

Dsa-Lab-Task : 3

Roll No: 24P-0706

Dept: BS-CS

Name: Aazan Noor Khuwaja

Section : 3D

Qns1:

```
#include<iostream>

using namespace std;

class node {

public:

int data;

node *next;

node(int val):data(val),next(NULL){}

};

class ll{

public:

node *head,*tail;

int val, Target;

ll():head(NULL),tail(NULL){

}

bool is_it_empty()

{

return (head==NULL);

}
```

```
void insert_at_end()
{
    cout <<"Enter the value :"<<endl;
    cin>>val;
    node *new_node=new node(val);
    if(is_it_empty()){
        head=new_node;
        tail=new_node;
        return;
    }
    tail->next=new_node;
    tail=new_node;
}
void display_values()
{
    if(is_it_empty())
    {
        cout <<"List is already empty first insert values."<<endl;
        return;
    }
    node *temp=head;
    while(temp!=NULL)
    {
        cout <<temp->data<<" ";
    }
}
```

```
temp=temp->next;

}

}

void insert_value_append()

{

cout <<"Enter Target value after whaich you want to input:"<<endl;

cin>>Target;

node *temp=head;

int count=0;

while(temp!=NULL)

{

if(Target==temp->data)

{

count++;

if(count==3)

{

cout<<"Target found 3 times\n enter the value for new node:"<<endl;

cin>>val;

node *new_node=new node(val);

new_node->next=temp->next;

temp->next=new_node;

if(temp==tail)

{

tail=new_node;
}
```

```
}

return;
}

}

temp=temp->next;

}

cout<<"Your target value is not appeard 3 times"<<endl;

}

};

int main()

{

ll list;

char c;

cout << "Insert values" << endl;

while (true) {

list.insert_at_end();

cout << "(Enter 'q' to quit | if want to continue then press any key except q): ";

cin >> c;

cin.ignore(1000,'\\n');

if (c == 'q' || c == 'Q') {

break;

}

}
```

```
list.insert_value_append();

list.display_values();

return 0;

}
```

Qns 2:

```
#include<iostream>

using namespace std;

class node {

public:

int data;

node *next;

node(int val):data(val),next(NULL){}

};

class ll{

public:

node *head,*tail;

int val, Target;

ll():head(NULL),tail(NULL){

}

bool is_it_empty()

{

return (head==NULL);
```

```
}

void insert_at_end()

{

    cout <<"Enter the value :"<<endl;

    cin>>val;

    node *new_node=new node(val);

    if(is_it_empty()){

        head=new_node;

        tail=new_node;

        return;

    }

    tail->next=new_node;

    tail=new_node;

}

void display_values()

{

    if(is_it_empty())

    {

        cout <<"List is already empty first insert values."<<endl;

        return;

    }

    node *temp=head;

    while(temp!=NULL)

    {
```

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

temp=temp->next;

}

}

void middle_node()

{

int count=0;

int middle = 0;

node *temp=head;

node *t2=head;

while(temp!=NULL)

{

count++;

temp=temp->next;

}

middle = (count/2) +1;

count = 1;

while (t2!=NULL){

if(count == middle) {

cout<<t2->data<<endl;

return;

}

count++;

t2 = t2->next;
```

```
}

}

};

int main()
{
    ll list;

    char c;

    cout << "Insert values" << endl;

    while (true) {

        list.insert_at_end();

        cout << "(Enter 'q' to quit | if want to continue then press any key except q): ";

        cin >> c;

        cin.ignore(1000, '\n');

        if (c == 'q' || c == 'Q') {

            break;
        }
    }

    list.display_values();

    list.middle_node();

    return 0;
}
```

Qns 3:

```
#include <iostream>

using namespace std;

class node {

public:

int data;

node *next;

node(int val):data(val),next(NULL) {}

};

class LL {

public:

node *head,*tail;

LL():head(NULL),tail(NULL) {}

bool is_it_empty() {

return (head==NULL);

}

void insert_at_end() {

int val;

cout << "Enter the value: ";

cin >> val;

node *new_node = new node(val);

if (is_it_empty()) {

head=tail=new_node;
```

```
    } else {

        tail->next=new_node;

        tail=new_node;

    }

}

void display_values() {

    if (is_it_empty()) {

        cout << "List is already empty. Insert values first." << endl;

        return;

    }

    node *temp=head;

    while (temp!=NULL) {

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

        temp=temp->next;

    }

    cout << endl;

}

void count_even_even_odd_odd() {

    int even_even = 0;

    int odd_odd = 0;

    int index = 0;

    node *temp = head;
```

```
while (temp!=NULL) {  
  
    if (index%2== 0 && temp->data%2==0) {  
  
        even_even++;  
  
    } else if (index%2 == 1 && temp->data%2 == 1) {  
  
        odd_odd++;  
  
    }  
  
    index++;  
  
    temp = temp->next;  
  
}  
  
cout << "Even numbers at even indices: "<< even_even << endl;  
  
cout << "Odd numbers at odd indices: "<< odd_odd << endl;  
  
}  
  
};  
  
int main() {  
  
    ll list;  
  
    char c;  
  
    cout << "Insert values:" << endl;  
  
    while (true) {  
  
        list.insert_at_end();  
  
        cout << "Enter 'q' to quit | press any other key to continue: ";  
  
        cin >> c;  
  
        cin.ignore(1000, '\n');  
  
        if (c == 'q' || c == 'Q') {  
  
            break;  
        }  
    }  
}
```

```
}

}

cout << "\nList : ";

list.display_values();

list.count_even_even_odd_odd();

return 0;

}
```