Name: **Muhammad Anas Baig**

Enrollment No.: **01-134152-037**

Section: **BS(CS)-4A**

LAB-JOURNAL-6

Exercise 1:

Write a C++ program that implement Stack using Linked List.

Solution:

node.h File:

```
#pragma once
class node
{
public:
    int data;
    node *next;
public:
    node(void);
};
```

node.cpp File:

```
#include "node.h"

node::node(void)
{
}
```

stack.h File:

```
#pragma once
#include "stack.h"
#include "node.h"
#include <iostream>
using namespace std;

class stack
{
public:
    node *top;
public:
    stack(void);
    bool isEmpty();
    void push(int);
    int pop();
    int _top();
    void display();
};
```

stack.cpp File:

```
#include "stack.h"
#include "node.h"
```

```
3.  #include <iostream>
4.  using namespace std;
5.
6.  stack::stack(void)
7.  {
8.      top = new node;
9.      top = '\0';
10. }
11.
12. bool stack::isEmpty()
13. {
14.     if(top == '\0')
15.     {
16.         return true;
17.     }
18.     else
19.     {
20.         return false;
21.     }
22. }
23.
24. void stack::push(int value)
25. {
26.     node *ptr = new node;
27.     ptr->data = 0;
28.     ptr->next = '\0';
29.
30.     ptr->data = value;
31.
32.     ptr->next = top;
33.     top = ptr;
34. }
35.
36. int stack::pop()
37. {
38.     if(!isEmpty())
39.     {
40.         node *temp;
41.         temp = top;
42.         int value;
43.         value = top->data;
44.
45.         top = top->next;
46.
47.         delete temp;
48.         return (value);
49.     }
50.     else
51.     {
52.         cout<<"SORRY!!! Stack is Empty."<<endl;
53.         return (-1);
54.     }
55. }
56.
57. int stack::_top()
58. {
59.     if(!isEmpty())
60.     {
61.         return top->data;
62.     }
63.     else
64.     {
65.         cout<<"SORRY!!! Stack is Empty."<<endl;
66.         return (-1);
67.     }
68. }
69.
70. void stack::display()
71. {
72.     if(!isEmpty())
```

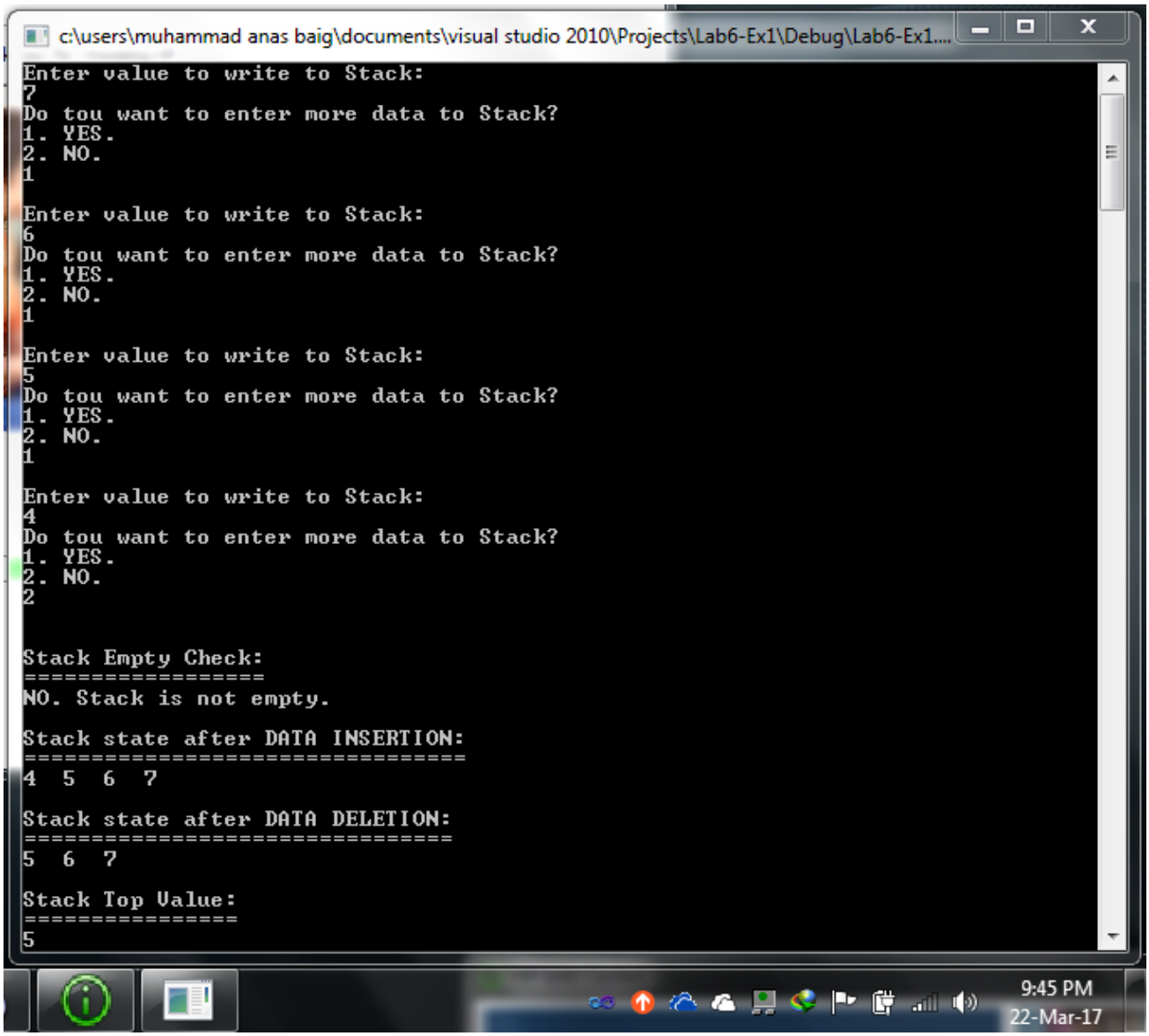
```
    {
        node *temp;
        temp = top;

        while(temp != '\0')
        {
            cout<<temp->data<<"  ";
            temp = temp->next;
        }
    }
    else
    {
        cout<<"SORRY!!! Stack is Empty."<<endl;
    }
}
```

main.cpp File:

```
#include "stack.h"
#include "node.h"
#include "conio.h"
#include <iostream>
using namespace std;

void main()
{
    stack s;
    int value, choice = 1;

    while(choice == 1)
    {
        cout<<"Enter value to write to Stack:"<<endl;
        cin>>value;
        s.push(value);
        cout<<"Do tou want to enter more data to Stack?"<<endl;
        cout<<"1. YES."<<endl;
        cout<<"2. NO."<<endl;
        cin>>choice;
        cout<<endl;
    }
    cout<<endl;

    cout<<"Stack Empty Check:"<<endl;
    cout<<"=================="<<endl;
    if(s.isEmpty())
    {
        cout<<"YES. Stack is empty."<<endl;
    }
    else
    {
        cout<<"NO. Stack is not empty."<<endl;
    }
    cout<<endl;

    cout<<"Stack state after DATA INSERTION:"<<endl;
    cout<<"================================"<<endl;
    s.display();
    cout<<endl;
    cout<<endl;

    s.pop();
    cout<<"Stack state after DATA DELETION:"<<endl;
    cout<<"================================"<<endl;
    s.display();
    cout<<endl;
    cout<<endl;

    cout<<"Stack Top Value:"<<endl;
    cout<<"================"<<endl;
```

```
52.     cout<<s._top();
53.     getch();
54. }
```

Output:



Exercise 2:

Write a C++ program that stores **Student-ID**, **Name** and **Age** in Linked List.
Write also functions to **Search** and **Display** student record.

Solution:

student.h File:

```
1.  #include <string>
2.  using namespace std;
3.
4.  #pragma once
5.  class student
6.  {
7.  public:
8.      int id;
9.      string name;
```

```
    int age;
    student *next;
public:
    student(void);
};
```

student.cpp File:

```
#include "student.h"
#include <string>
#include <iostream>
using namespace std;

student::student(void)
{
}
```

studentList.h File:

```
#include "student.h"
#include <string>
using namespace std;

#pragma once
class studentList
{
public:
    student *head;
public:
    studentList(void);
    bool isEmpty();
    void addStudent(int, string, int);
    void findStudent(int);
    void display();
};
```

studentList.cpp File:

```
#include "studentList.h"
#include "student.h"
#include <string>
#include <iostream>
using namespace std;

studentList::studentList(void)
{
    //head = new student;
    head = '\0';
}

bool studentList::isEmpty()
{
    if(head == '\0')
    {
        return true;
    }
    else
    {
        return false;
    }
}

void studentList::addStudent(int id, string name, int age)
{
    student *ptr = new student;
    ptr->id = 0;
```

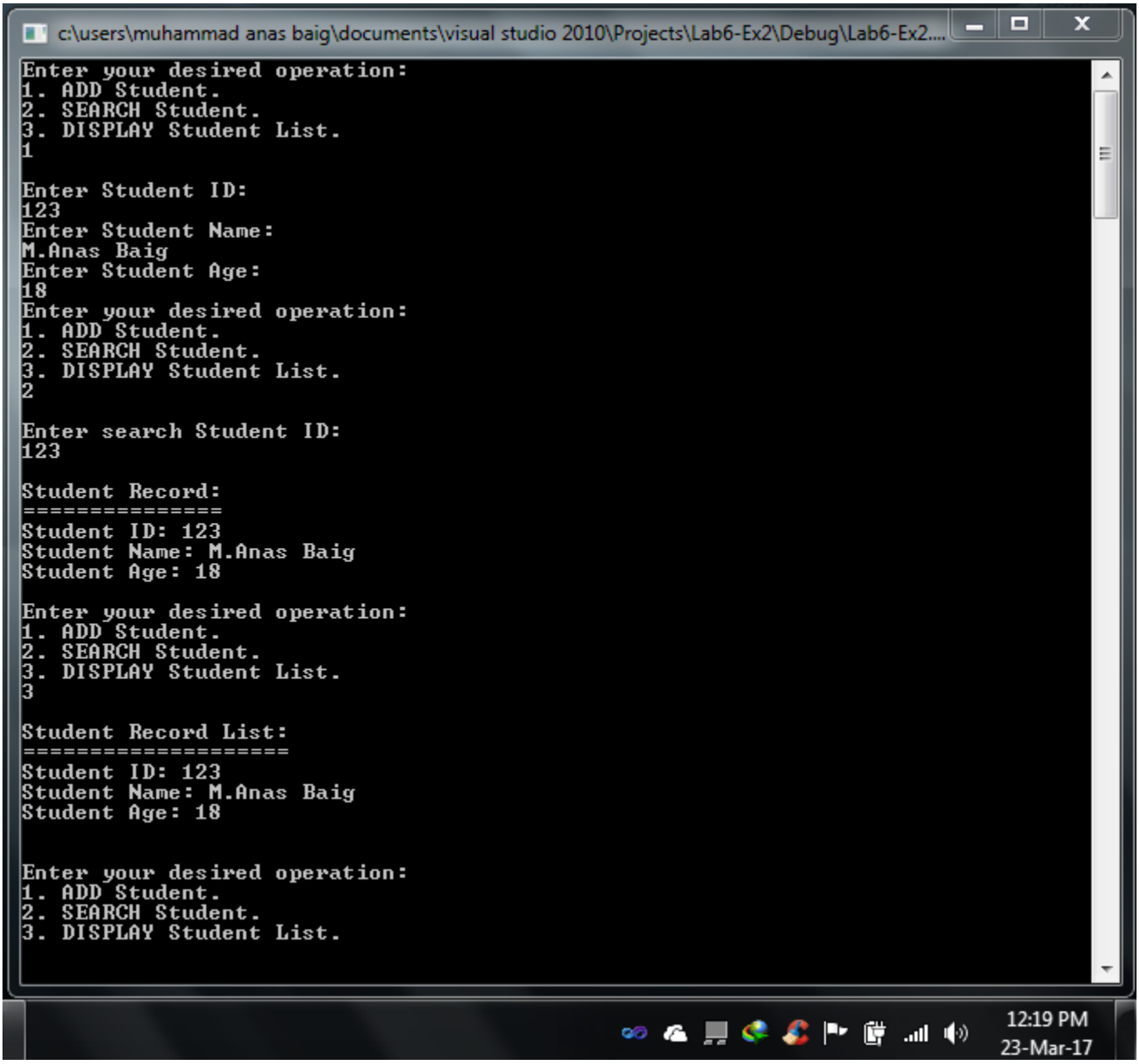
```
    ptr->name = '\0';
    ptr->age = 0;
    ptr->next = '\0';

    ptr->id = id;
    ptr->name = name;
    ptr->age = age;

    ptr->next = head;
    head = ptr;
}

void studentList::findStudent(int id)
{
    if(!isEmpty())
    {
        student *temp;
        temp = head;

        while(temp->id != id && temp!= '\0')
        {
            temp = temp->next;
        }

        cout<<"Student Record:"<<endl;
        cout<<"================"<<endl;
        cout<<"Student ID: "<<temp->id<<endl;
        cout<<"Student Name: "<<temp->name<<endl;
        cout<<"Student Age: "<<temp->age<<endl;
    }
    else
    {
        cout<<"SORRY!!! Student List is Empty"<<endl;
    }
}

void studentList::display()
{
    if(!isEmpty())
    {
        student *temp;
        temp = head;

        cout<<"Student Record List:"<<endl;
        cout<<"====================="<<endl;
        while(temp != '\0')
        {
            cout<<"Student ID: "<<temp->id<<endl;
            cout<<"Student Name: "<<temp->name<<endl;
            cout<<"Student Age: "<<temp->age<<endl;
            cout<<endl;
            temp = temp->next;
        }
    }
    else
    {
        cout<<"SORRY!!! Student List is Empty"<<endl;
    }
}
```

main.cpp File:

```
#include "studentList.h"
#include <string>
#include "student.h"
#include "conio.h"
#include <iostream>
using namespace std;

```

```
8.  void main()
9.  {
10.     studentList l;
11.     int choice;
12.     int id;
13.     string name;
14.     int age;
15.
16.     do
17.     {
18.         cout<<"Enter your desired operation:"<<endl;
19.         cout<<"1. ADD Student."<<endl;
20.         cout<<"2. SEARCH Student."<<endl;
21.         cout<<"3. DISPLAY Student List."<<endl;
22.         cin>>choice;
23.         cout<<endl;
24.         if(choice == 1)
25.         {
26.             cout<<"Enter Student ID:"<<endl;
27.             cin>>id;
28.             cout<<"Enter Student Name:"<<endl;
29.             cin.ignore(); //getline is having issue in while loop so that this statement is used
30.             getline(cin, name);
31.             cout<<"Enter Student Age:"<<endl;
32.             cin>>age;
33.             l.addStudent(id, name, age);
34.         }
35.         else if(choice == 2)
36.         {
37.             cout<<"Enter search Student ID:"<<endl;
38.             cin>>id;
39.             cout<<endl;
40.             l.findStudent(id);
41.             cout<<endl;
42.         }
43.         else
44.         {
45.             l.display();
46.             cout<<endl;
47.         }
48.     }
49.     while(choice == 1 || choice == 2 || choice == 3); //1 for ADD STUDENT, 2 for SEARCH STUDENT, 3 for DISPLAY ST
    UDENT LIST
50.     getch();
51. }
```

Output:



Exercise 3:

Write a C++ program which have functions to **Display Even Number in a List** and a function that **Deletes First Half of List**. Implement using Linked List.

Solution:

node.h file:

```
1.  #pragma once
2.  class node
3.  {
4.  public:
5.      int data;
6.      node *next;
7.  public:
8.      node(void);
9.  };
```

node.cpp File:

```cpp
#include "node.h"

node::node(void)
{
}
```

list.h File:

```cpp
#include "node.h"
#pragma once
class list
{
public:
    node *head;
public:
    list(void);
    bool isEmpty();
    void add(int);
    void deleteFirstHalf();
    void displayEven();
    void displayList();
};
```

list.cpp File:

```cpp
#include "list.h"
#include "node.h"
#include <iostream>
using namespace std;

list::list(void)
{
    head = '\0';
}
bool list::isEmpty()
{
    if(head == '\0')
    {
        return true;
    }
    else
    {
        return false;
    }
}

void list::add(int value)
{
    node *ptr = new node;
    ptr->data = 0;
    ptr->next = '\0';

    ptr->data = value;

    ptr->next = head;
    head = ptr;
}

void list::deleteFirstHalf()
{
    if(!isEmpty())
    {
        node *temp1 = head;
        int count1 = 0;
```

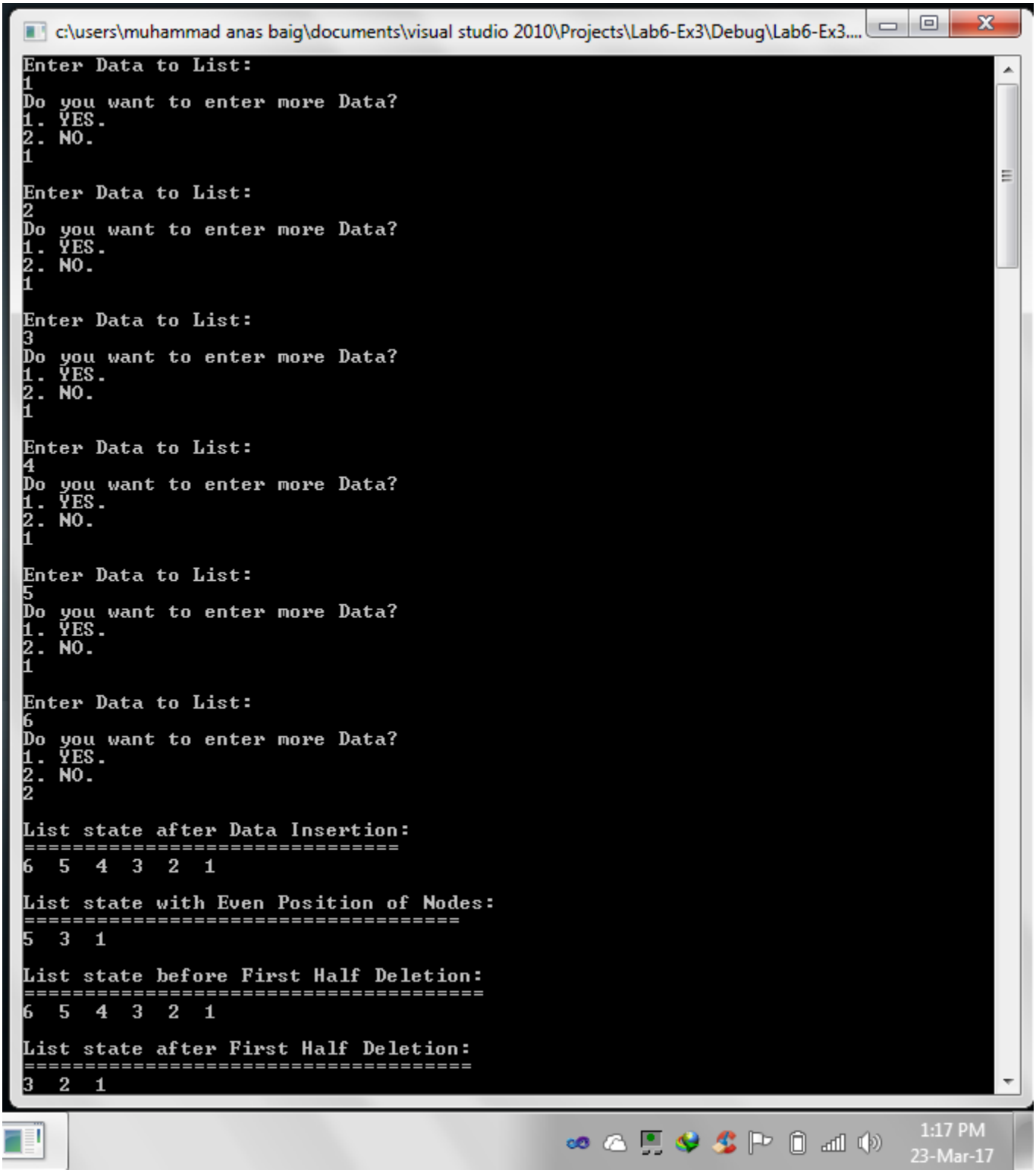
```
40.         int half;
41.
42.         while(temp1 != '\0')
43.         {
44.             count1++;
45.             temp1 = temp1->next;
46.         }
47.
48.         half = (count1/2);
49.
50.         node *temp2 = head;
51.         node *delTemp = '\0';
52.         int count2 = 0;
53.
54.         while(count2 != half)
55.         {
56.             count2++;
57.             delTemp = temp2;
58.             temp2 = temp2->next;
59.             delete delTemp;
60.         }
61.         head = temp2;
62.     }
63.     else
64.     {
65.         cout<<"SORRY!!! List is Empty."<<endl;
66.     }
67. }
68.
69. void list::displayEven()
70. {
71.     if(!isEmpty())
72.     {
73.         node *temp;
74.         temp = head;
75.         int count = 0;
76.
77.         while(temp != '\0')
78.         {
79.             count++;
80.             if( (count%2) == 0 )
81.             {
82.                 cout<<temp->data<<"  ";
83.             }
84.             temp = temp->next;
85.         }
86.     }
87.     else
88.     {
89.         cout<<"SORRY!!! List is Empty."<<endl;
90.     }
91. }
92.
93. void list::displayList()
94. {
95.     if(!isEmpty())
96.     {
97.         node *temp;
98.         temp = head;
99.
100.                while(temp != '\0')
101.                {
102.                    cout<<temp->data<<"  ";
103.                    temp = temp->next;
104.                }
105.            }
106.            else
107.            {
108.                cout<<"SORRY!!! List is Empty."<<endl;
109.            }
```

```
110.        }
```

main.cpp File:

```
1.  #include "list.h"
2.  #include "node.h"
3.  #include "conio.h"
4.  #include <iostream>
5.  using namespace std;
6.
7.  void main()
8.  {
9.      list l;
10.     int value;
11.     int choice;
12.
13.     do
14.     {
15.         cout<<"Enter Data to List:"<<endl;
16.         cin>>value;
17.         l.add(value);
18.         cout<<"Do you want to enter more Data?"<<endl;
19.         cout<<"1. YES."<<endl;
20.         cout<<"2. NO."<<endl;
21.         cin>>choice;
22.         cout<<endl;
23.     }
24.     while(choice == 1);
25.
26.     cout<<"List state after Data Insertion:"<<endl;
27.     cout<<"================================"<<endl;
28.     l.displayList();
29.     cout<<endl;
30.     cout<<endl;
31.
32.     cout<<"List state with Even Position of Nodes:"<<endl;
33.     cout<<"======================================"<<endl;
34.     l.displayEven();
35.     cout<<endl;
36.     cout<<endl;
37.
38.     cout<<"List state before First Half Deletion:"<<endl;
39.     cout<<"======================================="<<endl;
40.     l.displayList();
41.     l.deleteFirstHalf();
42.     cout<<endl;
43.     cout<<endl;
44.
45.     cout<<"List state after First Half Deletion:"<<endl;
46.     cout<<"======================================="<<endl;
47.     l.displayList();
48.     cout<<endl;
49.     cout<<endl;
50.
51.     getch();
52. }
```

Output:


c:\users\muhammad anas baig\documents\visual studio 2010\Projects\Lab6-Ex3\Debug\Lab6-Ex3....
Enter Data to List:
1
Do you want to enter more Data?
1. YES.
2. NO.
1
Enter Data to List:
2
Do you want to enter more Data?
1. YES.
2. NO.
1
Enter Data to List:
3
Do you want to enter more Data?
1. YES.
2. NO.
1
Enter Data to List:
4
Do you want to enter more Data?
1. YES.
2. NO.
1
Enter Data to List:
5
Do you want to enter more Data?
1. YES.
2. NO.
1
Enter Data to List:
6
Do you want to enter more Data?
1. YES.
2. NO.
2
List state after Data Insertion:
6 5 4 3 2 1
List state with Even Position of Nodes:
5 3 1
List state before First Half Deletion:
6 5 4 3 2 1
List state after First Half Deletion:
3 2 1
1:17 PM
23-Mar-17