Data structure & Algorithm

Lab # 09

Submitted to:

Sir Rehan

Submitted by:

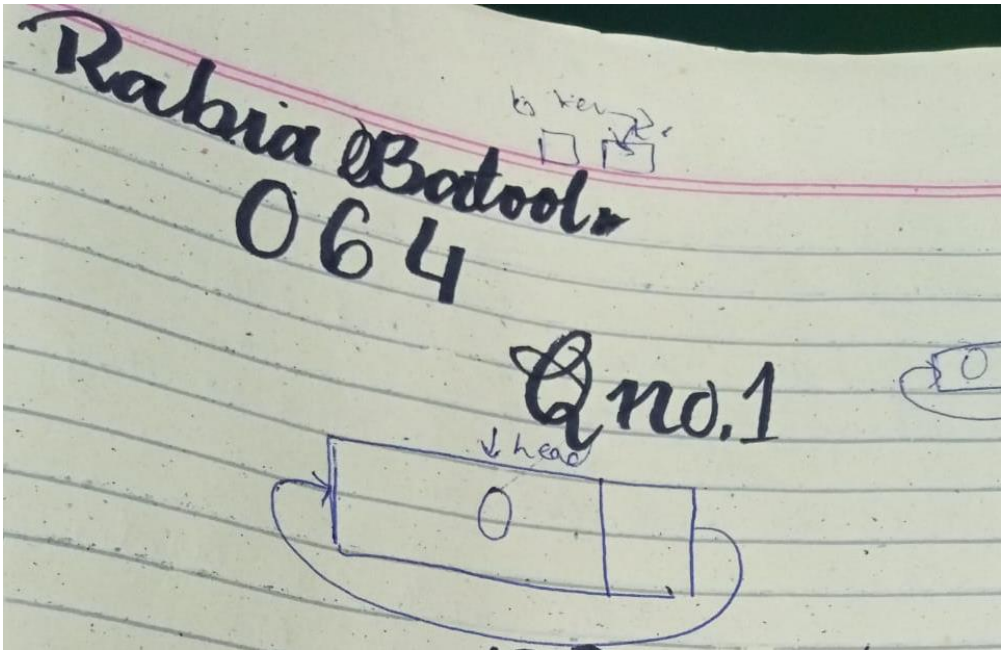
Rabia batool

2022-BSE-064

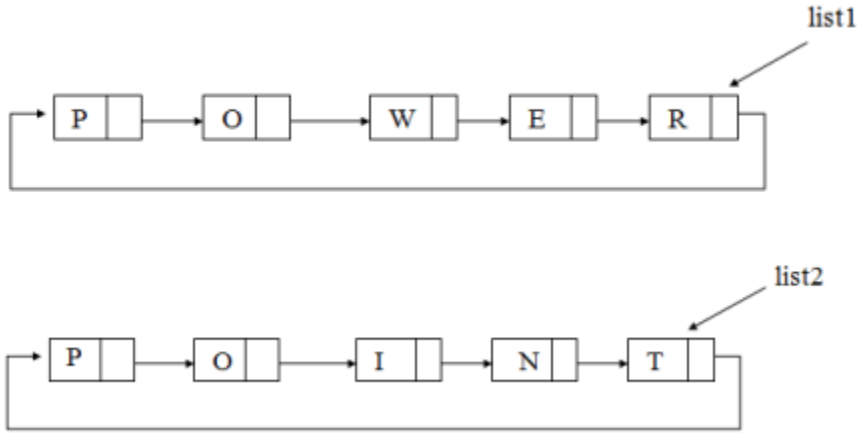
Task 1 :

Give answers to the following.

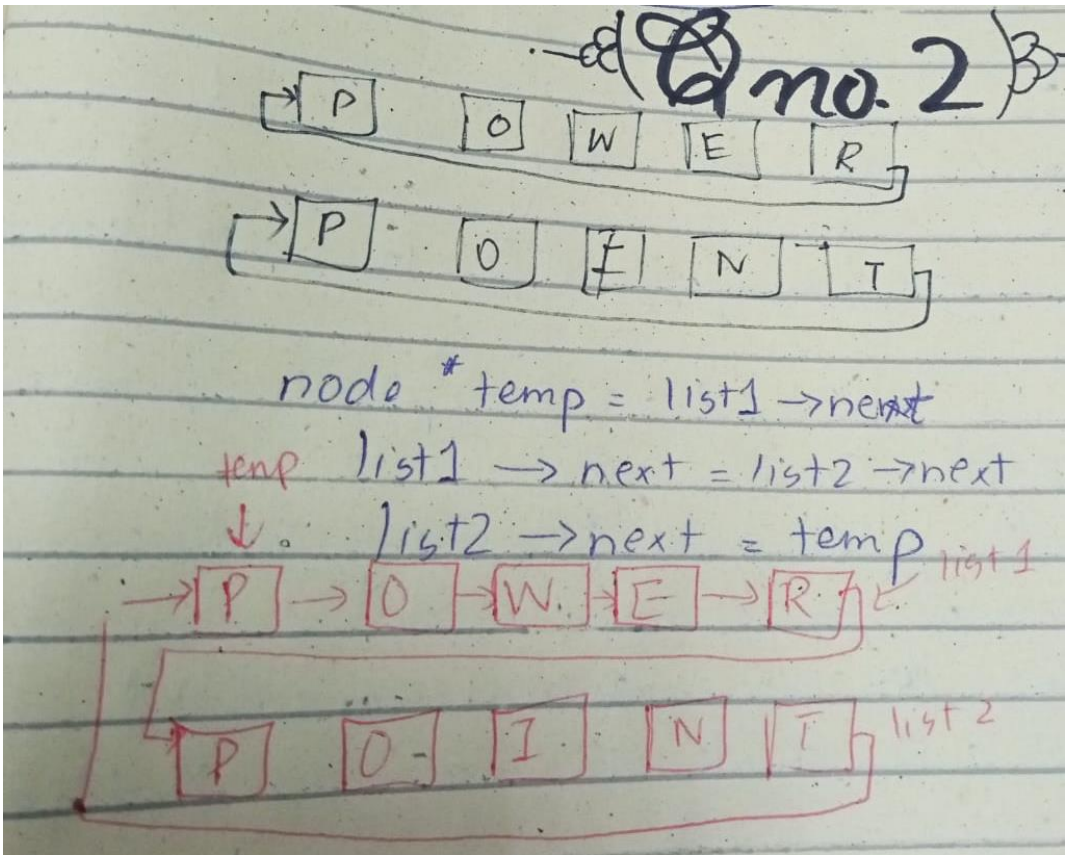
1.Draw a circular linked list of integers with a single node having a value



2. Consider the following two circular linked lists with pointers on their last nodes.



Write C++ statements to merge the two lists into one single list with contents POWERPOINT.



Code task 1

Implement the (class) Circular Linked List to create a list of integers. You need to provide the implementation of the member functions as described in the following.

Code:

```
#include<iostream>
using namespace std;
class circularlist
{public:
struct node
{
int data;
node* next;
}*head;
circularlist()
{head=NULL;
}
bool emptyList()
{
if(head==NULL)
return true;
else
return false;
}
```

```
void insert_at_begin(int value)
{
node * ptr1 = new node;
node * temp = head;
ptr1->data = value;
ptr1->next = head;

if (head != NULL)
{
while (temp->next != head)
temp = temp->next;
temp->next = ptr1;
}
else
ptr1->next = ptr1;
head = ptr1;
}
void insert_end(int value)
{
node *temp,*ptr;
ptr=new node;
ptr->data=value;

if(head==NULL){
head=ptr;
ptr->next=head;
}
else{
temp=head;
while(temp->next!=head)
temp=temp->next;
ptr->next=head;
temp->next=ptr;
}
}

void insert_after(int pos,int data) {
node *temp,*t;
temp=head;
for (int i = 1;i < pos;i++)
{
temp = temp->next;
if (temp == NULL)
{
cout<<"There are less than ";
cout<<pos<<" elements."<<endl;
return;
}
}
t=new node;
t->data=data;
t->next=temp->next;
temp->next=t;
}

void traverse()
{
node *temp = head;
```

```
if (head!=NULL)
{
do{
cout<<temp->data<<" "<<endl;
temp = temp->next;
}
while (temp != head);
}
cout<<endl;
}

void del_begin()
{
node *temp,*temp1;
temp=head;
if(head==NULL){
cout<<"\nList has no nodes";
return;
}
if(head->next==head){
head=NULL;
delete temp;
return;
}
temp1=head;
while(temp1->next!=head)
temp1=temp1->next;
head=temp->next;
temp1->next=head;
delete temp;
}
void del_end()
{
node *temp,*t;
temp=head;
if(temp==NULL){
cout<<"Empty linked list ";
return;
}
if(head->next==head) {
delete temp;
head=NULL; return;
}
t=head->next;
while(t->next!=head){
t=t->next;
temp=temp->next;
}
temp->next=head;
delete t;
}
};
int main()
{
circularlist c1;
c1.insert_at_begin(1);
c1.insert_at_begin(2);
c1.insert_at_begin(3);
```

```
c1.traverse();
c1.insert_end(4);
c1.traverse();
c1.insert_after(1,5);
c1.traverse();
c1.del_begin();
c1.del_end();
c1.traverse();
system("pause");
return 0;
}
```

Output

```
/tmp/lugmTKyHuj.o
insertion at start
3
2
1

insertion at end
3
2
1
4

insert after
3
5
2
1
4

deletion at start
5
2
1
4

deletion at end
5
2
1
```