COMSATS University Islamabad, Lahore Campus

3/21/2022

LAB ASSIGNMENT

(DSA)

Submitted to: *Sir M.Mudassir*
Submitted by: *Hassan Abdullah*
Reg No.: *Fa20-BCS-012 B*

DOUBLY LINKED LIST

```
#include<bits/stdc++.h>
using namespace std;

class node{
public:
    int data;
    node* next;
    node* prev;

    node(int value){
        next=NULL;
        data=value;
        prev=NULL;
    }
};

void insert_at_start(node* &head,int value){

    node* newNode = new node(value);
    if(head==NULL){
        head=newNode;
    }else{
        newNode->next=head;
        head->prev=newNode;
        head=newNode;
    }
```

```
}

void insert_at_end(node* &head,int value){
    node* newNode = new node(value);
    if(head==NULL){
        head=newNode;
    }else{
    node* ptr = head;
    while(ptr->next != NULL){
        ptr = ptr->next;
    }
    ptr->next = newNode;
    newNode->prev = ptr;
    }
}

void insert_at_index(node* &head,int value,int loc){
    node* newNode = new node(value);
    if(head==NULL){
        head=newNode;
    }else if(loc == 1 ){
        newNode->next=head;
        head->prev=newNode;
        head=newNode;
    }
    else{
        node* ptr = head;
        int count = 0;

        while(ptr != NULL){
            count = count+1;
            if(count == loc){
                (ptr->prev)->next = newNode;
                newNode->prev = ptr->prev;
                newNode->next = ptr;
                ptr->prev = newNode;
                break;
            }
            ptr = ptr->next;
        }
    }
}

void print(node* &node){
    while (node != NULL)
    {
        cout<<" "<<node->data;
        node = node->next;
    }
}

int main(){
node *head = NULL;
insert_at_start(head,3);
```

```
insert_at_end(head,2);
insert_at_index(head,4,1);

print(head);

}
```

CIRCULAR DOUBLY LINKED LIST

```
#include<bits/stdc++.h>
using namespace std;

class node{
public:
    int data;
    node* next;
    node* prev;

    node(int value){
        next=NULL;
        data=value;
        prev=NULL;
    }
};

void insert_at_start(node* &head,node* &tail,int value){
    node* newNode = new node(value);
    if(head==NULL){
        head=newNode;
        tail = newNode;
    }else{
        newNode->next=head;
        head->prev=newNode;
        head=newNode;
        head->prev=tail;
        tail->next = head;
    }
}

void insert_at_end(node* &head,node* &tail,int value){
    node* newNode = new node(value);
    if(head==NULL){
        head=newNode;
        tail=newNode;
    }else{
        tail->next = newNode;
        tail = newNode;
        head->prev = tail;
        tail->next = head;
    }
}

void insert_at_index(node* &head,node* &tail,int value,int loc){
```

```
node* newNode = new node(value);
if(head==NULL){
    head=newNode;
    tail=newNode;
}else if(loc == 1 ){
    newNode->next=head;
    head->prev=newNode;
    head=newNode;
    head->prev=tail;
    tail->next = head;
}
else{
    node* ptr = head;
    int count = 0;

    while(ptr->next != head){
        count = count+1;
        if(count == loc){
            (ptr->prev)->next = newNode;
            newNode->prev = ptr->prev;
            newNode->next = ptr;
            ptr->prev = newNode;
            break;
        }
        ptr = ptr->next;
    }
}
```

```
}

void print(node* &head){
    node* ptr = head;
    while (ptr->next!= head)
    {
        cout<<" "<<ptr->data;
        ptr = ptr->next;
    }
    cout<<" "<<ptr->data<<endl;
}

int main(){
node *head = NULL;
node *tail = NULL;
insert_at_start(head,tail,3);
insert_at_end(head,tail,2);
insert_at_index(head,tail,5,1);

print(head);

}
```