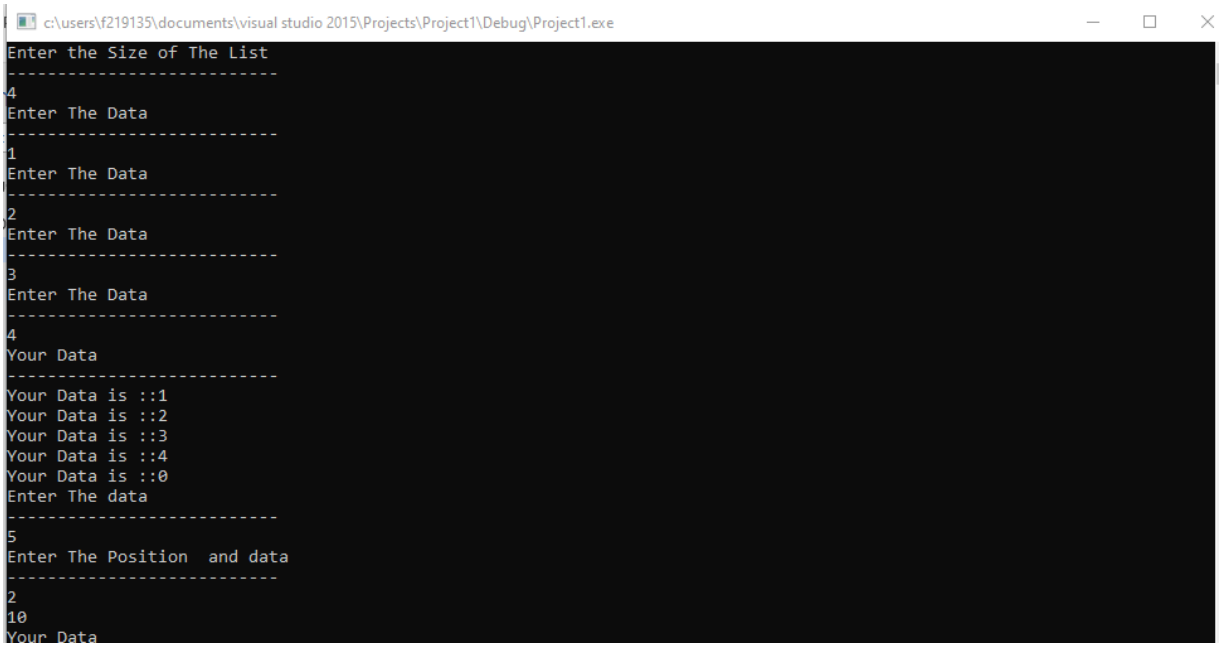
QUESTION NO 1

```
#include<iostream>
using namespace std;
struct node
{
	int data;
	node* next;
};
class LL
{
	node *head = nullptr;
	node *curr;
public:
	LL()
	{
		head = nullptr;
		curr = nullptr;
	}
	bool isempty() {}
	node* create_node(int value, node *temp) {
		node *temp2;
		if (head == nullptr)
		{
			temp = new node;
			head = temp;
			temp->data = value;
			temp2 = temp;
			temp = new node;
			temp2->next = temp;
			temp->next = nullptr;
		}
		else {
			temp->data = value;
			temp2 = temp;
			curr = temp;
			temp = new node;
			temp2->next = temp;
			temp->data = 0;
			temp->next = nullptr;
		}
		return temp;
	}
	void insert_pos(int data) {
		node*temp = curr;
		temp = temp->next;
		temp->data = data;
	}
	void middle(int pos,int data) {
		node*temp2;
		node*temp = head;
		int counter = 1;
		temp2 = temp;
		temp = temp->next;
		node*temp3;
		if (pos == 0) {
			temp3 = new node;
```

```
                        temp3->data = data;
                        temp2->next = temp3;
                        temp3->next = temp;
                }
                else {
                        while (true) {
                                if (counter == pos) {
                                        temp3 = new node;
                                        temp3->data = data;
                                        temp2->next = temp3;
                                        temp3->next = temp;
                                        break;
                                } {
                                        counter = counter + 1;
                                        temp2 = temp;
                                        temp = temp->next;
                                }
                        }
                }
        }
        void display() {
                node*temp = head;
                while (temp != nullptr) {
                        cout << "Your Data is ::" << temp->data;
                        cout << endl;
                        temp = temp->next;
                }
        }
};
int main()
{
        node *temp = nullptr;
        LL obj;
        int size;
        cout << "Enter the Size of The List";
        cout << endl;
        cout << "---------------------------";
        cout << endl;
        cin >> size;
        int data;
        for (int i = 0; i < size; i++) {
                cout << "Enter The Data ";
                cout << endl;
                cout << "---------------------------";
                cout << endl;
                cin >> data;
                temp = obj.create_node(data, temp);
        }
        cout << "Your Data";
        cout << endl;
        cout << "---------------------------";
        cout << endl;
        obj.display();
        int pos;
        cout << "Enter The data";
        cout << endl;
        cout << "---------------------------";
        cout << endl;
```

```
        cin >> data;
        obj.insert_pos(data);
        cout << "Enter The Position  and data";
        cout << endl;
        cout << "---------------------------";
        cout << endl;
        cin >> pos;
        cin >> data;
        obj.middle(pos, data);
        cout << "Your Data";
        cout << endl;
        cout << "---------------------------";
        cout << endl;
        obj.display();
        system("pause");
        return 0;
}
```

OUTPUT



QUESTION NO 2

```
#include<iostream>
using namespace std;
struct node
{
	int data;
	node* next;
};
class LL
{
	node *head = nullptr;
	node *curr;
public:
	LL()
	{
		head = nullptr;
		curr = nullptr;
	}
	bool isempty() {}
	node* create_node(int value, node *temp) {
		node *temp2;
		if (head == nullptr)
		{
			temp = new node;
			head = temp;
			temp->data = value;
			temp2 = temp;
			temp = new node;
			temp2->next = temp;
			temp->next = nullptr;
		}
		else {
			temp->data = value;
			temp2 = temp;
			curr = temp;
```

```
                temp = new node;
                temp2->next = temp;
                temp->data = 0;
                temp->next = nullptr;
        }
        return temp;
}
void insert_pos(int data) {
        node*temp2;
        node*temp = head;
        temp2 = temp;
        temp = temp->next;
        node*temp3 = temp;
        temp3 = temp3->next;
        if (data == temp2->data) {
                delete temp2;
                head = temp;
        }
        else {
                while (true) {
                        if (data == temp->data) {
                                delete temp;
                                while (temp3 != nullptr) {
                                        temp2->next = temp3;
                                        temp3 = temp3->next;
                                }
                                break;
                        } {
                                temp2 = temp;
                                temp = temp->next;
                                temp3 = temp;
                                temp3 = temp3->next;
                        }
                }
        }
        }
void middle(int pos,int data) {
        node*temp2;
        node*temp = head;
        int counter = 1;
        temp2 = temp;
        temp = temp->next;
        node*temp3;
        if (pos == 0) {
                temp3 = new node;
                temp3->data = data;
                temp2->next = temp3;
                temp3->next = temp;
        }
        else {
                while (true) {
                        if (counter == pos) {
                                temp3 = new node;
                                temp3->data = data;
                                temp2->next = temp3;
                                temp3->next = temp;
                                break;
                        } {
```

```
                                counter = counter + 1;
                                temp2 = temp;
                                temp = temp->next;
                        }
                }
        }
    }
    void display() {
        node*temp = head;
        while (temp != nullptr) {
            cout << "Your Data is ::" << temp->data;
            cout << endl;
            temp = temp->next;
        }
    }
};
int main()
{
    node *temp = nullptr;
    LL obj;
    int size;
    cout << "Enter the Size of The List";
    cout << endl;
    cout << "---------------------------";
    cout << endl;
    cin >> size;
    int data;
    for (int i = 0; i < size; i++) {
        cout << "Enter The Data ";
        cout << endl;
        cout << "---------------------------";
        cout << endl;
        cin >> data;
        temp = obj.create_node(data, temp);
    }
    cout << "Your Data";
    cout << endl;
    cout << "---------------------------";
    cout << endl;
    obj.display();
    cout << "Enter The data";
    cout << endl;
    cout << "---------------------------";
    cout << endl;
    cin >> data;
    obj.insert_pos(data);
    cout << "Your Data";
    cout << endl;
    cout << "---------------------------";
    cout << endl;
    obj.display();
    system("pause");
    return 0;
}
```

SCREEN SHOTS

QUESTION 3

```
#include<iostream>
using namespace std;
struct Node {
	int data;
	Node* Next;
};
Node* Insert(int x, Node *head) {
	Node* temp = new Node;
	temp->data = x;
	temp->Next = NULL;
	if (head != NULL) {
		temp->Next = head;
	}
	head = temp;
	return head;
}
void print(Node* head) {
	Node* temp = head;
	cout << "Your List is ";
	cout << endl;
	while (temp != NULL) {
		cout << temp->data << " ";
		cout << endl;
		temp = temp->Next;
	}
};
int main() {
	Node* head;
	head = NULL;
	int n = 0;
	cin >> n;
	int x = 0;
	for (int i = 0; i < n; i++) {
		cin >> x;
```

```
                head = Insert(x, head);

        }
        print(head);
        system("pause");
        return 0;

}
```

SCREEN SHOT