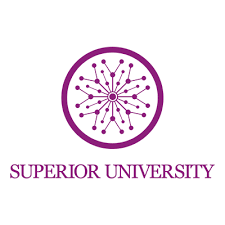
**TASK NO 8**

****

**Name:  
M. Zuhaib Anwar**

**Class :**

**SE – 3A**

**Roll No:**

**SU92-BSSEM-S24-005**

#include <iostream>

using namespace std;

class SNode {

public:

int value;

SNode\* next;

SNode(int val) : value(val), next(nullptr) {}

};

class SLL {

private:

SNode\* head;

public:

SLL() : head(nullptr) {}

void add\_to\_end(int val)

{

SNode\* newNode = new SNode(val);

if (!head)

{

head = newNode;

return;

}

SNode\* current = head;

while (current->next)

{

current = current->next;

}

current->next = newNode;

}

void merge(SLL& list)

{

if (!head)

{

head = list.head;

return;

}

SNode\* current = head;

while (current->next)

{

current = current->next;

}

current->next = list.head;

}

void display()

{

if (!head)

{

cout << "The list is empty." << endl;

return;

}

SNode\* current = head;

while (current)

{

cout << current->value << " ";

current = current->next;

}

cout << endl;

}

};

int main() {

SLL list1;

list1.add\_to\_end(1);

list1.add\_to\_end(2);

list1.add\_to\_end(3);

SLL list2;

list2.add\_to\_end(4);

list2.add\_to\_end(5);

list2.add\_to\_end(6);

cout << "List 1: ";

list1.display();

cout << "List 2: ";

list2.display();

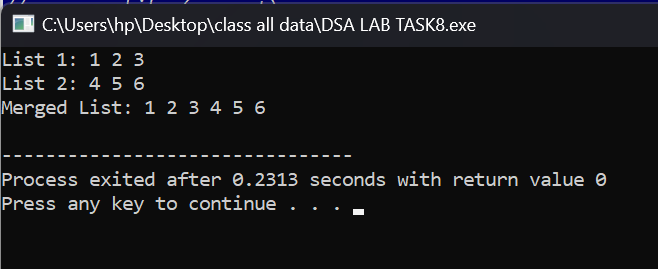
list1.merge(list2);

cout << "Merged List: ";

list1.display();

}

OUTPUT 1:



#include <iostream>

using namespace std;

class DNode {

public:

int value;

DNode\* next;

DNode\* prev;

DNode(int val) : value(val), next(nullptr), prev(nullptr) {}

};

class DLL {

private:

DNode\* head;

public:

DLL() : head(nullptr) {}

void add\_to\_end(int val)

{

DNode\* newNode = new DNode(val);

if (!head)

{

head = newNode;

return;

}

DNode\* current = head;

while (current->next)

{

current = current->next;

}

current->next = newNode;

newNode->prev = current;

}

void merge(DLL& list)

{

if (!head)

{

head = list.head;

return;

}

DNode\* current = head;

while (current->next)

{

current = current->next;

}

current->next = list.head;

if (list.head)

{

list.head->prev = current;

}

}

void display()

{

if (!head)

{

cout << "EMPTY" << endl;

return;

}

DNode\* current = head;

while (current)

{

cout << current->value << " ";

current = current->next;

}

cout << endl;

}

};

int main() {

DLL list1;

list1.add\_to\_end(1);

list1.add\_to\_end(2);

list1.add\_to\_end(3);

DLL list2;

list2.add\_to\_end(4);

list2.add\_to\_end(5);

list2.add\_to\_end(6);

cout << "List 1: ";

list1.display();

cout << "List 2: ";

list2.display();

list1.merge(list2);

cout << "Merged List: ";

list1.display();

}

OUTPUT 2:

