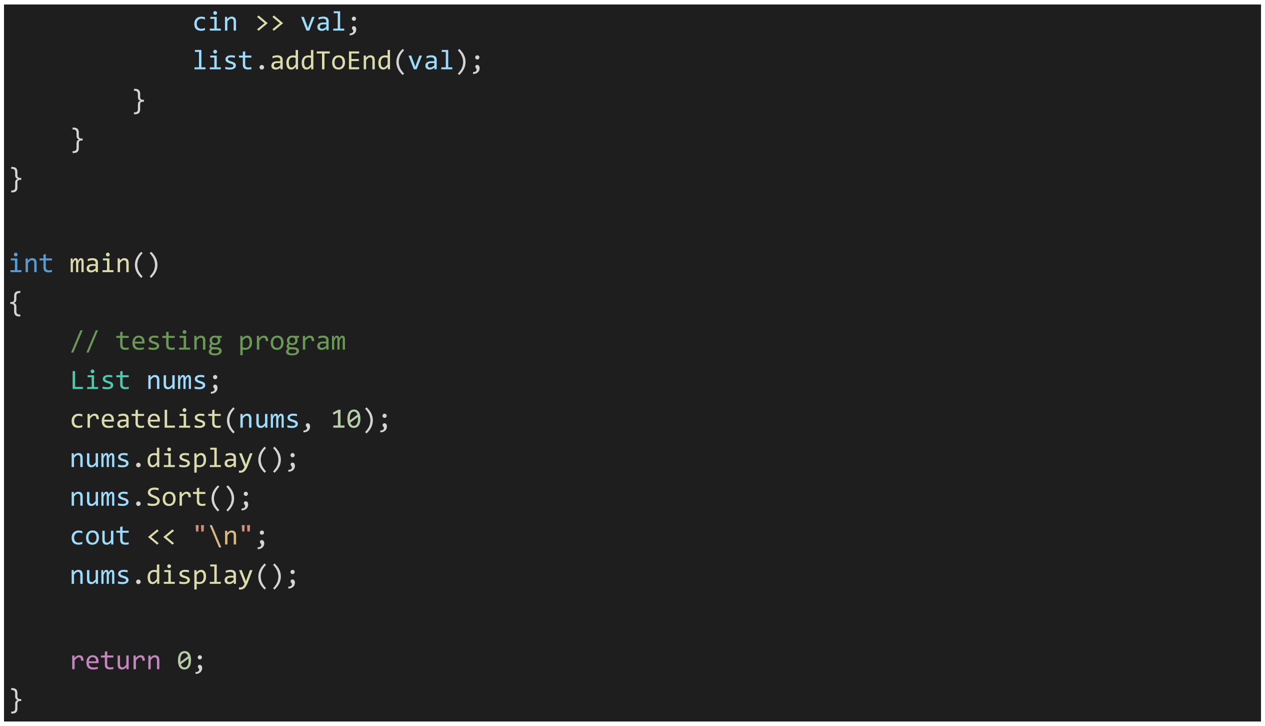
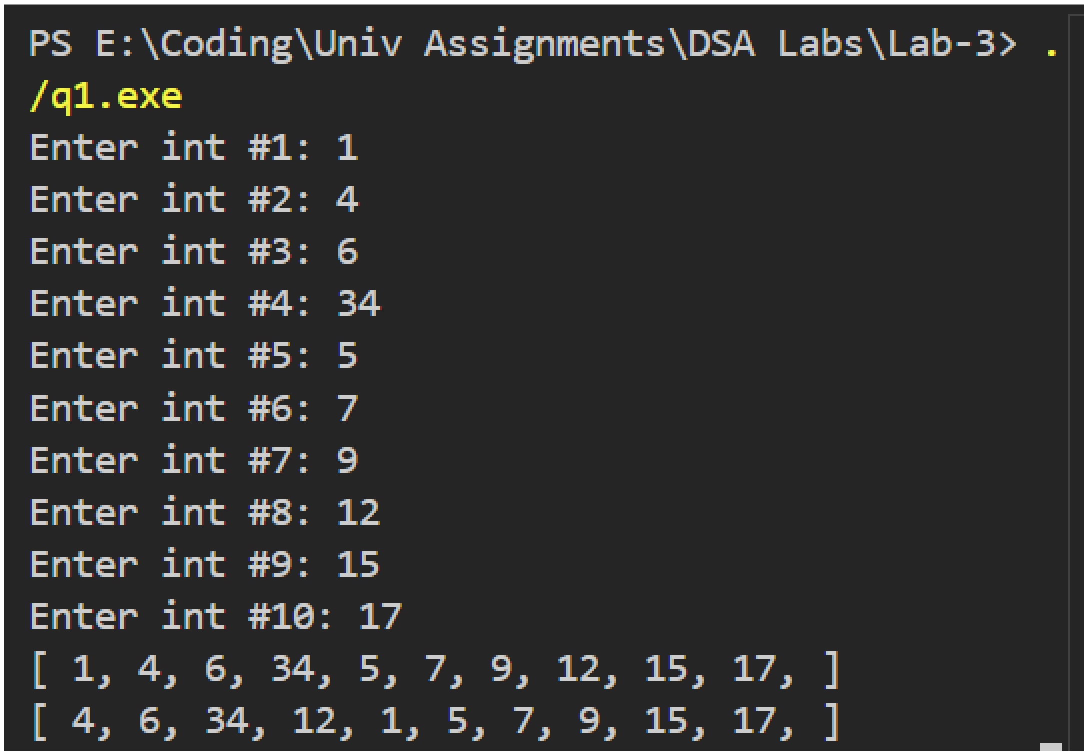
Q1:

```
#include "iostream"
using namespace std;

class Node
{
public:
    int data;
    Node *next;
    Node(int val) : data(val)
    {
        next = NULL; // default value of a new Node is pointing at NULL
    }
};

class List
{
    Node *head;
    Node *tail;

public:
    List()
    {
        head = NULL;
        tail = NULL; // when list is empty, these 2 should be pointing to null
    }
    bool isEmpty() { return head == NULL; }
    void addToStart(int val)
    {
        Node *newNode = new Node(val); // a new node created and initialized with
val
        if (isEmpty())
        {
            head = newNode;
            tail = newNode;
        }
        else
        {
            newNode->next = head; // now the new node points towards the head
            head = newNode;       // now the head is the new node which is the
starting one
        }
    }
    void addToEnd(int val)
    {
        Node *newNode = new Node(val); // a new node created and initialized with
val
        if (isEmpty())
        {
            head = newNode;
            tail = newNode;
        }
        else
        {
            tail->next = newNode;
            tail = newNode;
        }
    }
    void display()
    {
        Node *temp = new Node(0);
        temp = head;  // will contain the starting node
        cout << "[ "; // added to display in an array style
        while (temp != NULL)
        {
            cout << temp->data << ", ";
            temp = temp->next; // so now temp can go to the next node
        }
        cout << "]"; // added to display in an array style
        delete temp;
    }

    void Sort()
    {
        Node *OLoop;
        Node *prev;
        Node *curr;
        OLoop = head;
        prev = head;
        while (OLoop != NULL)
        {
            prev = NULL;
            curr = head;
```

```cpp
            while (curr != NULL && curr->next != NULL)
            {
                Node *temp = curr->next;

                if (curr->data % 2 != 0 && temp->data % 2 == 0)
                {
                    if (prev == NULL)
                    {
                        head = temp;
                    }
                    else
                    {
                        prev->next = temp;
                    }

                    curr->next = temp->next;
                    temp->next = curr;
                    prev = temp;
                }
                else
                {
                    prev = curr;
                    curr = curr->next;
                }
            }
            OLoop = OLoop->next;
        }
    }
};

void createList(List &list, int size) // function for list creation from sratch
{
    if (!list.isEmpty())
    {
        cout << "List already has content" << endl;
        return;
    }
    else
    {
        int val;
        for (int i = 0; i < size; i++)
        {
            cout << "Enter int #" << i + 1 << ": ";
```


```
            cin >> val;
            list.addToEnd(val);
        }
    }
}

int main()
{
    // testing program
    List nums;
    createList(nums, 10);
    nums.display();
    nums.Sort();
    cout << "\n";
    nums.display();

    return 0;
}
```


```
PS E:\Coding\Univ Assignments\DSA Labs\Lab-3> .
/q1.exe
Enter int #1: 1
Enter int #2: 4
Enter int #3: 6
Enter int #4: 34
Enter int #5: 5
Enter int #6: 7
Enter int #7: 9
Enter int #8: 12
Enter int #9: 15
Enter int #10: 17
[ 1, 4, 6, 34, 5, 7, 9, 12, 15, 17, ]
[ 4, 6, 34, 12, 1, 5, 7, 9, 15, 17, ]
```