 in the list

ID : 69 exist in list.
ID : 12 doesn't exist in list.
PS C:\Users\USER\OneDrive\CADT YEAR 2\term-1\Structure>
```

Exercise 5 :



```cpp
    }
    bool deleteAtFisrt() {
        if (head == nullptr){
            return false;
        }
        Node* temp = head;
        head = head->next_Node;
        delete temp;
        return true;
    }

};
int main(){
    SinglyLinkedList s1;
    s1.InsertAtBegin(67,"Tho vicheka","AI Researcher");
    s1.InsertAtBegin(68,"Sliden","DS");
    s1.InsertAtBegin(69,"Reach","Data Analyist");
    s1.display();
    s1.count();
    s1.search(69);
    s1.search(12);
    s1.deleteAtFisrt();
    s1.display();
    return 0 ;
}
```

Output:



```
soft-MIEngine-Pid-zok1o43v.ja1' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--in
(Student name: Reach Student major )Data Analyist
(Student name: Sliden Student major )DS
(Student name: Tho vicheka Student major )AI Researcher
Null
There are 3 in the list

ID : 69 exist in list.
ID : 12 doesn't exist in list.
(Student name: Sliden Student major )DS
(Student name: Tho vicheka Student major )AI Researcher
Null
PS C:\Users\USER\OneDrive\CADT_YEAR_2\term-1\Structure>
```