Name – Yashwant patidar

Scholar Number – 191112243

Section – CSE 2

Subject – Data Structures Lab

**Ques1.** Write a program to merge two doubly circular linked lists.

:-

#include <iostream>

using namespace std;

struct node {

int data;

node\* next;

node\* prev;

} \*head , \*head2 ;

void Create\_List (int data)

{

node\* new\_node = new node;

new\_node->data = data;

new\_node->next = NULL;

new\_node->prev = NULL;

if (head == NULL)

{

head = new\_node;

new\_node->next = head;

head->prev = new\_node;

}

node\* ptr = head;

while (ptr->next != head)

ptr = ptr->next;

ptr->next = new\_node;

new\_node->prev = ptr;

new\_node->next = head;

head->prev = new\_node;

}

void Create\_List2 (int data2)

{

node\* new\_node2 = new node;

new\_node2->data = data2;

new\_node2->next = NULL;

new\_node2->prev = NULL;

if (head2 == NULL)

{

head2 = new\_node2;

new\_node2->next = head2;

head2->prev = new\_node2;

}

node\* ptr2 = head2;

while (ptr2->next != head2)

ptr2 = ptr2->next;

ptr2->next = new\_node2;

new\_node2->prev = ptr2;

new\_node2->next = head2;

head2->prev = new\_node2;

}

void print\_List\_1 ()

{

if (head == NULL)

cout<<" NO NODES in the List 1 ---> List Is EMPTY "<<endl;

else

{

cout<<" "<<"Circular Doubly Linked List 1 : "<<endl

<<" "<<"------------------------------- "<<endl;

node\* temp = head;

do

{

cout<<" "<<temp->data<<" <-->";

temp = temp->next;

} while (temp != head);

cout<<" ( Pointing to 1st Node ) ";

}

}

void print\_List\_2 ()

{

if (head2 == NULL)

cout<<" NO NODES in the List 2 ---> List Is EMPTY "<<endl;

else

{

cout<<" "<<"Circular Doubly Linked List 2 : "<<endl

<<" "<<"------------------------------- "<<endl;

node\* temp2 = head2;

do

{

cout<<" "<<temp2->data<<" <-->";

temp2 = temp2->next;

} while (temp2 != head2);

cout<<" ( Pointing to 1st Node ) ";

}

}

void Concatenate ()

{

node\* p1 = head;

node\* p2 = head2;

while (p1->next != head)

p1 = p1->next;

p1->next = head2;

head2->prev = p1;

while (p2->next != head2)

p2 = p2->next;

p2->next = head;

head->prev = p2;

}

void print\_merged\_list ()

{

cout<<" CONCATENATED CIRCULAR DOUBLY LINKED LISTS : "<<endl

<<" ------------------------------------------- "<<endl;

node\* temp = head;

do {

cout<<" "<<temp->data<<" <-->";

temp = temp->next;

} while (temp != head);

cout<<" ( Pointing to 1st Node ) ";

}

int main ()

{

cout<<endl;

Create\_List(10);

Create\_List(55);

Create\_List(30);

Create\_List(78);

Create\_List(95);

cout<<" List 1 Created...!!"<<endl;

cout<<endl;

Create\_List2(25);

Create\_List2(67);

Create\_List2(80);

Create\_List2(100);

Create\_List2(45);

cout<<" List 2 Created...!!"<<endl;

cout<<endl<<endl;

print\_List\_1 ();

cout<<endl<<endl;

print\_List\_2 ();

cout<<endl<<endl;

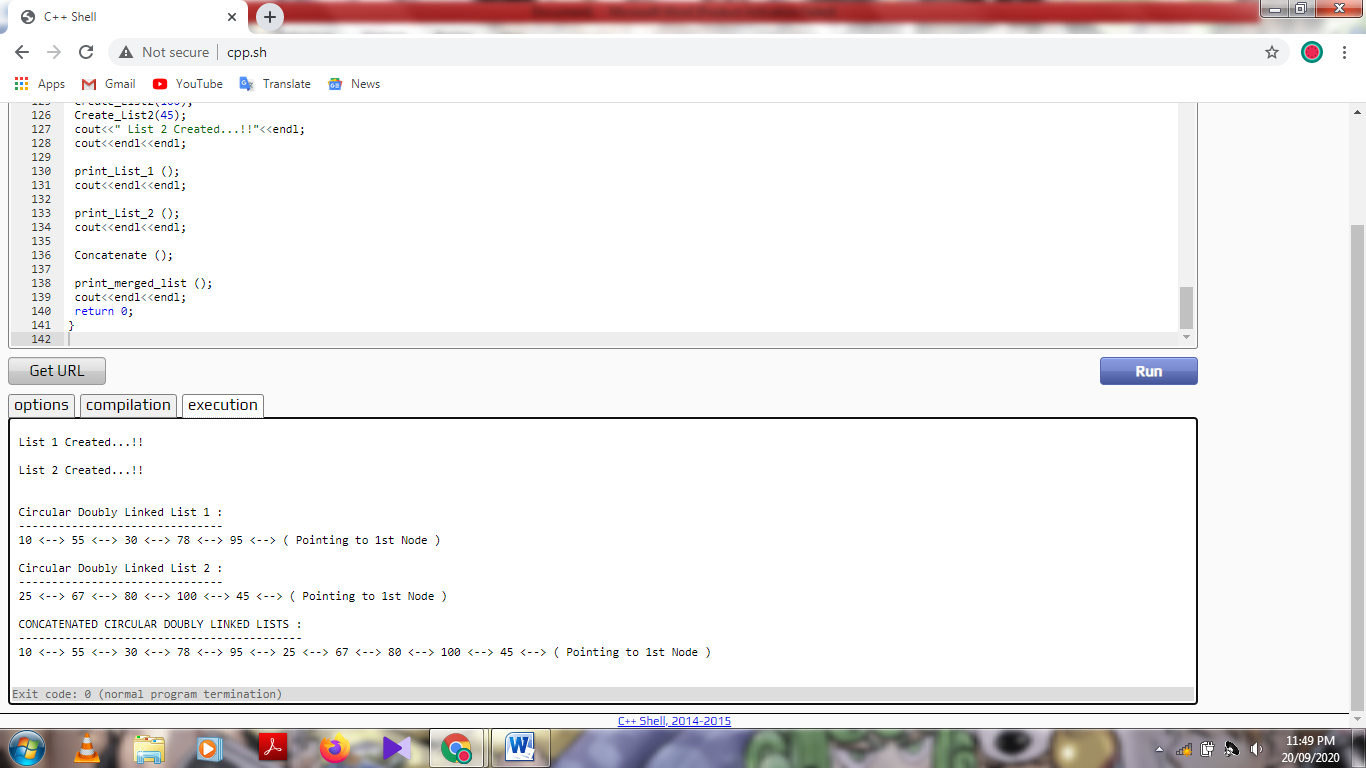
Concatenate ();

print\_merged\_list ();

cout<<endl<<endl;

return 0;

}



**Ques2.** Write a program to merge two sorted linked list into one linked list.

:-

#include <bits/stdc++.h>

using namespace std;

struct Node {

int data;

Node\* next;

} \*head1 , \*head2 , \*head3;

void create\_list\_1 (int data)

{

Node\* temp1 = new Node;

temp1->data = data;

temp1->next = NULL;

if (head1 == NULL)

{

head1 = temp1;

cout<<" Head Node Inserted !!"<<endl;

}

else

{

Node\* ptr1 = head1;

while (ptr1->next!= NULL)

{

ptr1 = ptr1->next;

}

ptr1->next = temp1;

cout<<" Node Inserted !!"<<endl;

}

}

void create\_list\_2 (int data2)

{

Node\* temp2 = new Node;

temp2->data = data2;

temp2->next = NULL;

if (head2 == NULL)

{

head2 = temp2;

cout<<" Head Node Inserted !!"<<endl;

}

else

{

Node\* ptr2 = head2;

while (ptr2->next!= NULL)

{

ptr2 = ptr2->next;

}

ptr2->next = temp2;

cout<<" Node Inserted !!"<<endl;

}

}

void create\_list\_3 (int data3)

{

Node\* temp3 = new Node;

temp3->data = data3;

temp3->next = NULL;

if (head3 == NULL)

{

head3 = temp3;

}

else

{

Node\* ptr3 = head3;

while (ptr3->next!= NULL)

{

ptr3 = ptr3->next;

}

ptr3->next = temp3;

}

}

void print\_1 ()

{

if (head1 == NULL)

cout<<" Empty List !!"<<endl;

else

{

Node \*ptr1 = head1;

while (ptr1 != NULL)

{

cout<<ptr1->data<<" --> ";

ptr1 = ptr1->next;

}

cout<<"NULL"<<endl;

}

}

void print\_2 ()

{

if (head2 == NULL)

cout<<" Empty List !!"<<endl;

else

{

Node \*ptr2 = head2;

while (ptr2 != NULL)

{

cout<<ptr2->data<<" --> ";

ptr2 = ptr2->next;

}

cout<<"NULL"<<endl;

}

}

void Merge ()

{

Node\* p1 = head1;

Node\* p2 = head2;

while (p1 != NULL && p2 != NULL)

{

if (p1->data < p2->data)

{

create\_list\_3 (p1->data);

p1 = p1->next;

}

else if (p1->data > p2->data)

{

create\_list\_3 (p2->data);

p2 = p2->next;

}

}

while (p1 != NULL)

{

create\_list\_3 (p1->data);

p1 = p1->next;

}

while (p2 != NULL)

{

create\_list\_3 (p2->data);

p2 = p2->next;

}

cout<<endl<<" Merged Lists : ";

if (head3 == NULL)

cout<<" Empty List !!"<<endl;

else

{

Node \*ptr3 = head3;

while (ptr3 != NULL)

{

cout<<ptr3->data<<" --> ";

ptr3 = ptr3->next;

}

cout<<"NULL"<<endl;

}

}

int main ()

{

int data , data2; char ch , ch2;

cout<<endl<<" FOR LIST 1 (Pls Enter Elements In Sorted Fashion) : "<<endl

<<" ---------------------------------------------------- "<<endl;

do {

cout<<" Enter data : ";

cin>>data;

create\_list\_1 (data);

cout<<endl<<" Want to Enter More : (y/n) ";

cin>>ch;

} while (ch == 'y');

cout<<endl<<endl;

cout<<" FOR LIST 2 (Pls Enter Elements In Sorted Fashion) : "<<endl

<<" ---------------------------------------------------- "<<endl;

do {

cout<<" Enter data : ";

cin>>data2;

create\_list\_2 (data2);

cout<<endl<<" Want to Enter More : (y/n) ";

cin>>ch2;

} while (ch2 == 'y');

cout<<endl<<endl;

cout<<" List 1 : ";

print\_1 ();

cout<<endl;

cout<<" List 2 : ";

print\_2 ();

Merge ();

cout<<endl;

return 0;

}

