|  |
| --- |
| #include<iostream> |
|  | using namespace std; |
|  | class Node //contains data and pointer which points to the next element in the list// |
|  | { |
|  | public: |
|  | int data; |
|  | Node\*next; |
|  |  |
|  | Node() //constructor=>initialises the values// |
|  | { |
|  | data=0; |
|  | next=NULL; |
|  | } |
|  | }; |
|  | class LinkedList /\*contains a variable, a pointer head which points to the |
|  | first element and tail which points to NULL in the linked list.\*/ |
|  | { |
|  | public: |
|  | Node\*head; |
|  | Node\*tail; |
|  | int c; //counter=>used to count the number of elements// |
|  |  |
|  | LinkedList() //constructor// |
|  | { |
|  | head=NULL; |
|  | tail=NULL; |
|  | c=0; |
|  | } |
|  |  |
|  | //calling the member functions of the class "LinkedList"// |
|  | void insert(int data); |
|  | void insertAt(int pos,int data); |
|  | void del(); |
|  | void deleteAt(int pos); |
|  | int countItems(); |
|  | void display(); |
|  | }; |
|  |  |
|  | void LinkedList :: insert(int data) /\* |
|  | \*Executing the member functions outside of the |
|  | class "LinkedList". |
|  | \*This function is used to add a new node at the |
|  | end of the linked list. |
|  | \*/ |
|  | { |
|  | Node\*temp=new Node; //making a new node// |
|  | temp->data=data; //assigning data to the node// |
|  |  |
|  | if(head==NULL) //for an empty linked list// |
|  | { |
|  | head=temp; |
|  | tail=temp; |
|  | } |
|  |  |
|  | else //for a non-empty linked list// |
|  | { |
|  | tail->next=temp; |
|  | tail=temp; |
|  | } |
|  | tail->next=head;//the next of the last node points toward the 1st node |
|  | c++; //the size of the linked list is now increased by 1.// |
|  | } |
|  |  |
|  | void LinkedList :: insertAt(int pos, int data) /\*used to insert a new node at a |
|  | position other than the end\*/ |
|  | { |
|  | if(pos>c) |
|  | { |
|  | cout<<"Not that many elements in the list dude!\n"; |
|  | } |
|  |  |
|  | else if(pos==c) |
|  | insert(data); //calling the insert function in this function// |
|  |  |
|  | else if(pos==1) |
|  | { |
|  | Node\*temp=new Node; |
|  | temp->data=data; |
|  | temp->next=head; |
|  | head=temp; |
|  | c++; |
|  | } |
|  |  |
|  | else |
|  | { |
|  | Node\*temp=new Node; |
|  | temp->data=data; |
|  | Node\*p=head; //creating a new pointer and assigning it as the head pointer// |
|  |  |
|  | for(int i=1;i<pos;i++) //to reach the intended position// |
|  | { |
|  | p=p->next; |
|  | } |
|  |  |
|  | temp->next=p->next; |
|  | p->next=temp; |
|  | c++; |
|  | } |
|  | } |
|  |  |
|  | void LinkedList :: del() //used to delete the element at the end// |
|  | { |
|  | Node\*p=head; |
|  | Node\*q; /\*creating a new temporary pointer that after reaching the end |
|  | points to the last element so that we can delete that element\*/ |
|  |  |
|  | for(int i=1;i<c;i++) |
|  | { |
|  | p=p->next; |
|  | } |
|  |  |
|  | q=p->next; |
|  | p->next=NULL; |
|  | delete q; //deleting the new pointer// |
|  | c--; //the size of the linked list is now reduced by 1.// |
|  | } |
|  |  |
|  | void LinkedList :: deleteAt(int pos) //used to delete an element at any position// |
|  | { |
|  | if(pos>c) |
|  | cout<<"Not that many elements in the list dude!\n"; |
|  |  |
|  | else if(pos==c) |
|  | del(); //calling the delete function// |
|  |  |
|  | else if(pos==1) |
|  | { |
|  | Node\*p=head; |
|  | head=p->next; |
|  | delete p; |
|  | c--; |
|  | } |
|  |  |
|  | else |
|  | { |
|  | Node\*p=head; |
|  | Node\*q; |
|  |  |
|  | for(int i=1;i<pos;i++) |
|  | { |
|  | p=p->next; |
|  | } |
|  |  |
|  | q=p->next; |
|  | p->next=q->next; |
|  | delete q; |
|  | c--; |
|  | } |
|  | } |
|  |  |
|  | int LinkedList :: countItems() //used to count the size of the linked list// |
|  | { |
|  | return c; //returns the value of 'c' to int main// |
|  | } |
|  |  |
|  | void LinkedList :: display() //displays the linked list// |
|  | { |
|  | Node\*p=head; |
|  |  |
|  | while(p->next!=head) |
|  | { |
|  | cout<<p->data<<"->"; |
|  | p=p->next; |
|  | } |
|  | cout<<p->next<<"\n"; |
|  | } |
|  |  |
|  | int main() //executing the functions// |
|  | { |
|  | LinkedList l; |
|  |  |
|  | for(int i=0;i<10;i++) |
|  | { |
|  | l.insert(i+1); |
|  | } |
|  |  |
|  | l.insertAt(5,7); |
|  | l.deleteAt(8); |
|  | l.display(); |
|  | cout<<"The number of elements in the list are "<<l.countItems()<<"\n"; |
|  |  |
|  | return 0; |
|  | } |