#include<iostream>

using namespace std;

#define MAX 10

class listgraph;

class graphnode

{

public:

int vertex;

graphnode \*next;

public:

graphnode(int n=0)

{

vertex=n;

next=NULL;

}

friend class graph;

};

class listgraph

{

public:

graphnode \*list[MAX];

int n;

public:

listgraph(int nov)

{

n=nov;

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

list[j]=NULL;

}

void create();

void del(int d);

void put(int w);

void display();

void rem();

};

void listgraph::create()

{

int i, j;

graphnode \*temp, \*newnode;

char ans;

for(i=0; i<n;i++)

{

for(j=0;j<n;j++)

{

cout<<"\nIs there an edge between vertex"<<i+1<<"and"<<j+1;

cin>>ans;

if (ans=='y')

{

newnode=new graphnode(j+1);

if(list[i]==NULL)

list[i]=temp=newnode;

else

{

temp->next=newnode;

temp=newnode;

}

}

}

}

}

void listgraph::rem()

{

graphnode \*temp;

int ver,pos;

cout<<"Enter number of vertices:";

cin>>ver;

cout<<"Enter position of vertex that is to be deleted:";

cin>>pos;

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

{

temp=list[i];

if(i==pos)

{

break;

}

else

{

cout<< "V "<<i+1<<"->";

while(temp!=NULL)

{

cout<<temp->vertex <<"->";

temp=temp->next;

}

cout<<"NULL\n";

}

}

}

void listgraph::display()

{

graphnode \*temp;

int i;

for(i=0;i<n;i++)

{

cout << endl;

cout<< "V "<<i+1<<"->";

temp=list[i];

while(temp!=NULL)

{

cout<<temp->vertex <<"->";

temp=temp->next;

}

cout<<"NULL";

}

}

void listgraph::put(int w)

{

struct graphnode \*temp,\*q;

int n;

cout<<"Enter edge node to insert at this vertex:";

cin>>n;

graphnode \*gnode=new graphnode;

temp=list[w];

gnode->vertex=n;

gnode->next=NULL;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=gnode;

}

void listgraph::del(int d)

{

struct graphnode \*tmp,\*q,\*temp;

int n;

temp=list[d];

cout<<"Enter edge node to delete at this vertex:";

cin>>n;

if(temp->vertex == n)

{

tmp=temp;

temp=temp->next; /\* first element deleted \*/

free(tmp);

return;

}

q=temp;

while(q->next->next != NULL)

{

if(q->next->vertex==n) /\* element deleted in between \*/

{

tmp=q->next;

q->next=tmp->next;

free(tmp);

return;

}

q=q->next;

}

if(q->next->vertex==n) /\* last element deleted \*/

{

tmp=q->next;

free(tmp);

q->next=NULL;

}

}

int main()

{

int n,d,a,m,size,p,ch;

cout<<"\nEnter the number of vertices :";

cin>>n;

listgraph g(n);

cout<<"\n1.Create\n2.Insert edge\n3.Delete edge\n4.Delete node\n";

while(1)

{

cout<<"\nEnter your choice:\n";

cin>>ch;

if(ch==1)

{

g.create();

g.display();

}

else if(ch==2)

{

cout<<"\nEnter in which vertex you want to insert edge [index of vertices 0 to n-1]:";

cin>>a;

g.put(a);

g.display();

}

else if(ch==3)

{

cout<<"\nEnter whose vertix's edge you want to delete [index of vertices 0 to n-1]:";

cin>>d;

g.del(d);

g.display();

}

else if(ch==4)

{

g.rem();

}

else

{

cout<<"invalid choice";

}

}

}