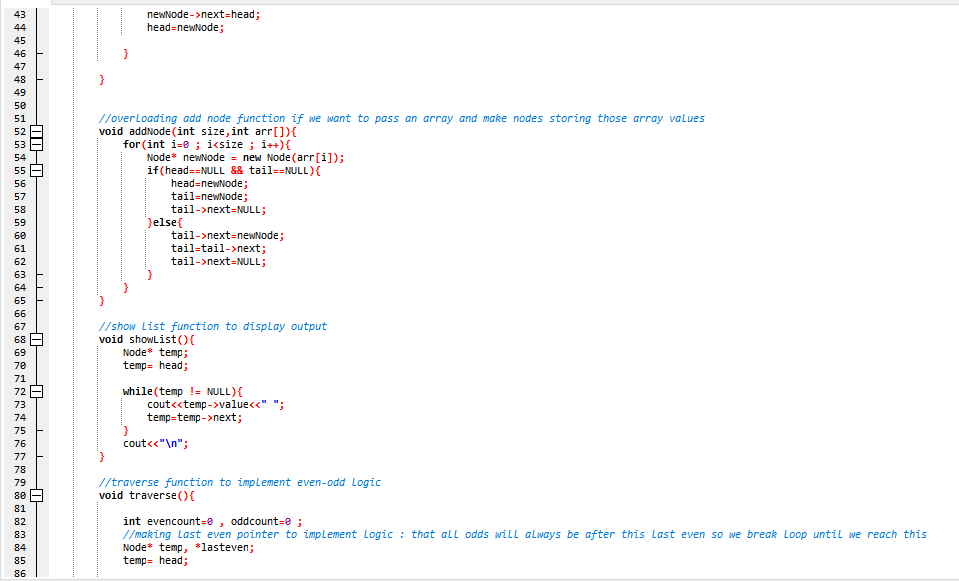
LAB3

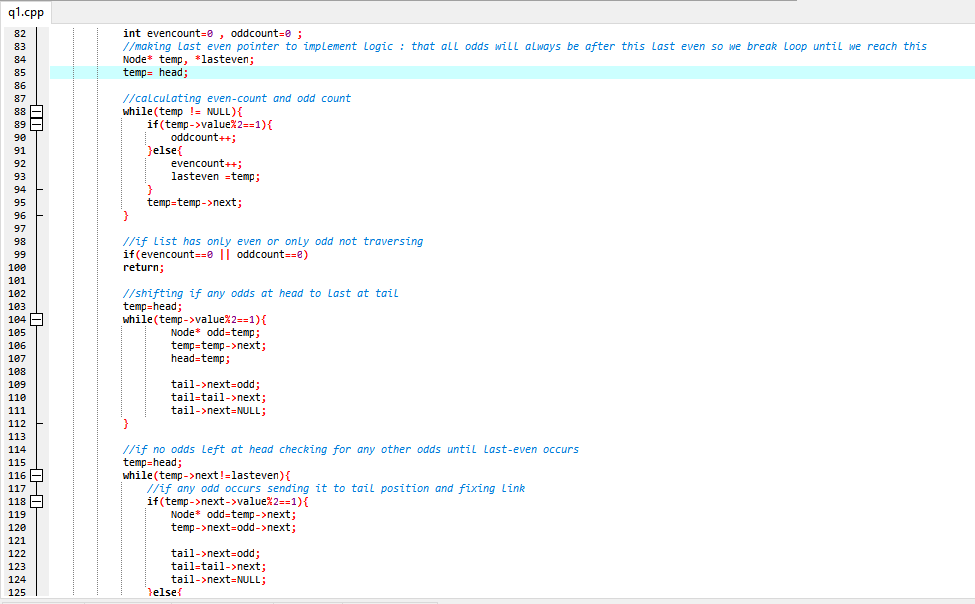
ASHAR ADNAN

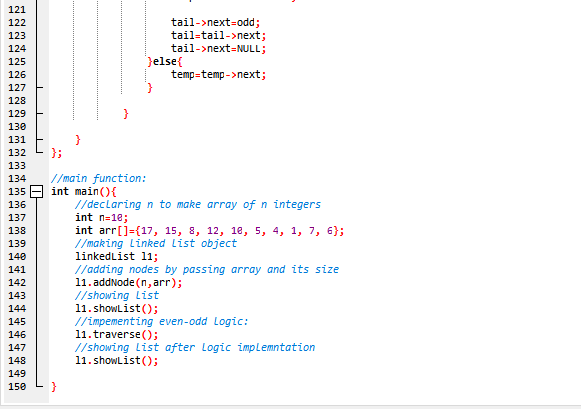
24K-0606

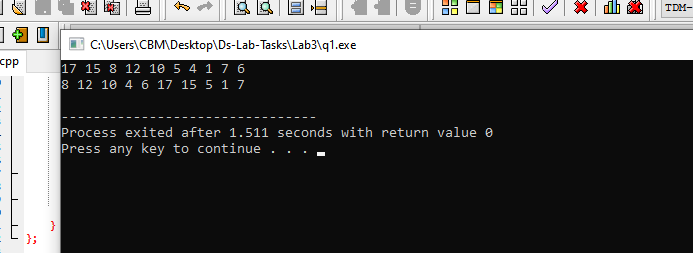
Q1:











**CODE(TASK1):**  
#include<iostream>

using namespace std;

//node class

class Node{

public:

int value;

Node\* next;

Node(){

}

Node(int val){

value=val;

}

};

//Linked list class

class linkedList{

private:

Node\* head;

Node\* tail;

public:

linkedList(){

head = NULL;

tail = NULL;

}

//add node function to add one node at a time

void addNode(int val){

Node\* newNode = new Node(val);

if(head==NULL){

head=newNode;

head->next=NULL;

tail=newNode;

}else{

newNode->next=head;

head=newNode;

}

}

//overloading add node function if we want to pass an array and make nodes storing those array values

void addNode(int size,int arr[]){

for(int i=0 ; i<size ; i++){

Node\* newNode = new Node(arr[i]);

if(head==NULL && tail==NULL){

head=newNode;

tail=newNode;

tail->next=NULL;

}else{

tail->next=newNode;

tail=tail->next;

tail->next=NULL;

}

}

}

//show list function to display output

void showList(){

Node\* temp;

temp= head;

while(temp != NULL){

cout<<temp->value<<" ";

temp=temp->next;

}

cout<<"\n";

}

//traverse function to implement even-odd logic

void traverse(){

int evencount=0 , oddcount=0 ;

//making last even pointer to implement logic : that all odds will always be after this last even so we break loop until we reach this

Node\* temp, \*lasteven;

temp= head;

//calculating even-count and odd count

while(temp != NULL){

if(temp->value%2==1){

oddcount++;

}else{

evencount++;

lasteven =temp;

}

temp=temp->next;

}

//if list has only even or only odd not traversing

if(evencount==0 || oddcount==0)

return;

//shifting if any odds at head to last at tail

temp=head;

while(temp->value%2==1){

Node\* odd=temp;

temp=temp->next;

head=temp;

tail->next=odd;

tail=tail->next;

tail->next=NULL;

}

//if no odds left at head checking for any other odds until last-even occurs

temp=head;

while(temp->next!=lasteven){

//if any odd occurs sending it to tail position and fixing link

if(temp->next->value%2==1){

Node\* odd=temp->next;

temp->next=odd->next;

tail->next=odd;

tail=tail->next;

tail->next=NULL;

}else{

temp=temp->next;

}

}

}

};

//main function:

int main(){

//declaring n to make array of n integers

int n=10;

int arr[]={17, 15, 8, 12, 10, 5, 4, 1, 7, 6};

//making linked list object

linkedList l1;

//adding nodes by passing array and its size

l1.addNode(n,arr);

//showing list

l1.showList();

//impementing even-odd logic:

l1.traverse();

//showing list after logic implemntation

l1.showList();

}