1.  
include <iostream>

#include <math.h>

using namespace std;

struct node{

int data;

node\* prev, \*next;

};

node\* createnode(int num) {

node\* newnode = new node;

newnode->data = num;

newnode->prev = NULL;

newnode->next = NULL;

return newnode;

}

void nextnode(node\*\* head, int value) {

node\* newnode = createnode(value);

if (\*head == NULL) {

\*head = newnode;

} else {

node\* temp = \*head;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = newnode;

newnode->prev = temp;

}

}

void firslist(node\* head){

cout <<"NULL"<<" <--> ";

node\* temp = head;

while (temp != NULL)

{

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

temp = temp->next;

}

cout <<"NULL"<<endl;

}

void seclist(node\* head){

node\* temp = head;

int counter = 0 ;

int counterkk = 0 ;

cout <<"NULL"<<" <--> ";

while (temp != NULL)

{ counterkk++;

if (temp->data == 0)

counter++;

else

cout<< "(" << counterkk<< ", " << temp->data<< ")"<< " <--> ";

temp = temp->next;

}

cout <<"NULL"<<endl;

cout<<"null elements"<<": "<<counter<<endl;

}

void zero(node\* head, int i, int j) {

node\* temp = head;

int countzero=0;

int counter=0;

while (temp != NULL)

{ counter++;

if (temp->data == 0 && counter>=i && counter<=j)

countzero++;

temp = temp->next;

}

cout<<"zero"<<": "<<countzero;

}

int main(){

node\* head = NULL;

nextnode(&head, 0);

nextnode(&head, 2);

nextnode(&head, 0);

nextnode(&head, 4);

nextnode(&head, 0);

nextnode(&head, 6);

nextnode(&head, 0);

nextnode(&head, 8);

nextnode(&head, 0);

nextnode(&head, 10);

nextnode(&head, 0);

nextnode(&head, 0);

firslist(head);

seclist(head);

int i=3;

int j = 8;

zero(head, i, j);

return 0;

}

2.

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <math.h>

using namespace std;

struct nodeM {

int data;

int index;

nodeM\* next;

};

struct matrix{

nodeM\* arr[4];

};

matrix matrixintoList(int arr[4][4], int arr1[4][4]) // (int \*\*arr, int size1, int size2)

{

matrix list1;

for (int i = 0; i < 4; i++)

list1.arr[i] = NULL;

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 4; j++)

{

if (arr[i][j]!=0 or arr1[i][j]!=0)

{

nodeM \*tempNew = new nodeM;

tempNew->data = arr[i][j]+arr1[i][j];

tempNew->index = j;

tempNew->next = NULL;

if (!list1.arr[i])

list1.arr[i] = tempNew;

else

{

nodeM\* temp = list1.arr[i];

while (temp->next)

temp = temp->next;

temp->next = tempNew;

}

}

}

}

return list1;

}

void showList1(matrix list1){

nodeM\* temp;

for (int i = 0; i < 4; i++)

{

cout << "line: " << i << ": ";

// cout << "i = " << i << ": ";

temp = list1.arr[i];

while (temp)

{

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

temp = temp->next;

}

cout << endl;

}

}

int main(){

std::cout <<"array"<<endl;

int array[4][4];

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

array[i][j] = rand() % 17;

}

}

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

std::cout << array[i][j] << " ";

}

std::cout << std::endl;

}

std::cout <<"array1"<<endl;

int array1[4][4];

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

array1[i][j] = rand() % 17;

}

}

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

std::cout << array1[i][j] << " ";

}

std::cout << std::endl;

}

matrix list1;

list1 = matrixintoList(array, array1);

showList1(list1);

return 0;

}