1. (Palindrome Line)

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

class Node
{
public:

    char data;
    Node* next;

    Node()
    {
        data = '\0';
        next = NULL;
    }
};

class Stack
{
    Node* top = new Node();

public:
    Stack()
    {
        top = NULL;
    }

    bool isEmpty()
    {
        if (top == NULL)
            return true;
        else
            return false;
    }

    void push(char item)
    {
        Node* newnode = new Node();
        newnode->data = item;
        if (isEmpty())
        {
            newnode->next = NULL;
            top = newnode;
        }
        else
        {
            newnode->next = top;
            top = newnode;
        }
```

```cpp
        }

        char pop()
        {
                char poped_item = '\0';
                Node* temp = new Node();
                temp = top;
                poped_item = top->data;
                top = top->next;
                return poped_item;
                delete temp;
        }

        void display()
        {
                Node* temp = new Node();
                temp = top;
                while (temp != NULL)
                {
                        cout << temp->data << " ";
                        temp = temp->next;
                }
                delete temp;
                cout << endl;
        }
};

void check_palindrome(string string)
{
        bool isPalindrome = true;
        Stack stack;
        int i = 0;

        while (string[i] != '\0')
        {
                stack.push(string[i]);
                i++;
        }

        i = 0;
        while (string[i] != '\0')
        {
                if (stack.pop() != string[i])
                {
                        isPalindrome = false;
                        break;
                }
                i++;
        }

        if (isPalindrome == true)
        {
```

```cpp
            cout << endl << string << " is a palindrome :)\n";
        }
        else
        {
            cout << endl << string << " is not a palindrome :(\n";
        }
}

int main()
{
    string text;
    cout << "Enter a line of text to check palindrome:\n";
    cin >> text;
    check_palindrome(text);
}
```



```
Microsoft Visual Studio Debug Console

Enter a line of text to check palindrome:
tom

tom is not a palindrome :(

C:\Users\Hesham Omar\source\repos\ConsoleApplication4\x64\Debug\ConsoleAp
 0.
Press any key to close this window . . .
```



```
Microsoft Visual Studio Debug Console

Enter a line of text to check palindrome:
101loooooooooooooool101

101loooooooooooooool101 is a palindrome :)

C:\Users\Hesham Omar\source\repos\arraySum\x64\Debug\
Press any key to close this window . . .
```
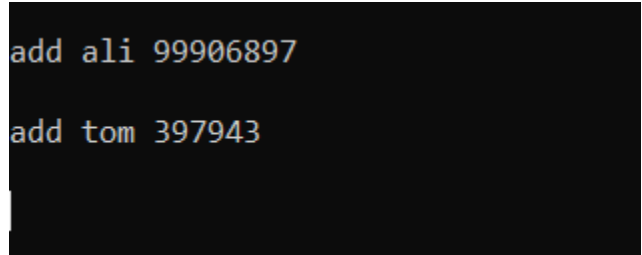
2. (Phone Book)

```cpp
#include<iostream>
#include<string>
using namespace std;
class List
{
private:
    int size = 0;
    struct Node
    {
        string name;
        long long number;
        Node* prev = NULL;
        Node* nxt = NULL;
        Node(string s, long long n) { name = s, number = n; }
    };
    Node* head = NULL;
    Node* tail = NULL;
public:
    void add(string s, long long n)
    {
        if (head == NULL)
        {
            head = new Node(s, n);
            tail = head;
        }
        else
        {
            Node* temp = new Node(s, n);
            tail->nxt = temp;
            temp->prev = tail;
            tail = temp;
        }
        size++;
    }
```



```cpp
    void remove(string s, long long n)
    {
        Node* node = head;
        if (node->name == s && node->number == n)
        {
            head = node->nxt;
            node = head;
```

```cpp
			return;

		}
		while (node != NULL)
		{
			if (node->name == s && node->number == n)
			{
				node->prev->nxt = node->nxt;
				break;
			}
			node = node->nxt;

		}
	}
```

```
remove ali 1

print

Entry: #1        ali      99906897
Entry: #2        tom      397943
Entry: #3        kim      478392
Entry: #4        ali      9933
```

```cpp
	string search(string s/*, long long n*/)
	{
		int id = 1;
		int found = 0;
		Node* node = head;
		while (node != NULL)
		{
			if (node->name == s /*&& node->number == n*/)
			{
				cout << "Entry: # " << to_string(id) << "\t";
				cout << node->name << " ";
				cout << node->number;
				cout << endl;
				found++;
			}
			node = node->nxt;
			id++;
		}
		if (found ==0)
		{
			return "NOT FOUND";
		}
		else
		{
			return"";
```

```
            }

        }
```

```
add ali 99906897

add tom 397943

add kim 478392

add ali 9933

add ali 1

search ali
Entry: # 1        ali 99906897
Entry: # 4        ali 9933
Entry: # 5        ali 1
```

```cpp
        void print()
        {
                int id = 1;
                Node* node = head;
                while (node != NULL)
                {
                        cout << "Entry: #" << id << "\t" << node->name <<
                                "\t" << node->number << "\n";
                        node = node->nxt;
                        id++;
                }
        }
};
void print_operations()
{
        cout << "-+-+-+-+-+-+-+-+-+-+-\n";
        cout << "operations:\n";
        cout << "add (name, number)\n";
        cout << "remove (name, number)\n";
        cout << "search (name, number)\n";
        cout << "print\n";
        cout << "exit\n";
        cout << "-+-+-+-+-+-+-+-+-+-+-\n\n";
```

```
}
```
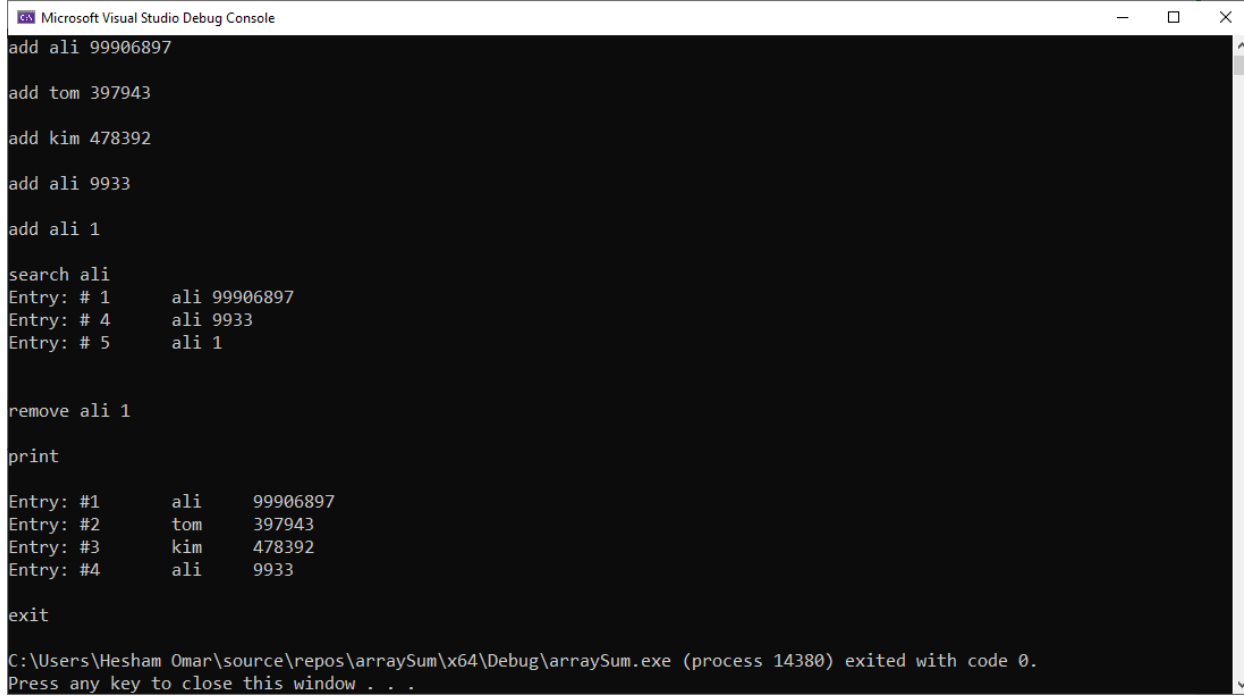
```cpp
int main()
{
    List phonebook;
    print_operations();
    while (1)
    {
        string op, name;
        long long number;
        cin >> op;
        if (op == "add")
        {
            cin >> name >> number;
            phonebook.add(name, number);
            cout << "\n";
        }
        else if (op == "remove")
        {
            cin >> name >> number;
            phonebook.remove(name, number);
            cout << "\n";
        }
        else if (op == "search")
        {
            cin >> name /*>> number;*/;
            cout << phonebook.search(name/*, number*/) << "\n\n";
        }
        else if (op == "print")
        {
            cout << "\n";
            phonebook.print();
            cout << "\n";
        }
        else if (op == "exit")
        {
            exit(0);
        }
        else
            cout << "invalid input\n";
```

```
        }
        return 0;
}
```

Microsoft Visual Studio Debug Console   —   □   ×

```
add ali 99906897

add tom 397943

add kim 478392

add ali 9933

add ali 1

search ali
Entry: # 1        ali 99906897
Entry: # 4        ali 9933
Entry: # 5        ali 1


remove ali 1

print

Entry: #1         ali       99906897
Entry: #2         tom       397943
Entry: #3         kim       478392
Entry: #4         ali       9933

exit

C:\Users\Hesham Omar\source\repos\arraySum\x64\Debug\arraySum.exe (process 14380) exited with code 0.
Press any key to close this window . . .
```