# Practice Problem 1

**Question:** Take a singly linked list as input and print the size of the linked list.

| Sample Input | Sample Output |
|---|---|
| 2 1 5 3 4 8 9 -1 | 7 |
| 5 1 4 5 -1 | 4 |

```cpp
#include <bits/stdc++.h>
using namespace std;
int main(){
    list<int>list_name;
    int val;
    while(cin>>val && val !=-1){
        list_name.push_back(val);
    }
    cout<<list_name.size()<<endl;
    for(int val:list_name){
        cout<<val<<" ";
    }
    return 0;
```

```
}
```

## Practice Problem 2

**Question:** Take a singly linked list as input and check if the linked list contains any duplicate value. You can assume that the maximum value will be 100.

| Sample Input | Sample Output |
|---|---|
| 5 4 8 6 2 1 -1 | NO |
| 2 4 5 6 7 4 -1 | YES |

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    list<int>list_name;
    int val;
    while(cin>>val && val != -1){
        list_name.push_back(val);
    }
    list_name.sort();
    auto it=unique(list_name.begin(),list_name.end());
```

```
    if(it !=list_name.end()){
        cout<<"YES"<<endl;
    }
    else{
        cout<<"NO";
    }
}
```

## Practice Problem 5

**Question:** Take a singly linked list as input and check if the linked list is sorted in ascending order.

| Sample Input | Sample Output |
|---|---|
| 1 5 6 8 9 -1 | YES |
| 2 4 6 5 8 4 -1 | NO |

```
#include <bits/stdc++.h>
using namespace std;
int main(){
```

```
    list<int>my_list;
    int val;
    while(cin>>val && val != -1){
        my_list.push_back(val);
    }
    list<int>my_list_2;
    my_list_2=my_list;
    my_list.sort();
    if(my_list_2==my_list){
        cout<<"YES";
    }
    else{
        cout<<"NO";
    }
    return 0;

}
```

# Practice Problem 3

**Question:** Take a singly linked list as input and print the middle element. If there are multiple values in the middle print both.

| Sample Input | Sample Output |
|---|---|
| 2 4 6 8 10 -1 | 6 |
| 1 2 3 4 5 6 -1 | 3 4 |

```cpp
#include <bits/stdc++.h>
using namespace std;
int main(){
    list<int>my_list;
    int val;
    while(cin >> val && val != -1){
        my_list.push_back(val);
    }

    int n=my_list.size();

    auto it=my_list.begin();
    advance(it,(n-1)/2);
```

```
    if(n%2==0){
        cout<< *it<<" "<<*next(it);
    }
    else{
        cout<<*it;
    }
}
```