Arab Academy for Science, Technology & Maritime Transport

College of Computing and Information Technology

Bonus Assignment Data Structure

Submitted by:

Nada Abd El Aziz Abd El Ghany Amna

ID: 231014025

Submitted to:

Dr. Ahmed Said

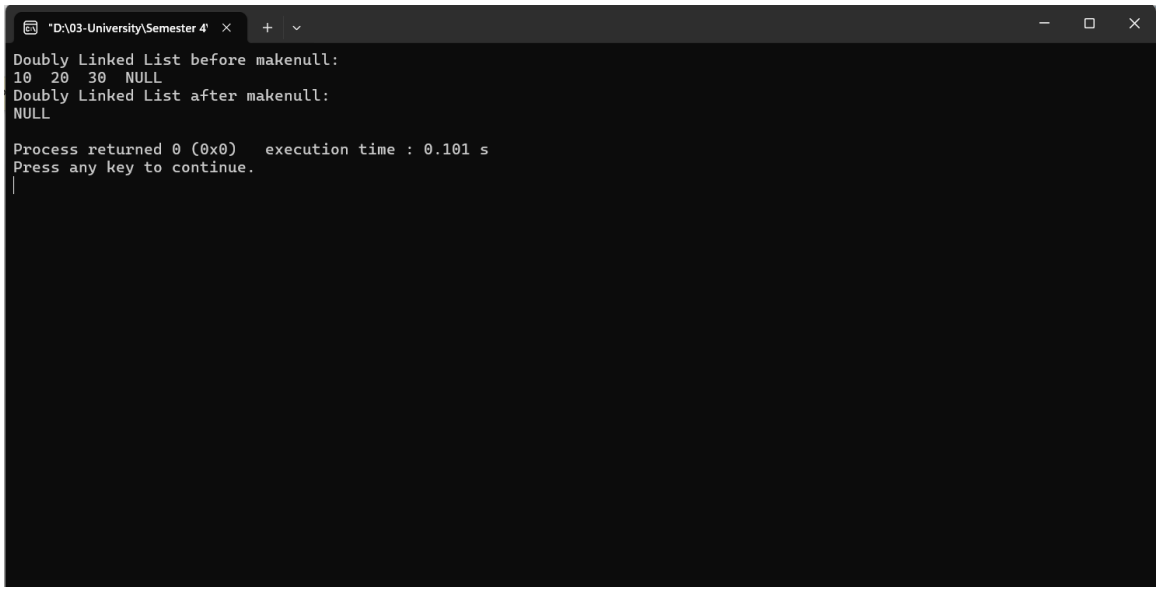
Question 1:

```
class Node {
public:
    int data;
    Node* next;
    Node* prev;
Node(int value) : data(value), next(nullptr), prev(nullptr){}
};
class DoublyList {
private:
    Node* head;
    Node* tail;
    int counter;
    public:
    DoublyList() : head(nullptr), tail(nullptr), counter(0)  {}
    void insert(int value) {
        Node* newNode = new Node(value);
        if (head == nullptr) {
            head = tail = newNode;
       }
 Else{
            tail->next = newNode;
            newNode->prev = tail;
            tail = newNode;
```

```
        }
        counter;++
    }
    void display(){
        Node* temp = head;
        while (temp != nullptr)}
            cout << temp->data;" <-> " >>
            temp = temp->next;
}
        cout << "NULL" << endl;
}
          void makenull (){
        while (head != nullptr)}
            Node* temp = head;
            head = head->next;
            delete temp;
        }
        tail = nullptr;
        counter = 0;
}
};
int main (){
    DoublyList list;
    list.insert;(10)
```

```
    list.insert;(20)
    list.insert;(30)
    cout << "Doubly Linked List before makenull: " << endl;
    list.display;()
    list.makenull;()
    cout << "Doubly Linked List after makenull: " << endl;
    list.display;()
    return 0;
}
```

Run:



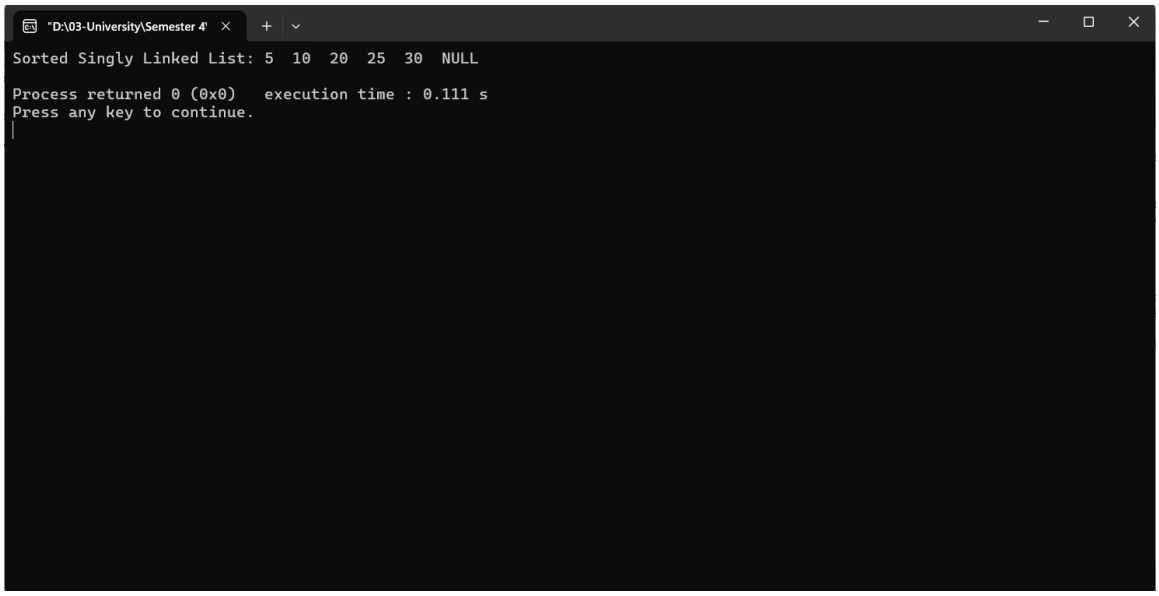
Question 2 “Singly”

```
class Node {
public:
    int data;
    Node* next;
    Node(int value) : data(value), next(nullptr) {}
};
class SinglyLinkedList {
private:
    Node* head;
public:
    SinglyLinkedList() : head(nullptr) {}
    void insertSorted(int value) {
        Node* newNode = new Node(value);

        if (!head || value < head->data) {
            newNode->next = head;
            head = newNode;
            return;
        }
        Node* current = head;
        while (current->next && current->next->data < value) {
            current = current->next;
        }
```

```
            newNode->next = current->next;
            current->next = newNode;
        }
    void display() {
            Node* temp = head;
            while (temp) {
                cout << temp->data << "    ";
                temp = temp->next;
            }
            cout << "NULL" << endl;
        }
        };
        int main() {
        SinglyLinkedList list;
        list.insertSorted(30);
        list.insertSorted(10);
        list.insertSorted(20);
        list.insertSorted(5);
        list.insertSorted(25);
        cout << "Sorted Singly Linked List: ";
        list.display();
        return 0;
}
```

Run:



Question 2 “Doubly”

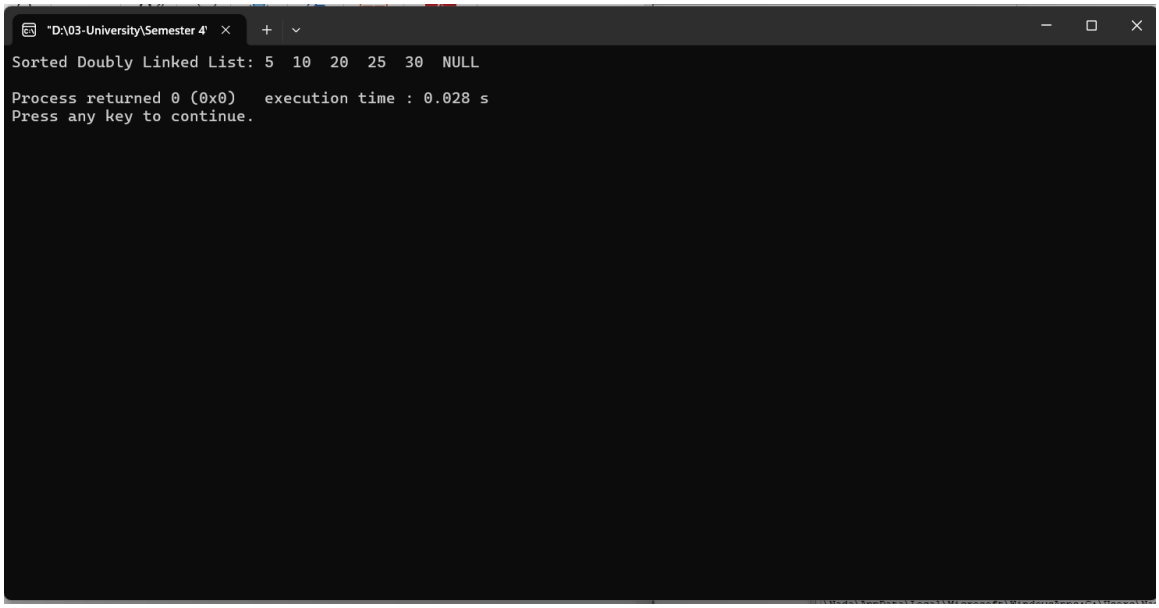
```
class Node {
public:
    int data;
    Node* next;
    Node* prev;
    Node(int value) : data(value), next(nullptr), prev(nullptr) {}
};
class DoublyLinkedList {
private:
    Node* head;
    Node* tail;
```

```
int counter;
public:
DoublyLinkedList() : head(nullptr), tail(nullptr), counter(0) {}
void insertSorted(int value) {
    Node* newNode = new Node(value);

    if (!head || value < head->data) {
        newNode->next = head;
        if (head) head->prev = newNode;
        head = newNode;
        if (!tail) tail = newNode;
        return;
    }
    Node* current = head;
    while (current->next && current->next->data < value) {
        current = current->next;
    }
    newNode->next = current->next;
    newNode->prev = current;
    current->next = newNode;
    if (newNode->next) {
        newNode->next->prev = newNode;
    } else {
        tail = newNode;
```

```
        }
    }
    void display() {
        Node* temp = head;
        while (temp) {
            cout << temp->data << "    ";
            temp = temp->next;
        }
        cout << "NULL" << endl;
    }
};
int main() {
    DoublyLinkedList list;
    list.insertSorted(30);
    list.insertSorted(10);
    list.insertSorted(20);
    list.insertSorted(5);
    list.insertSorted(25);
    cout << "Sorted Doubly Linked List: ";
    list.display();
    return 0;
}
```

Run:


"D:\03-University\Semester 4"
Sorted Doubly Linked List: 5 10 20 25 30 NULL
Process returned 0 (0x0) execution time : 0.028 s
Press any key to continue.