**ASSIGNMENT NO # 03**

**------------------------------------------------------**

**NAME # NOMAN AMJAD**

**SECTION # BSCS 3(B)**

**REG NO # 21-ARID-654**

**SUBMITTED TO # DR.TARIQ ALI**

**------------------------------------------------------**

CODE :

#include<iostream>

#include<conio.h>

#include<string>

using namespace std;

class student;

student \*start=NULL;

class student

{

int rolNo;

string name;

float cgpa;

student \*next;

public:

void insertatstart()

{

system("CLS");

student \*T=new student();

cout<<"Enter a roll No : ";

cin>>T->rolNo;

cout<<"Enter a name :";

cin>>T->name;

cout<<"Enter a cgpa :";

cin>>T->cgpa;

T->next=start;

start=T;

}

void display()

{

system("CLS");

if(start==NULL)

{

cout<<"No Node exist!!!\n";

}

else

{

student \*T1=start;

cout<<"\n\n\t\t\t\t\t\t ~~~~~~~STUDENT RECORD~~~~~~~";

cout<<"\n\n\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\t\t\t\t\t"<<"Rol No "<<"\t | \t "<<" Name \t"<<" | "<<"\tCGPA |"<<endl;

while(T1!=NULL)

{

cout<<"\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout<<" \t\t\t\t\t"<<T1->rolNo<<" \t |\t "<<T1->name<<" \t |\t"<<T1->cgpa<<" \t |"<<endl;

T1=T1->next;

}

cout<<"\n\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

}

void deleteatstart()

{

system("CLS");

if(start==NULL)

{

cout<<"No Node to delete....\n";

}

else

{

student \*T2=start;

start=start->next;

delete T2;

cout<<"\nNode is deleted succussfullay.......\n";

}

}

void insertAtEnd()

{

system("CLS");

student \*t=new student();

cout<<"Enter a name :";

cin>>t->name;

cout<<"Enter a Roll Num :";

cin>>t->rolNo;

cout<<"Enter a CGPA :";

cin>>t->cgpa;

t->next=NULL;

if(start==NULL)

{

start=t;

}

else

{

student \*s=start;

while(s->next!=NULL)

{

s=s->next;

}

s->next=t;

}

}

void deleteatEnd()

{

system("CLS");

if(start==NULL)

{

cout<<"NO Node is exists...\n";

}

else if(start->next==NULL)

{

start=NULL;

cout<<"\nNODE is deleted successfully!!!\n";

}

else

{

student \*t1=start;

while(t1->next->next!=NULL)

{

t1=t1->next;

}

t1->next=NULL;

cout<<"NODE is deleted successfully!!!";

}

}

void search\_by\_RolNo()

{

system("CLS");

int roln,found=0;

cout<<"Enter a ROLL NO do you want to search : ";

cin>>roln;

if(start==NULL)

{

cout<<"NO node exists...\n";

}

else

{

student \*sr=start;

while(sr!=NULL)

{

if(sr->rolNo==roln)

{

cout<<"\t\t\t\t\t"<<"Rol No "<<"\t | \t "<<" Name \t"<<" | "<<"\tCGPA |"<<endl;

cout<<"\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout<<" \t\t\t\t\t"<<sr->rolNo<<" \t |\t "<<sr->name<<" \t |\t"<<sr->cgpa<<" \t |"<<endl;

found++;

}

sr=sr->next;

}

if(found==0)

{

cout<<"\nYour searching Roll NO not found...\n";

}

}

}

void search\_by\_name()

{

system("CLS");

string name1;

int found=0;

cout<<"\nEnter a Name do you want to search : ";

cin>>name1;

if(start==NULL)

{

cout<<"\nNO Node exists...!!!\n";

}

else

{

student \*sn=start;

while(sn!=NULL)

{

if(sn->name==name1)

{

cout<<"\t\t\t\t\t"<<"Rol No "<<"\t | \t "<<" Name \t"<<" | "<<"\tCGPA |"<<endl;

cout<<"\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout<<" \t\t\t\t\t"<<sn->rolNo<<" \t |\t "<<sn->name<<" \t |\t"<<sn->cgpa<<" \t |"<<endl;

found++;

}

sn=sn->next;

}

if(found==0)

{

cout<<"Your searching name not found...!!\n";

}

}

}

void searchbypos()

{

system("CLS");

int pos,found=0,count=0;

cout<<"\nEnter a position of node do you want to search : ";

cin>>pos;

if(start==NULL)

{

cout<<"\nList is Empty No node position exists.....\n";

}

else

{

student \*p=start;

while(p!=NULL)

{

count++;

if(pos==count)

{

cout<<"\n\n\t\t\t\t\t"<<"Rol No "<<"\t | \t "<<" Name \t"<<" | "<<"\tCGPA |"<<endl;

cout<<"\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout<<" \t\t\t\t\t"<<p->rolNo<<" \t |\t "<<p->name<<" \t |\t"<<p->cgpa<<" \t |"<<endl;

found++;

}

p=p->next;

}

if(found==0)

{

cout<<"\nNo position found here in list....!!!!\n";

}

}

}

void searchbyCGPA()

{

system("CLS");

float cgpa1;

int found=0;

cout<<"\nEnter a CGPA do you want to search : ";

cin>>cgpa1;

if(start==NULL)

{

cout<<"\nList is Empty Invalid CGPA....\n";

}

else

{

student \*c=start;

while(c!=NULL)

{

if(cgpa1==c->cgpa)

{

cout<<"\t\t\t\t\t"<<"Rol No "<<"\t | \t "<<" Name \t"<<" | "<<"\tCGPA |"<<endl;

cout<<"\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout<<" \t\t\t\t\t"<<c->rolNo<<" \t |\t "<<c->name<<" \t |\t"<<c->cgpa<<" \t |"<<endl;

found++;

}

c=c->next;

}

if(found==0)

{

cout<<"\nYour searching CGPA Not found...!!!\n";

}

}

}

void search\_by\_criteria()

{

system("CLS");

if(start==NULL)

{

cout<<"\nNO node exists...!!!\n";

}

else

{

student \*sc=start;

int les,gre;

cout<<"\nEnter Criteria of seaching roll No :\n";

cout<<"\nFor less than :";

cin>>les;

cout<<"\nFor greater than :";

cin>>gre;

cout<<"\t\t\t\t\t"<<"Rol No "<<"\t | \t "<<" Name \t"<<" | "<<"\tCGPA |"<<endl;

while(sc!=NULL)

{

if(sc->rolNo>gre && sc->rolNo<les )

{

cout<<"\t\t\t\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout<<" \t\t\t\t\t"<<sc->rolNo<<" \t |\t "<<sc->name<<" \t |\t"<<sc->cgpa<<" \t |"<<endl;

}

sc=sc->next;

}

}

}

void count\_a\_Node()

{

system("CLS");

int count=0;

if(start==NULL)

{

cout<<"\nNO node exists...!!!\n";

}

else

{

student \*c=start;

while(c!=NULL)

{

count=count+1;

c=c->next;

}

cout<<"The Total Number of Node is : "<<count<<endl;

}

}

void average\_cgpa()

{

system("CLS");

float avg,sum=0;

int num;

if(start==NULL)

{

cout<<"\nNo node exists....\n";

}

else

{

student \*a=start;

while(a!=NULL)

{

sum=sum+a->cgpa;

num=num+1;

a=a->next;

}

avg=float(sum/num);

cout<<"The average CGPA of student is :"<<avg<<endl;

}

}

void evenRolNshow()

{

system("CLS");

if(start==NULL)

{

cout<<"\nNo Node exists....\n";

}

else

{

student \*s=start;

int j=1;

cout<<"ROll No"<<"\t"<<" Name"<<"\t"<<"CGPA"<<endl;

while(s!=NULL)

{

if(s->rolNo%2==0)

{

cout<<"-------------------------------------\n";

cout<<"Record No :"<<j<<endl;

cout<<s->rolNo<<"\t"<<s->name<<"\t"<<s->cgpa<<"\t"<<endl;

j++;

}

s=s->next;

}

}

}

void cpga\_criteria()

{

system("CLS");

int count=0,j=1;

if(start==NULL)

{

cout<<"\nNo node exist...\n";

}

else

{

student \*c=start;

cout<<"ROll No"<<"\t"<<"Name"<<"\t"<<"CGPA"<<endl;

while(c!=NULL)

{

if(c->cgpa>3.0)

{

cout<<"-------------------------------------\n";

cout<<"Record No :"<<j<<endl;

cout<<c->rolNo<<"\t"<<c->name<<"\t"<<c->cgpa<<"\t"<<endl;

count=count+1;

j++;

}

c=c->next;

}

cout<<"The Total number of student that cgpa greater than Three is :"<<count<<endl;

}

}

void insertAfterpos()

{

system("CLS");

int pos,count=0,found=0;

cout<<"Enter a position please :";

cin>>pos;

if(start==NULL)

{

cout<<"\nList is empty So we insert Node at front..!!"<<endl;

student \*s=new student();

start=s;

s->next=NULL;

cout<<"\nEnter a Name : ";

cin>>s->name;

cout<<"\nEnter a Roll No : ";

cin>>s->rolNo;

cout<<"\nEnter a CGPA : ";

cin>>s->cgpa;

found++;

cout<<"\nNode is inserted succussfully....\n";

}

else

{

student \*t=start;

while(t!=NULL)

{

count++;

if(count==pos)

{

student \*t1=new student();

t1->next=t->next;

t->next=t1;

cout<<"\nEnter ROLL NO : ";

cin>>t1->rolNo;

cout<<"\nEnter NAME: ";

cin>>t1->name;

cout<<"\nEnter CGPA: ";

cin>>t1->cgpa;

cout<<"\nNode is inserted succussfully after given position...\n";

found++;

break;

}

else

{

t=t->next;

}

}

if(found==0)

{

cout<<"\nYour Enter position not exists so node inserted at END...."<<endl;

student \*e=new student();

cout<<"\nEnter a Name: ";

cin>>e->name;

cout<<"\nEnter a ROLL NO : ";

cin>>e->rolNo;

cout<<"\nEnter a CGPA : ";

cin>>e->cgpa;

e->next=NULL;

student \*s=start;

while(s->next!=NULL)

{

s=s->next;

}

s->next=e;

cout<<"\nNode is inserted succussfully...\n";

}

}

}

void insertAfterId()

{

system("CLS");

int id,found=0,count=0;

cout<<"\nPlease Enter ID Here : ";

cin>>id;

if(start==NULL)

{

cout<<"\nList is empty So we insert Node at front...!!!\n";

student \*s=new student();

s->next=NULL;

start=s;

cout<<"\nEnter a Name : ";

cin>>s->name;

cout<<"\nEnter a Roll No : ";

cin>>s->rolNo;

cout<<"\nEnter a CGPA : ";

cin>>s->cgpa;

found++;

cout<<"\nNode is inserted succussfully....\n";

}

else

{

student \*t=start;

while(t!=NULL)

{

count++;

if(id==t->rolNo)

{

student \*t1=new student();

t1->next=t->next;

t->next=t1;

cout<<"\nEnter a Roll No : ";

cin>>t1->rolNo;

cout<<"\nEnter a Name : ";

cin>>t1->name;

cout<<"\nEnter a CGPA : ";

cin>>t1->cgpa;

found++;

cout<<"\nNode is inserted after given id successfully...\n";

break;

}

else

{

t=t->next;

}

}

if(found==0)

{

cout<<"\nYour Enter ID not found So node inserted at END....!!!\n";

student \*e=new student();

cout<<"\nEnter a Name : ";

cin>>e->name;

cout<<"\nEnter a Roll NO : ";

cin>>e->rolNo;

cout<<"\nEnter a CGPA : ";

cin>>e->cgpa;

e->next=NULL;

student \*s=start;

while(s->next!=NULL)

{

s=s->next;

}

s->next=e;

cout<<"\nNode inserted successfully ....\n";

}

}

}

void insertafterName()

{

system("CLS");

string nam;

int found=0,count=0;

cout<<"\nPlease Enter a Name here : ";

cin>>nam;

if(start==NULL)

{

cout<<"\nList is Empty So we insert Node at front ....\n";

student \*s=new student();

s->next=NULL;

start=s;

cout<<"\nEnter a Name : ";

cin>>s->name;

cout<<"\nEnter a ROll No : ";

cin>>s->rolNo;

cout<<"\nEnter a CGPA : ";

cin>>s->cgpa;

found++;

cout<<"\nNode is inserted succussfully....\n";

}

else

{

student \*n=start;

while(n!=NULL)

{

count++;

if(nam==n->name)

{

student \*n1=new student();

n1->next=n->next;

n->next=n1;

cout<<"\nEnter a Name : ";

cin>>n1->name;

cout<<"\nEnter a Rol No : ";

cin>>n1->rolNo;

cout<<"\nEnter a CGPA : ";

cin>>n1->cgpa;

found++;

cout<<"\nNode is insert after given Name successfully....\n";

break;

}

else

{

n=n->next;

}

}

if(found==0)

{

cout<<"\nYour Enter Name not found here So node insert at END....\n";

student \*e=new student();

cout<<"\nEnter Name : ";

cin>>e->name;

cout<<"\nEnter Roll No : ";

cin>>e->rolNo;

cout<<"\nEnter CGPA : ";

cin>>e->cgpa;

e->next=NULL;

student \*s=start;

while(s->next!=NULL)

{

s=s->next;

}

s->next=e;

cout<<"\nNode is inserted successfully....\n";

}

}

}

void insertAfterCGPA()

{

system("CLS");

float cgpa1;

int count=0,found=0;

cout<<"\nEnter a CGPA here : ";

cin>>cgpa1;

if(start==NULL)

{

cout<<"\nList is Empty So we insert Node at front ....\n";

student \*s=new student();

s->next=NULL;

start=s;

cout<<"\nEnter a Name : ";

cin>>s->name;

cout<<"\nEnter a ROll No : ";

cin>>s->rolNo;

cout<<"\nEnter a CGPA : ";

cin>>s->cgpa;

found++;

cout<<"\nNode is inserted succussfully....\n";

}

else

{

student \*c=start;

while(c!=NULL)

{

count++;

if(cgpa1==c->cgpa)

{

student \*c1=new student();

c1->next=c->next;

c->next=c1;

cout<<"\nEnter a Name : ";

cin>>c1->name;

cout<<"\nEnter a Roll No : ";

cin>>c1->rolNo;

cout<<"\nEnter a CGPA :";

cin>>c1->cgpa;

found++;

cout<<"\nNode is inserted successfully after given cgpa....\n";

break;

}

else

{

c=c->next;

}

}

if(found==0)

{

cout<<"\nYour Enter CGPA not found So node inserted at END....!!!\n";

student \*e=new student();

cout<<"\nEnter Name : ";

cin>>e->name;

cout<<"\nEnter Roll No : ";

cin>>e->rolNo;

cout<<"\nEnter CGPA : ";

cin>>e->cgpa;

e->next=NULL;

student \*s=start;

while(s->next!=NULL)

{

s=s->next;

}

s->next=e;

cout<<"\nNode is inserted successfully....\n";

}

}

}

void deletenodebypos()

{

system("CLS");

int pos,count=0,found=0;

cout<<"\nEnter the position : ";

cin>>pos;

student \*d=start;

if(start==NULL)

{

cout<<"\nList is Empty and invalid position..!!\n";

}

else if(start->next==NULL && pos==1)

{

start=NULL;

cout<<"\nNo is deleted succusfully by position...\n";

found++;

}

else

{

pos--;

while(d!=NULL)

{

count++;

if(count==pos)

{

if(d->next->next!=NULL)

{

d->next=d->next->next;

cout<<"\nNode is deleted by given position..!!\n";

found++;

break;

}

else

{

d->next=d->next->next;

cout<<"\nNode is deleted by given position..!!\n";

found++;

break;

}

}

else

{

d=d->next;

}

}

}

if(found==0 && start!=NULL)

{

cout<<"\nNo Node is deleted because ID not found here...!!!\n";

system("pause");

}

}

void deleteNodebyId()

{

system("CLS");

int roln,found=0;

cout<<"\nEnter a Roll num do you want to delete: ";

cin>>roln;

if(start==NULL)

{

cout<<"\nList is Empty invalid ID....!!!\n";

}

else if(start->rolNo==roln && start->next==NULL)

{

start=NULL;

cout<<"\nNode is deleted succussfully by Roll No....\n";

found++;

}

else if(start->rolNo==roln && start->next!=NULL)

{

start=start->next;

cout<<"\nNode is deleted successfully by Roll No...\n";

found++;

}

else

{

student \*d=start;

while(d!=NULL)

{

if(roln==d->next->rolNo)

{

if(d->next->next!=NULL)

{

d->next=d->next->next;

cout<<"\nNode is deleted successfully by roll No...\n";

found++;

break;

}

else

{

d->next=d->next->next;

cout<<"\nNode is deleted successfully by roll No...\n";

found++;

break;

}

}

else

{

d=d->next;

}

}

}

if(start!=NULL && found==0)

{

cout<<"\nNo Node is deleted because ID not found here...!!!\n";

}

}

void deletenodebyname()

{

system("CLS");

string nam;

int found=0;

cout<<"\nEnter a Name do you want to delete Node : ";

cin>>nam;

if(start==NULL)

{

cout<<"\nList is Empty And invalid Name...!!\n";

}

else if(start->next==NULL && start->name==nam)

{

start=NULL;

cout<<"\nNode is deleted succussfully by name...\n";

found++;

}

else if(start->next!=NULL && start->name==nam)

{

start=start->next;

cout<<"\nNode is deleted succussfully by name...\n";

found++;

}

else

{

student \*n=start;

while(n!=NULL)

{

if(nam==n->next->name)

{

if(n->next->next!=NULL)

{

n->next=n->next->next;

found++;

cout<<"\nNode is deleted succussfully by name....\n";

break;

}

else

{

n->next=n->next->next;

cout<<"\nNode is deleted succussfully by name...\n";

found++;

break;

}

}

else

{

n=n->next;

}

}

}

if(start!=NULL && found==0)

{

cout<<"\nNo Node is deleted besause name Not found here..!!!\n";

}

}

void deletenodebyCGPA()

{

system("CLS");

float cgpa1;

int found=0;

cout<<"\nEnter a CGPA do you want to delete Node : ";

cin>>cgpa1;

if(start==NULL)

{

cout<<"\nList is empty and invalid CGPA...!!!\n";

}

else if(start->next==NULL && start->cgpa==cgpa1)

{

start=NULL;

cout<<"\nNode is deleted successfully by CGPA...\n";

found++;

}

else if(start->next!=NULL && start->cgpa==cgpa1)

{

start=start->next;

cout<<"\nNode is deleted successfully by CGPA...\n";

found++;

}

else

{

student \*c=start;

while(c!=NULL)

{

if(cgpa1==c->next->cgpa)

{

if(c->next->next!=NULL)

{

c->next=c->next->next;

cout<<"\nNode is deleted successfully by CGPA...\n";

found++;

break;

}

else

{

c->next=c->next->next;

cout<<"\nNode is deleted successfully by CGPA...\n";

found++;

break;

}

}

else

{

c=c->next;

}

}

if(found==0)

{

cout<<"\nNo node is deleted because CGPA not found here...!!!\n";

}

}

if(start!=NULL && found==0)

{

cout<<"\nNo node is deleted because CGPA not found here...!!!\n";

}

}

};

int main()

{

int opt;

student s;

do

{

system("CLS");

cout<<"\nPress 1 to Insert node at start : \t\t\t Press 12 to show student that cgpa>3:\n";

cout<<"\nPress 2 to display node : \t\t\t \t Press 13 to insert After position :\n ";

cout<<"\nPress 3 to delete node at start : \t\t \t Press 14 to delete Node by position :\n";

cout<<"\nPress 4 to Insert Node at END : \t\t\t Press 15 to insert node after id :\n";

cout<<"\nPress 5 to DELETE at END Node: \t\t\t\t Press 16 to insert node after Name :\n";

cout<<"\nPress 6 to search by ROLL NO : \t\t\t\t Press 17 to insert node after CGPA :\n";

cout<<"\nPress 7 to search by Name : \t\t\t\t Press 18 to search node by position :\n";

cout<<"\nPress 8 to count Number of Nodes : \t\t\t Press 19 to search node by CGPA :\n";

cout<<"\nPress 9 to search Node by roll No criteria : \t\t Press 20 to delete node by ID :\n";

cout<<"\nPress 10 to show average cgpa of all student : \t\t Press 21 to delete node by Name :\n";

cout<<"\nPress 11 to show Even ROll no student : \t\t Press 22 to delete node by CGPA :\n";

cout<<"\n\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\t\t\t\t\tEnter your choice :";

cin>>opt;

switch(opt)

{

case 1:

s.insertatstart();

break;

case 2:

s.display();

break;

case 3:

s.deleteatstart();

break;

case 4:

s.insertAtEnd();

break;

case 5:

s.deleteatEnd();

break;

case 6:

s.search\_by\_RolNo();

break;

case 7:

s.search\_by\_name();

break;

case 8:

s.count\_a\_Node();

break;

case 9:

s.search\_by\_criteria();

break;

case 10:

s.average\_cgpa();

break;

case 11:

s.evenRolNshow();

break;

case 12:

s.cpga\_criteria();

break;

case 13:

s.insertAfterpos();

break;

case 14:

s.deletenodebypos();

break;

case 15:

s.insertAfterId();

break;

case 16:

s.insertafterName();

break;

case 17:

s.insertAfterCGPA();

break;

case 18:

s.searchbypos();

break;

case 19:

s.searchbyCGPA();

break;

case 20:

s.deleteNodebyId();

break;

case 21:

s.deletenodebyname();

break;

case 22:

s.deletenodebyCGPA();

break;

default:

cout<<"Invalid input !!!!\n";

}

cout<<"\nDo you want to repeat then press 1 otherwise 0: ";

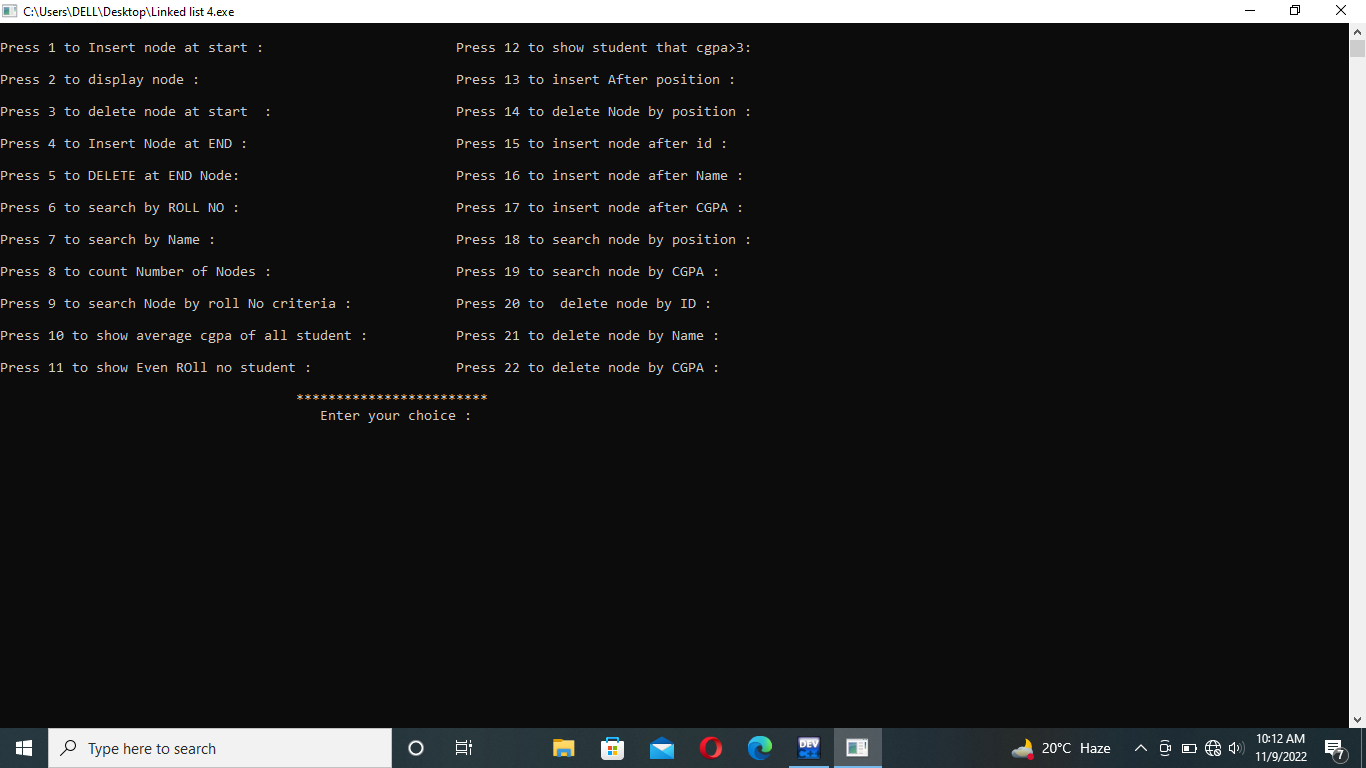
cin>>opt;

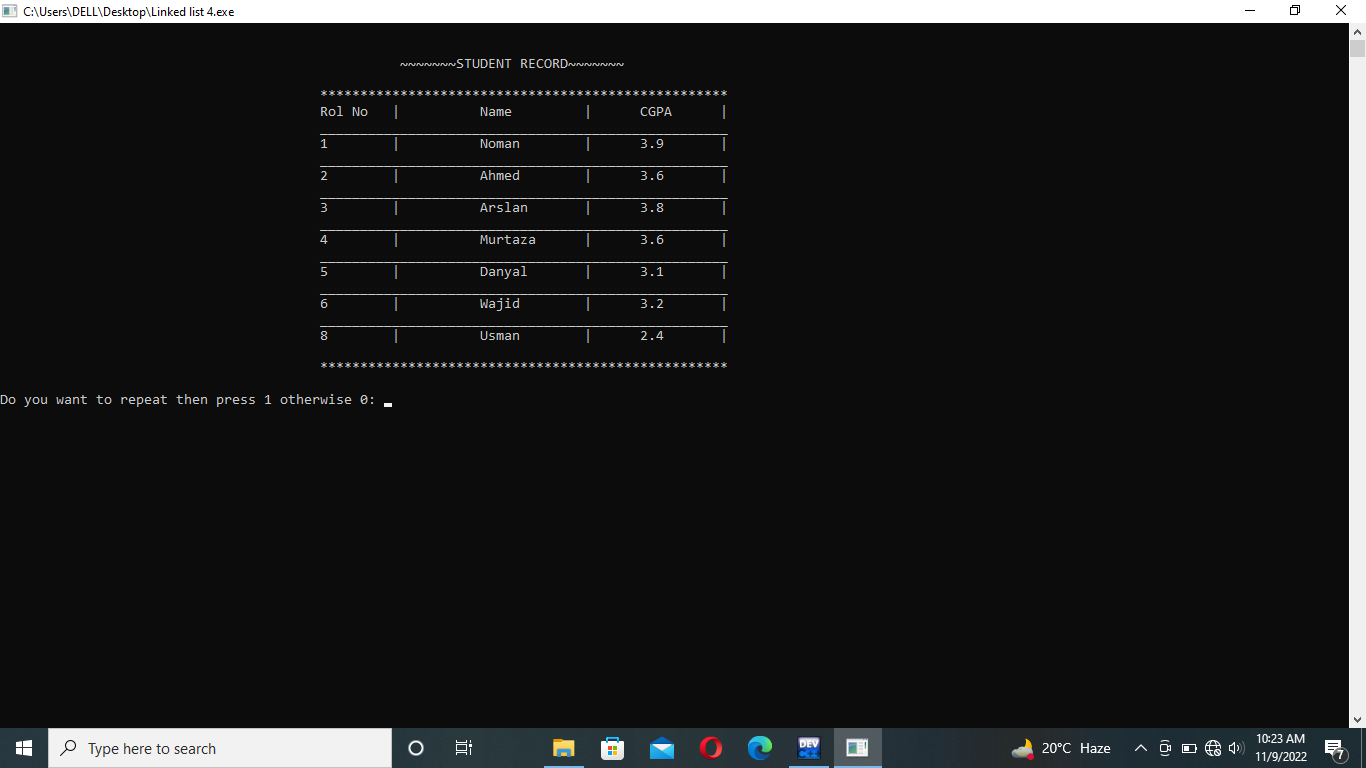
}while(opt==1);

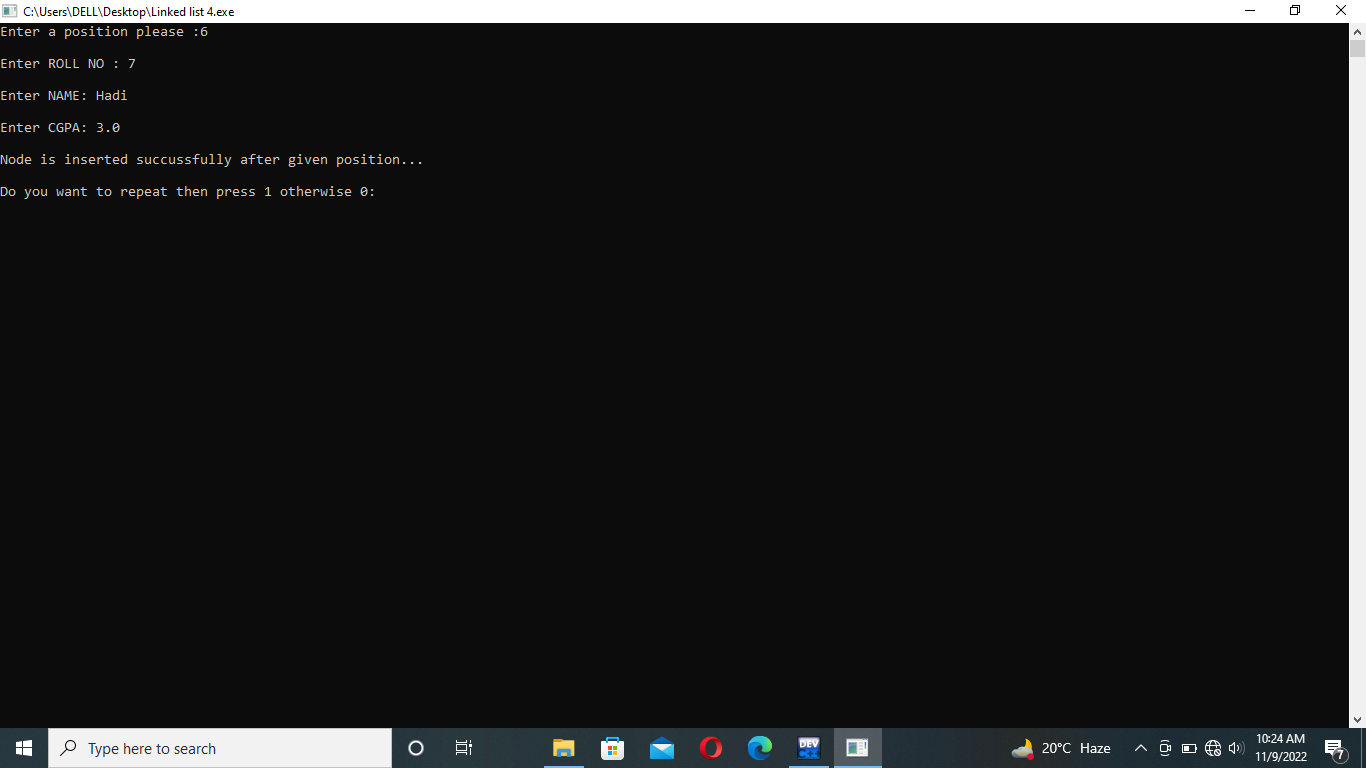
}

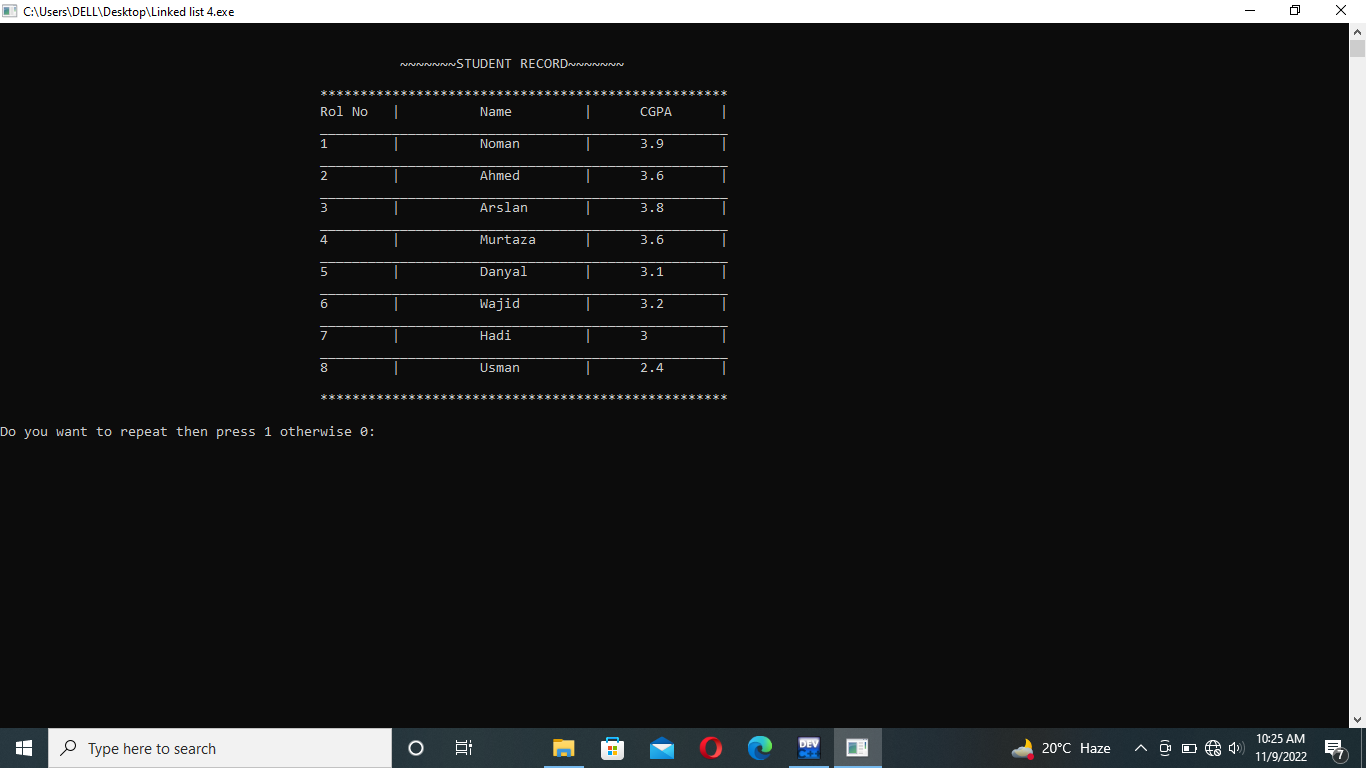
OUTPUT:

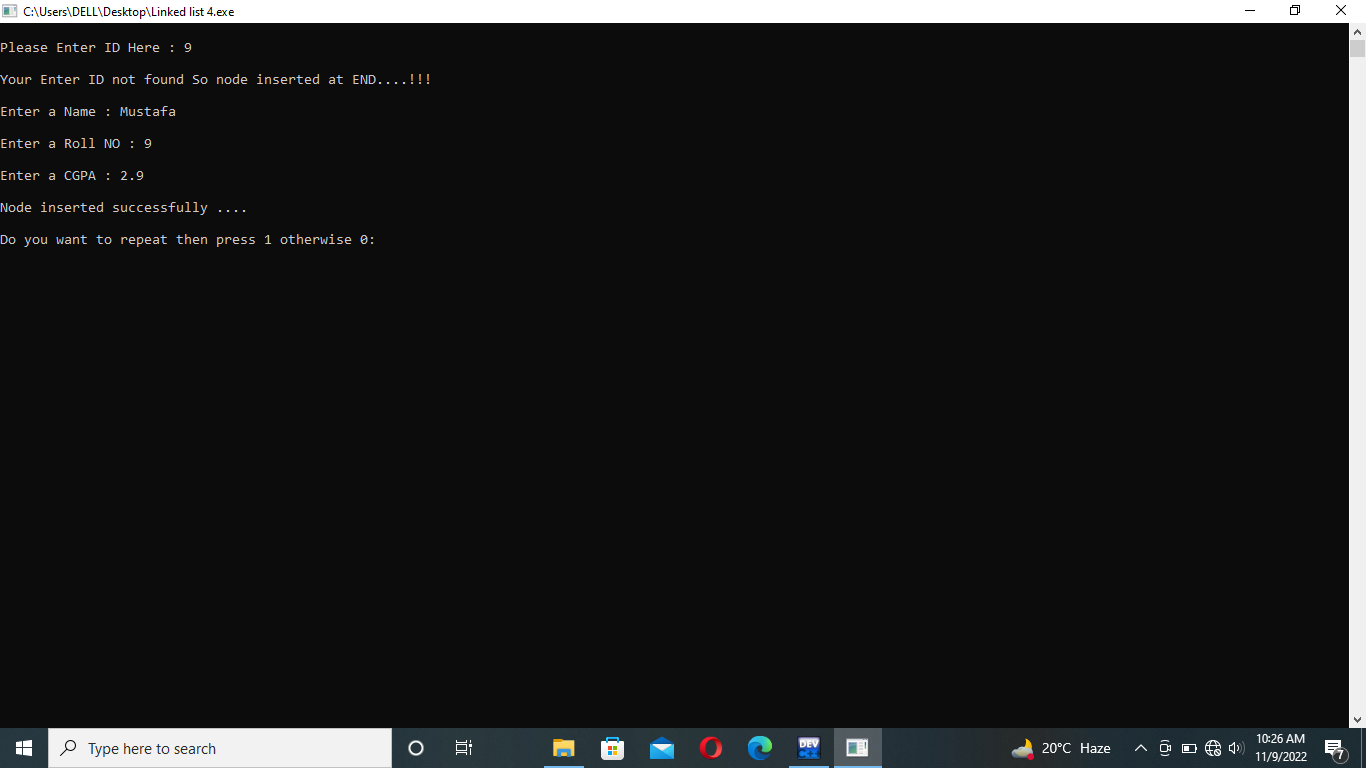
->Select your choice:

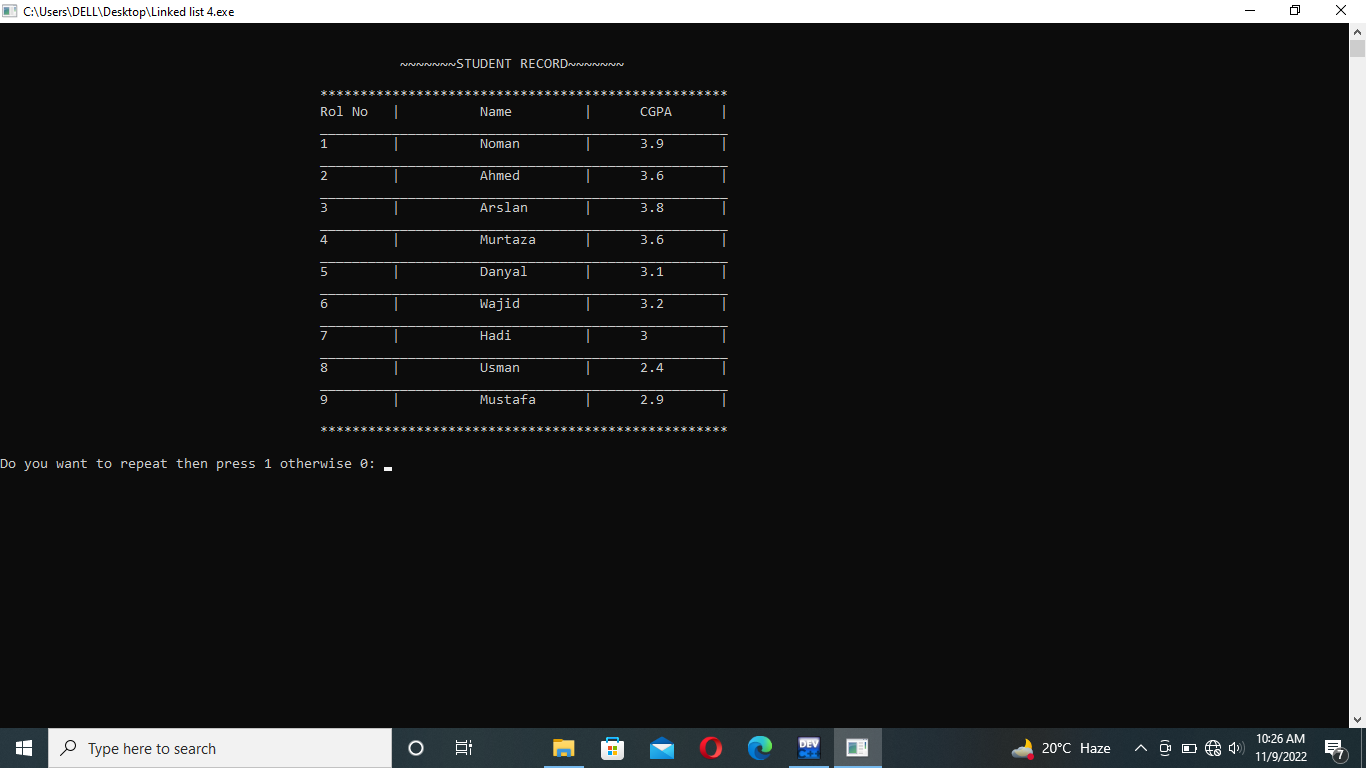


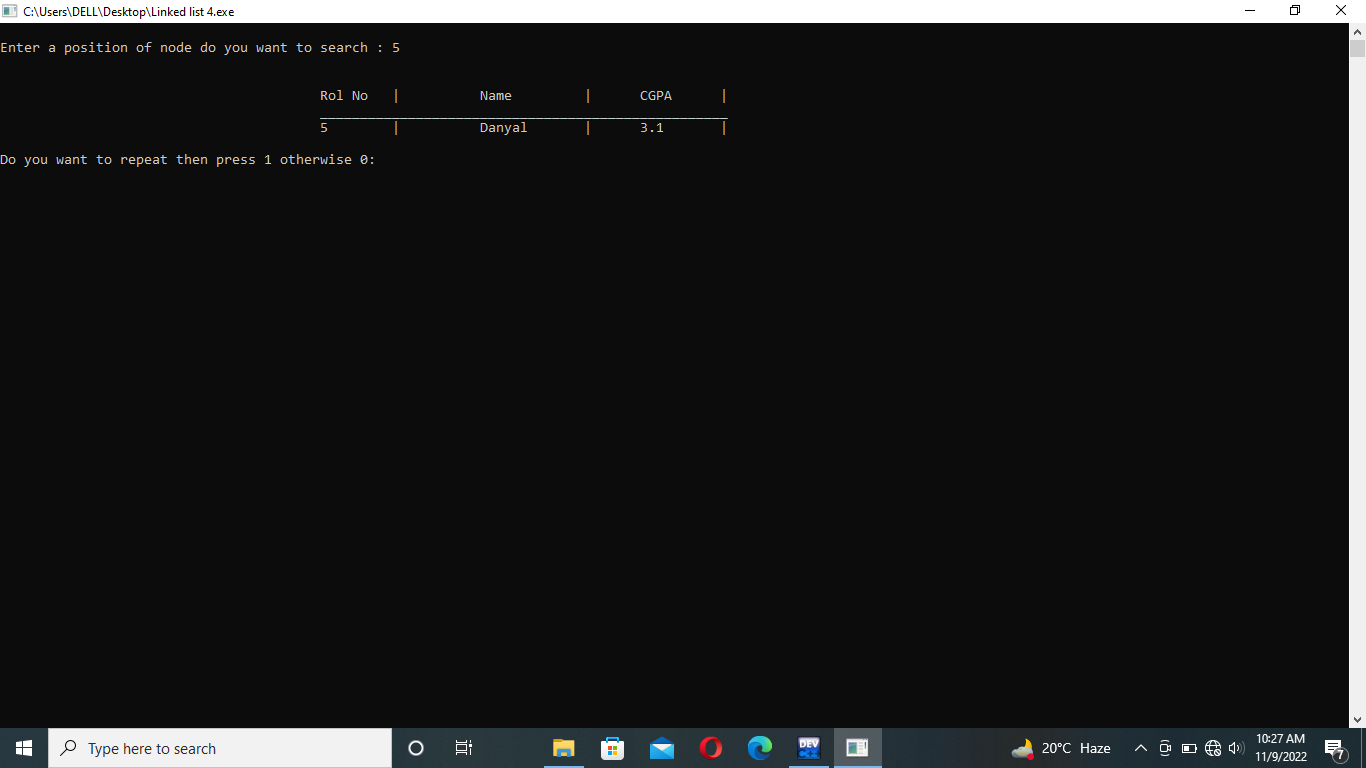
->Entered data Display :--

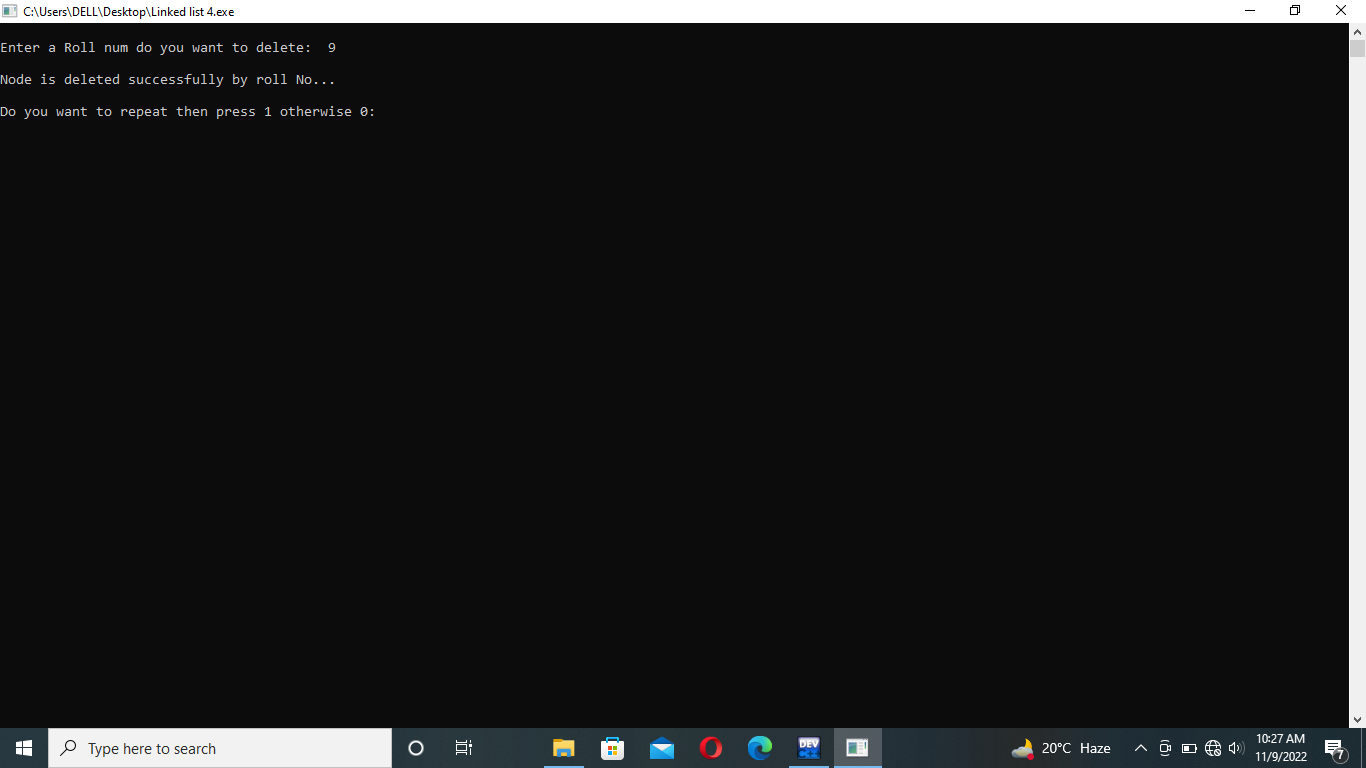
->Insert after position:

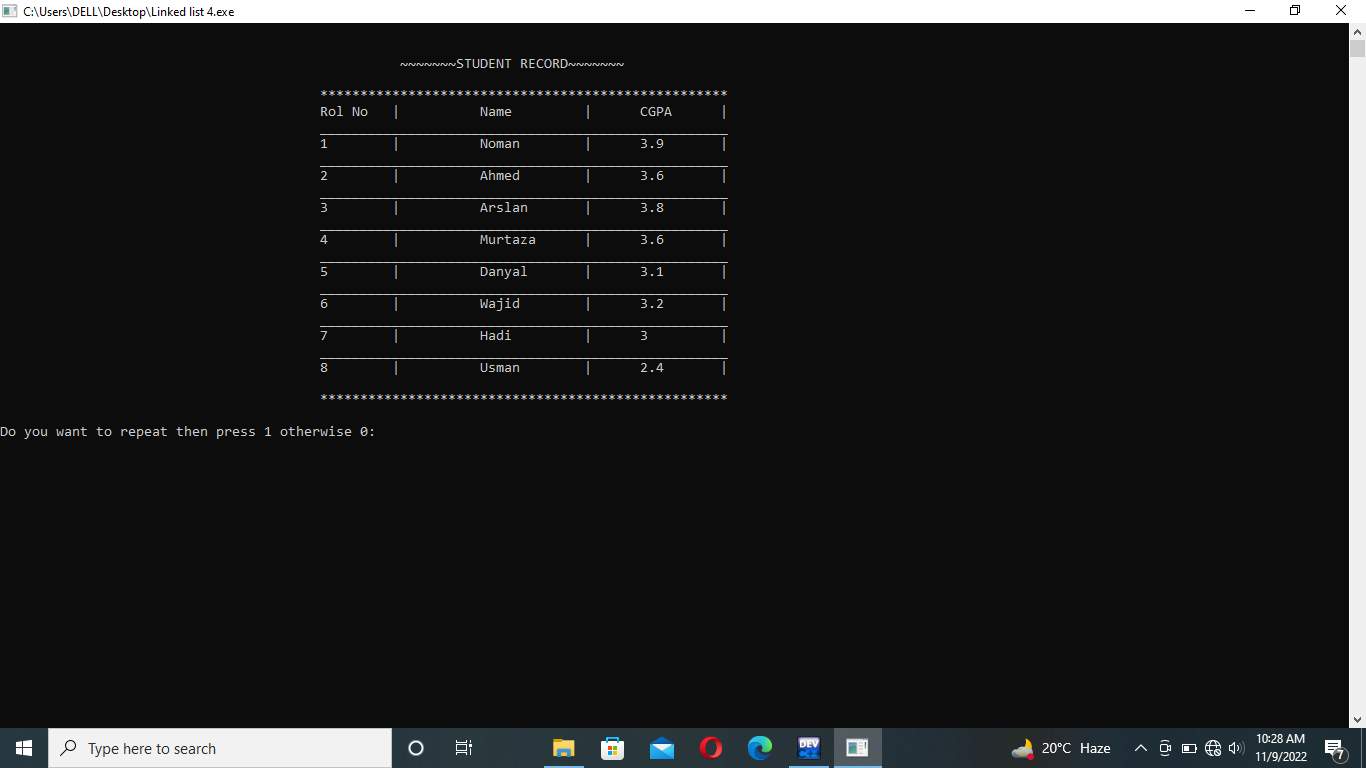
->display insert after position:-

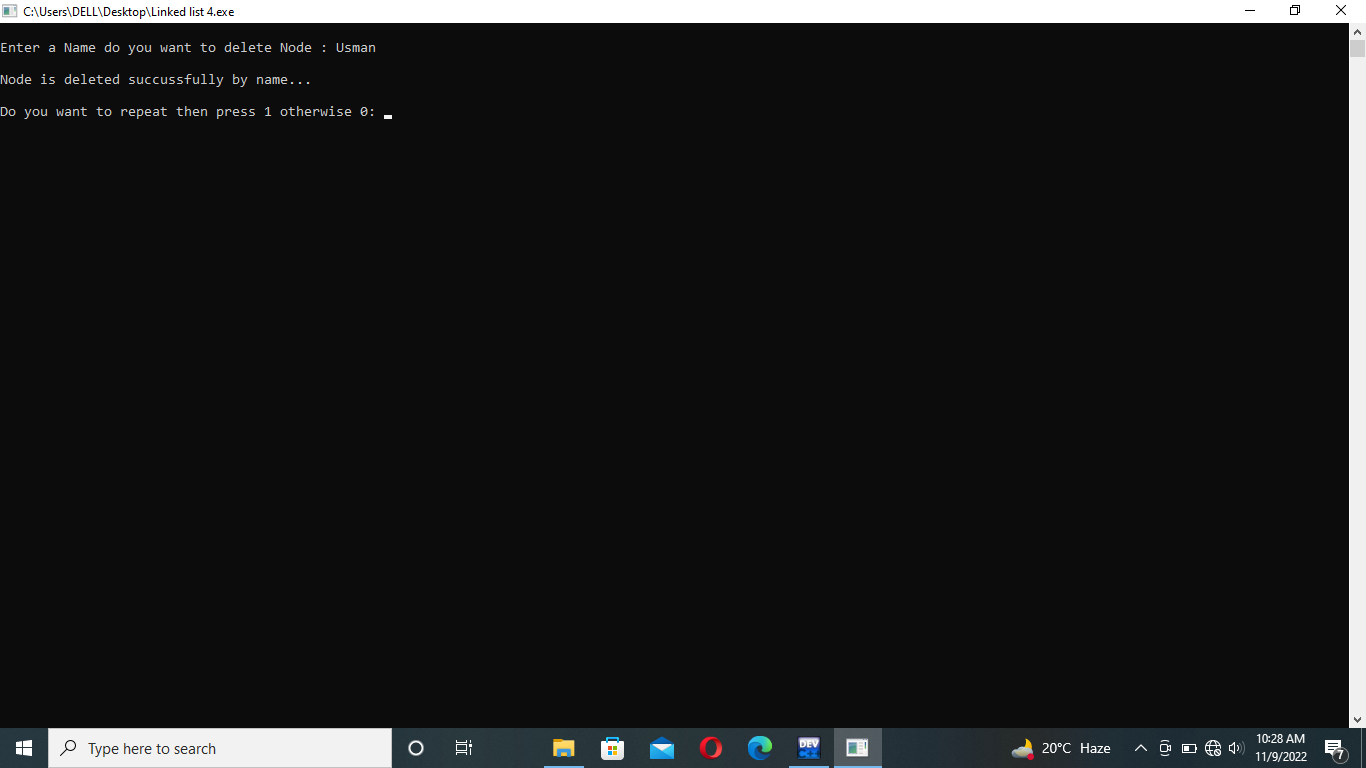
->insert after ID:

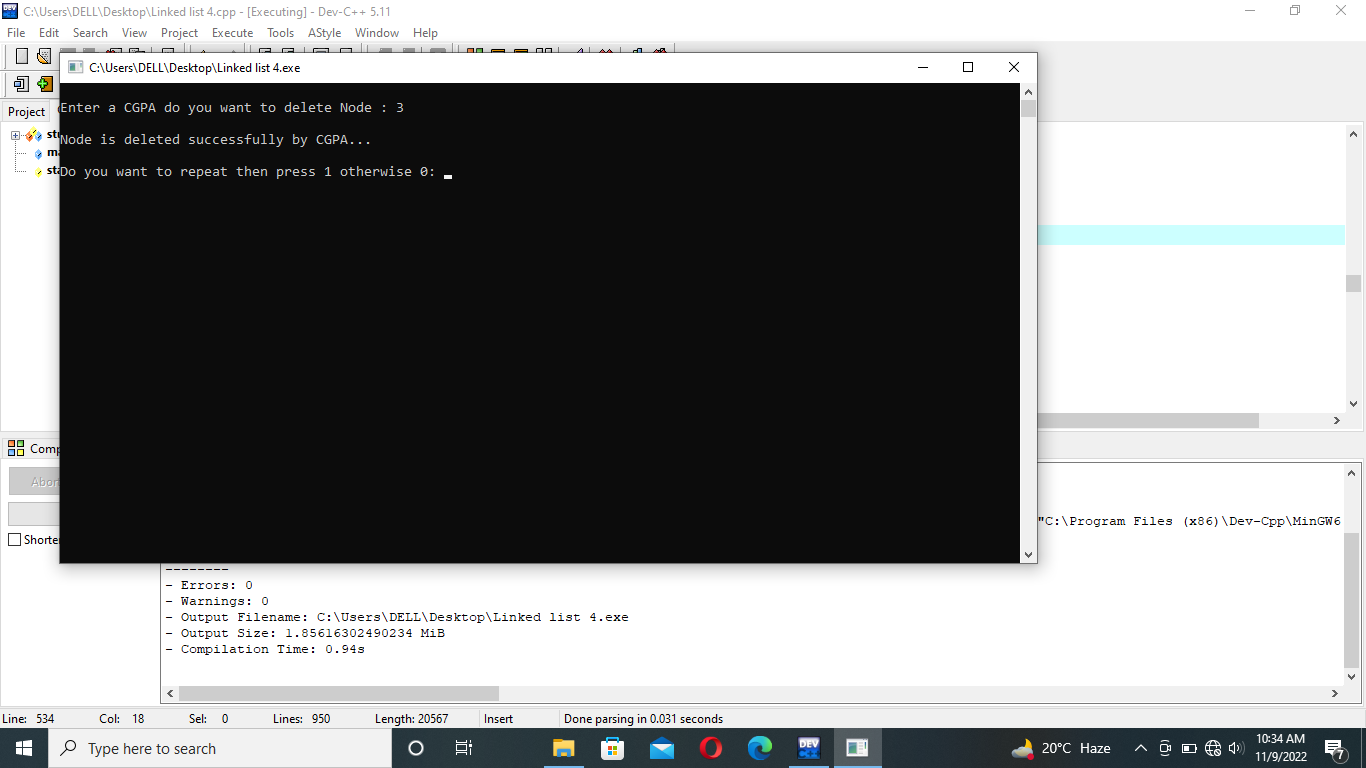
->Display insert after ID:

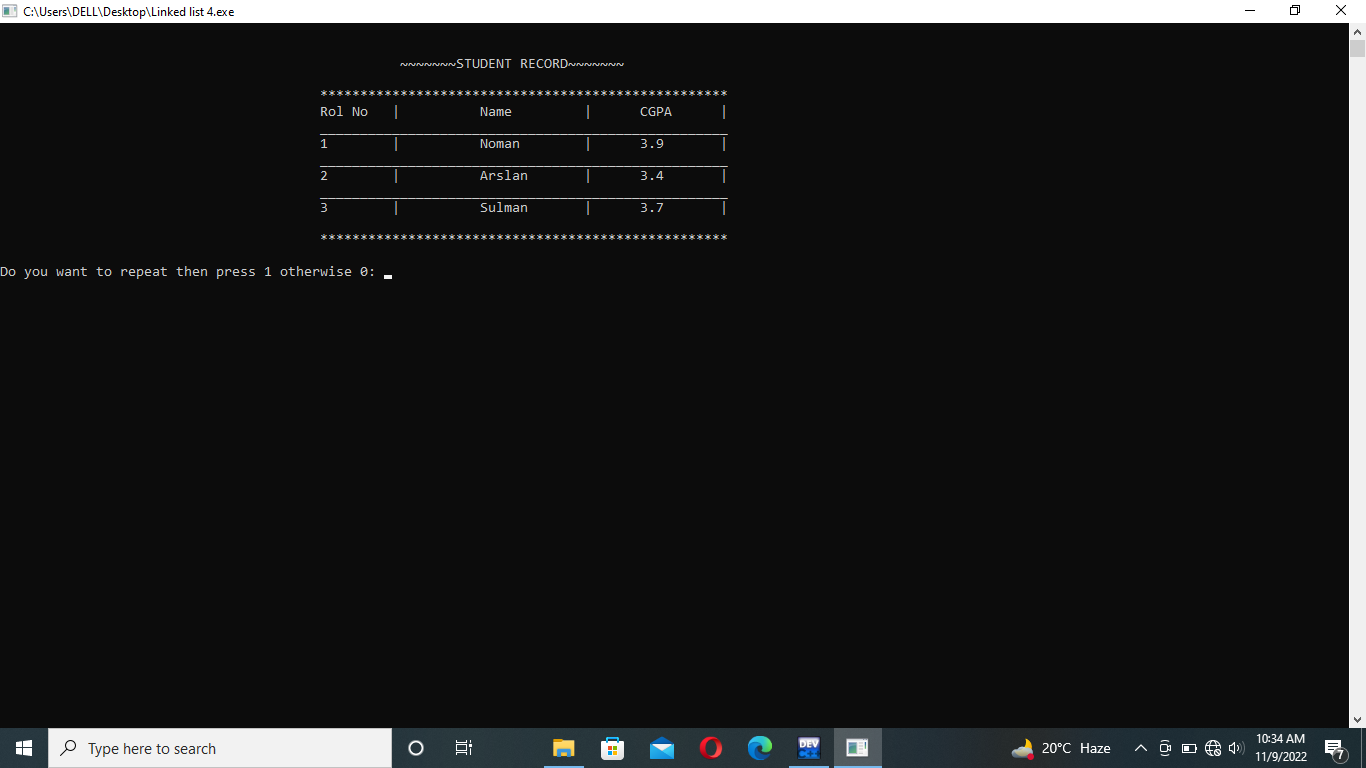
>Search by position:

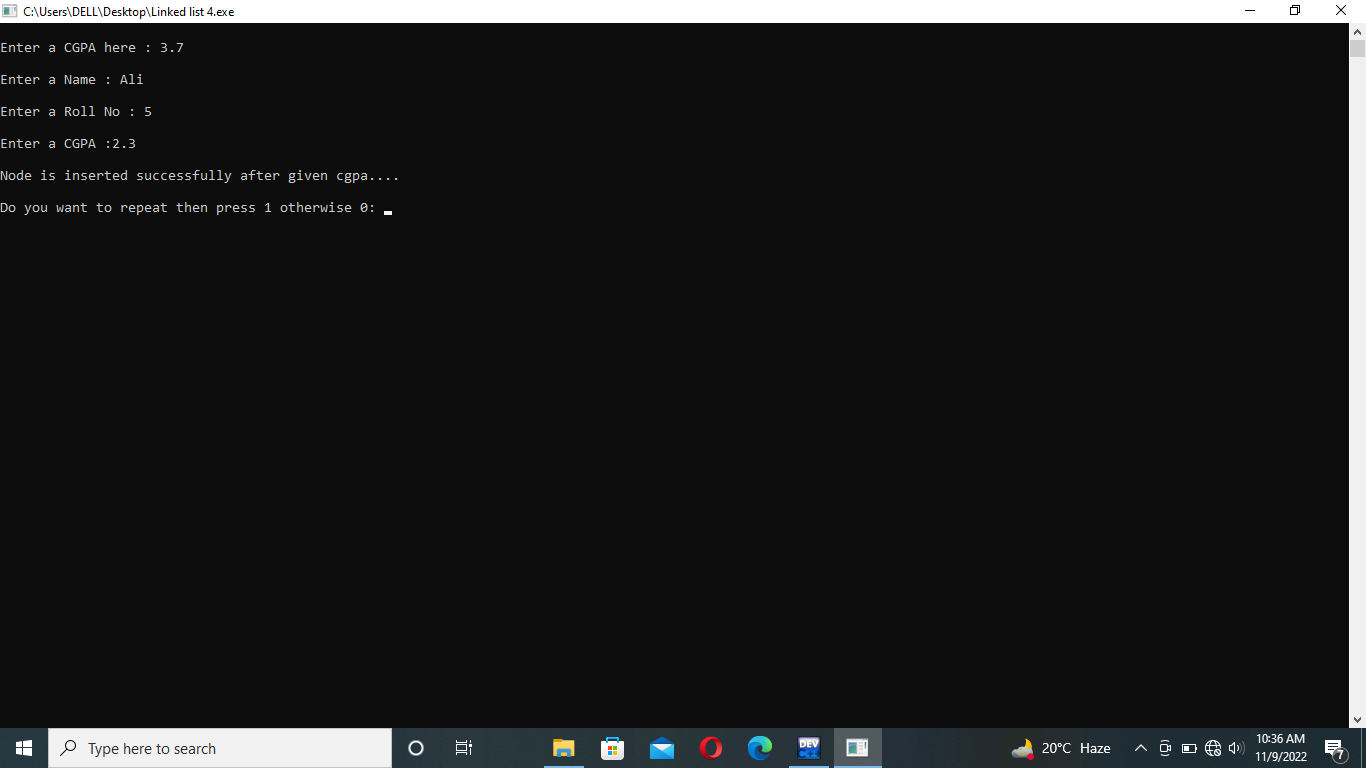
->DELETE by roll NO:

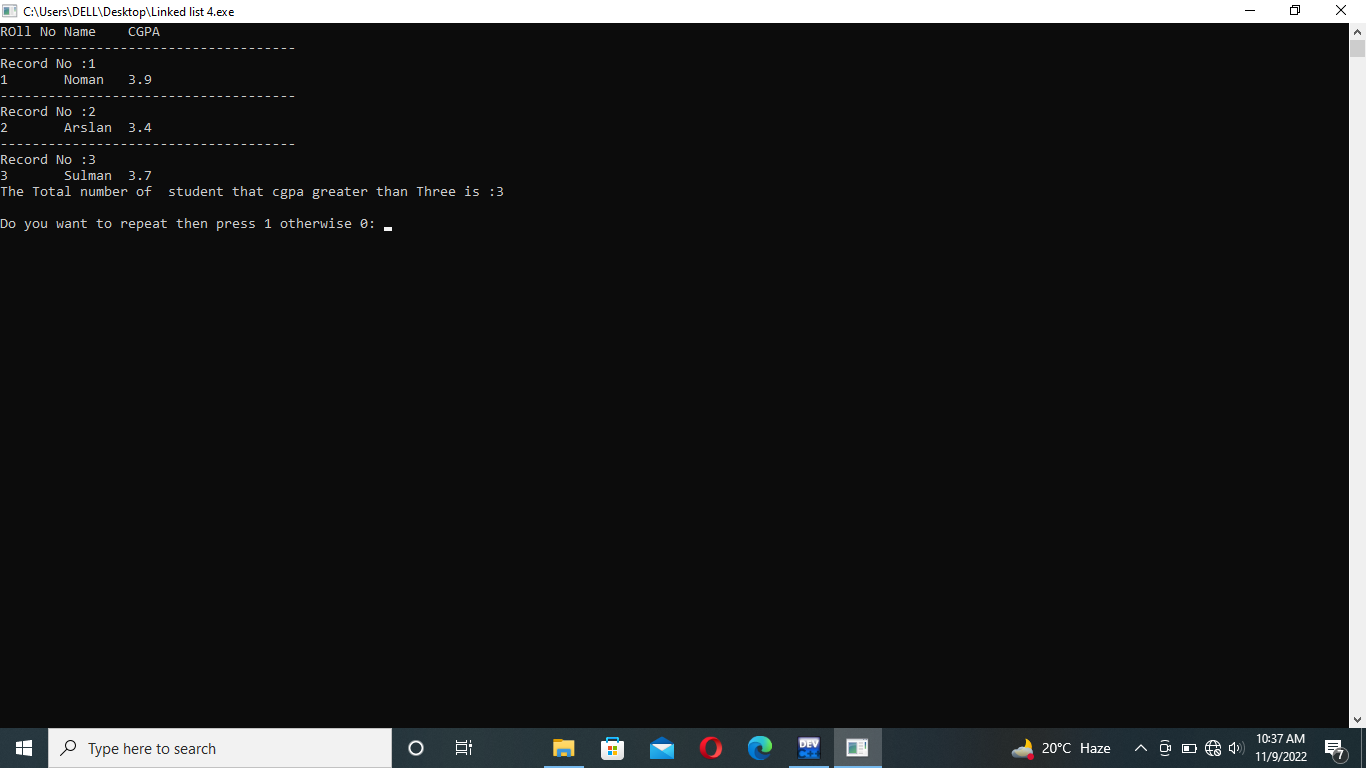
->display after DELETE by roll NO;

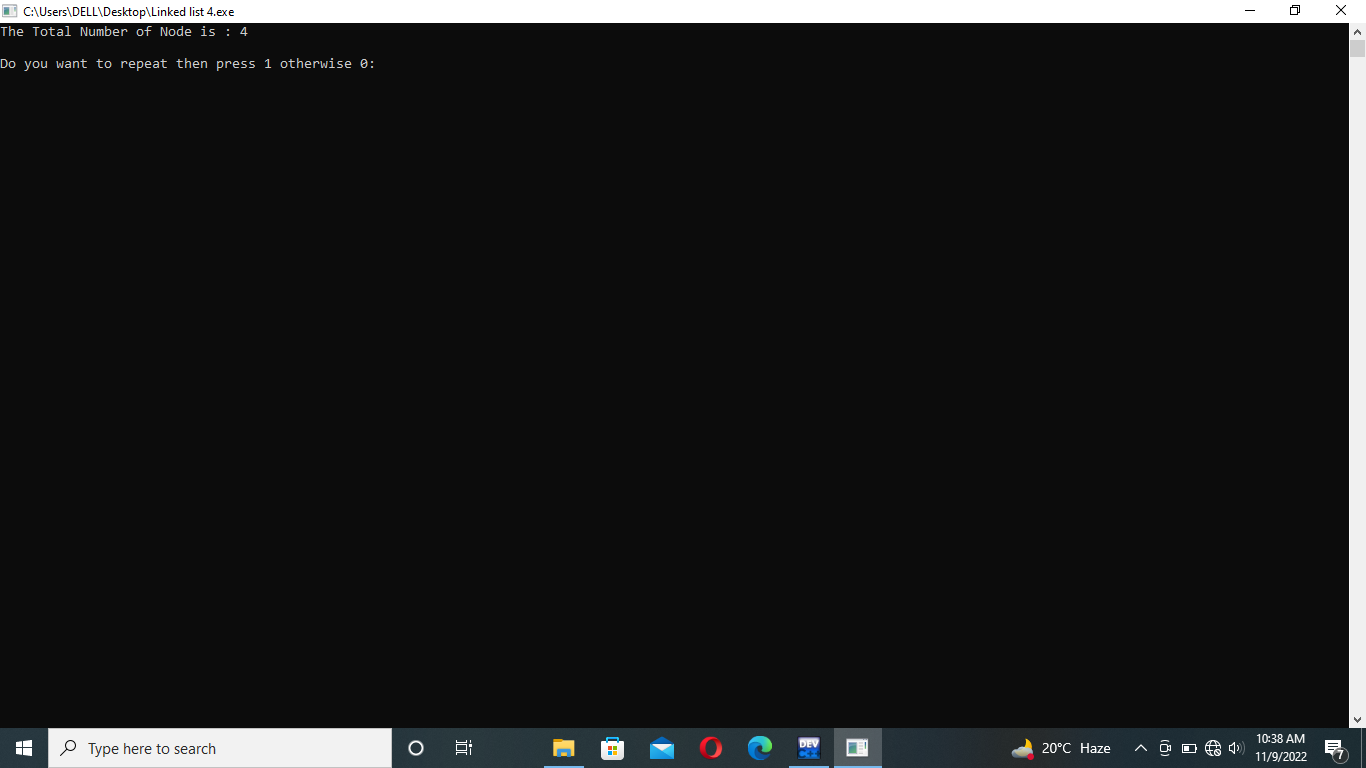
->DELETE by name:

->delete by CGPA:

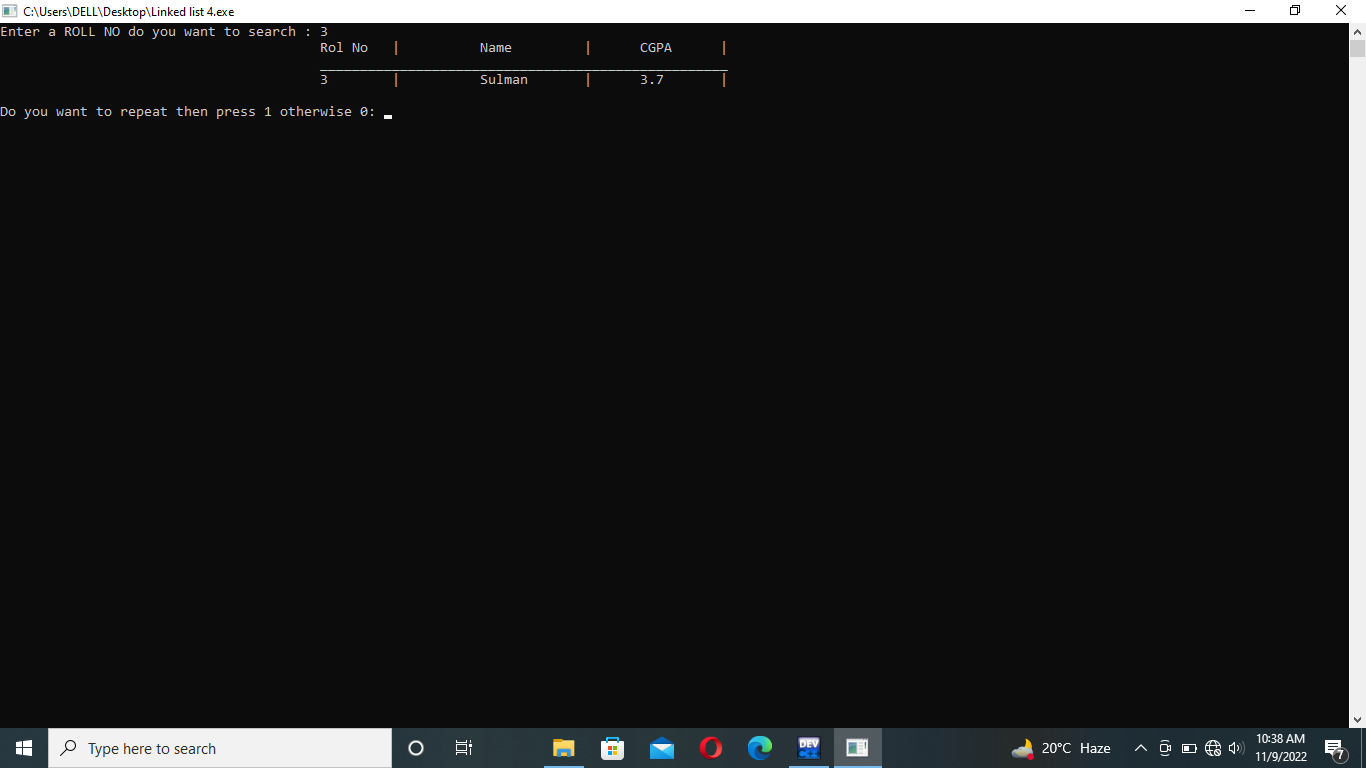
->display after delete by CGPA:

->insert after CGPA:

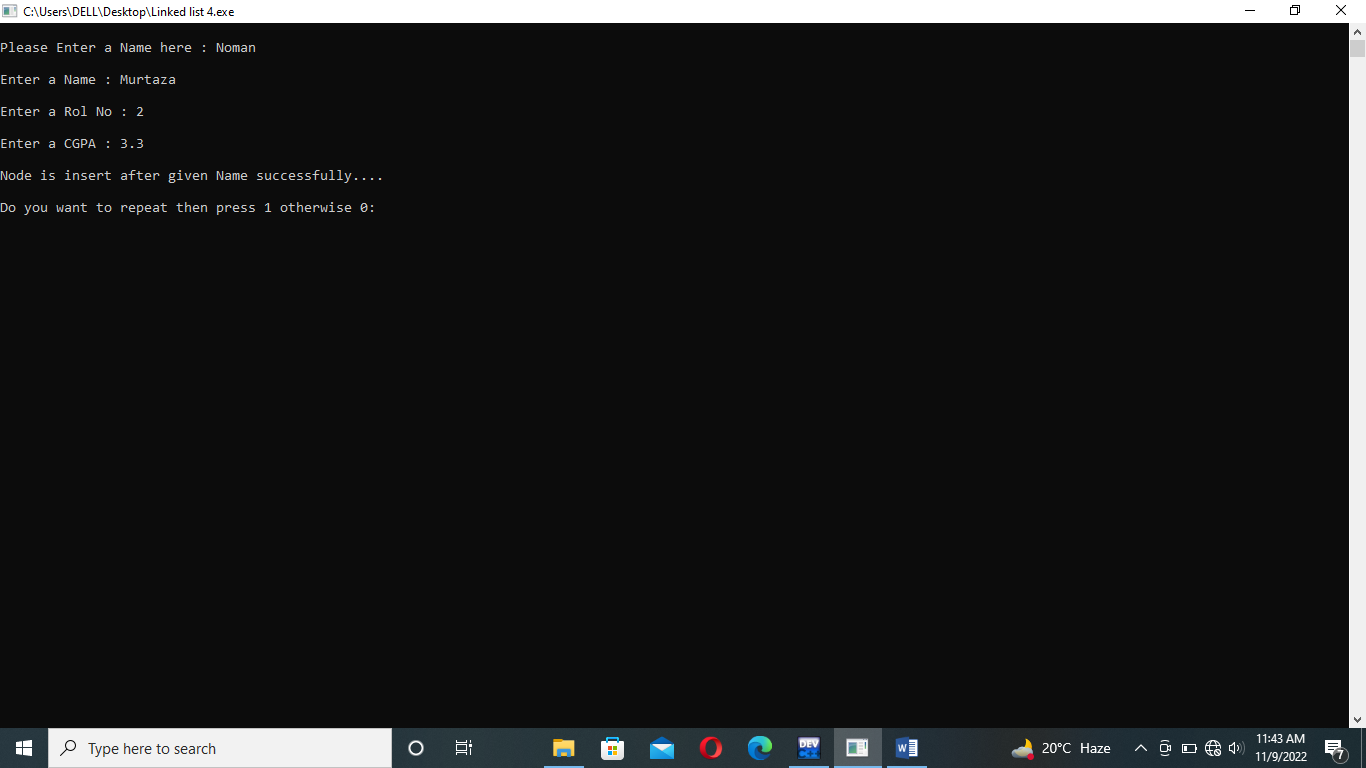
->CGPA criteria greater than 3:

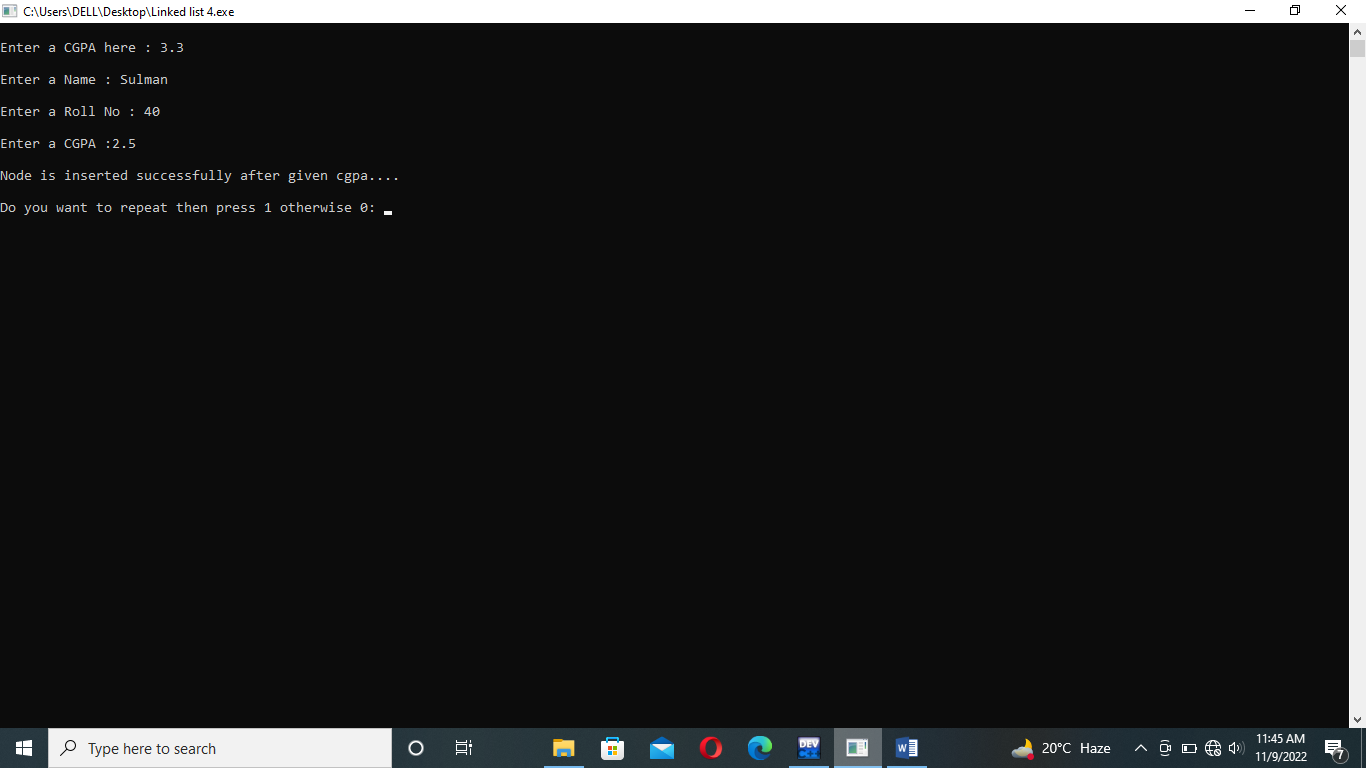
->Count total Number of Node:

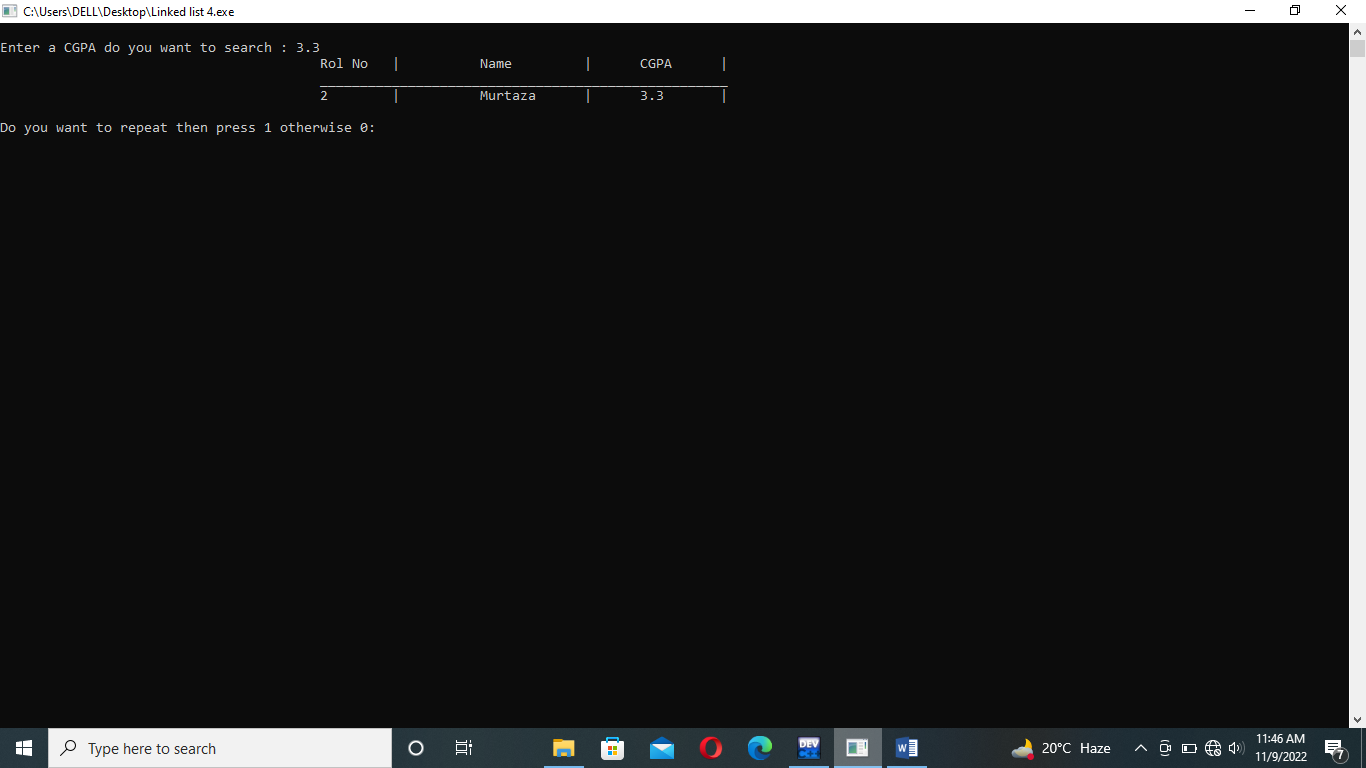
->Search by roll No:

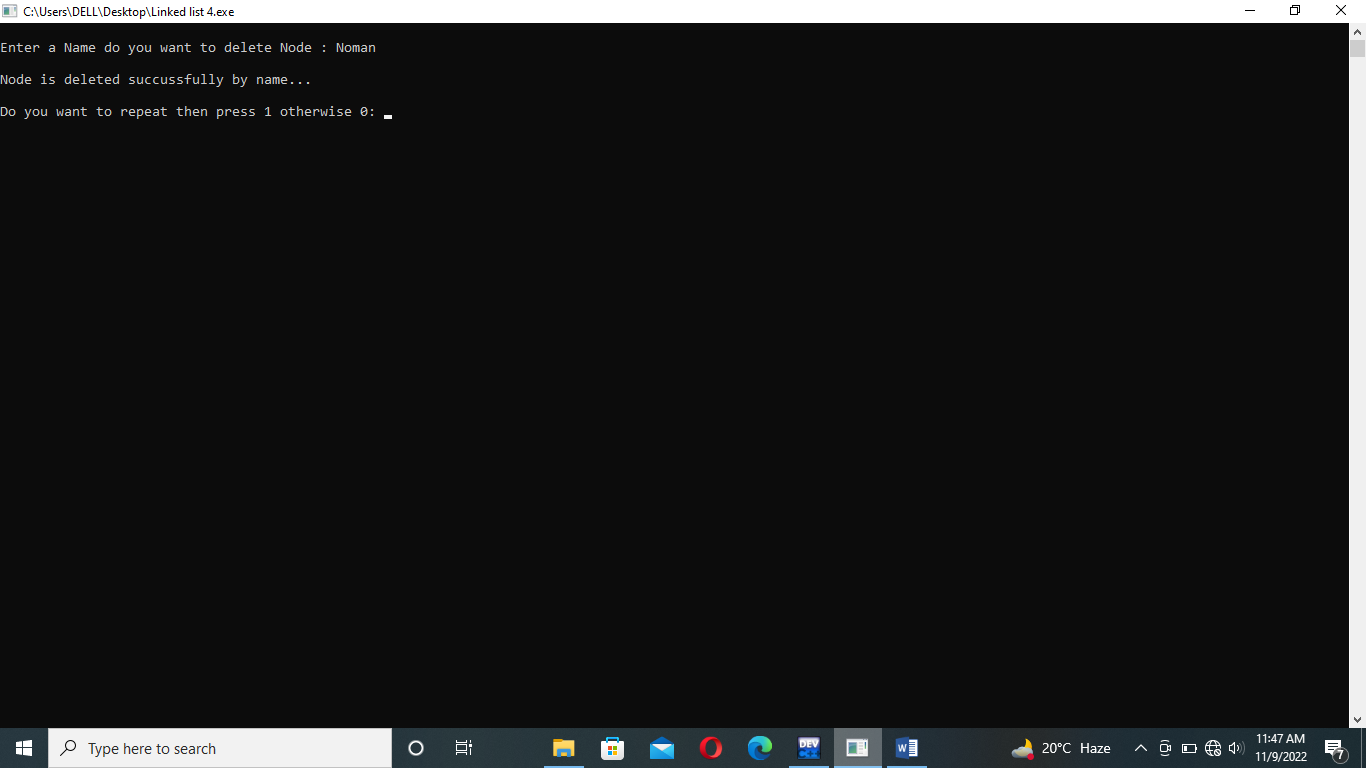


->Insert after Name:



->Insert after CGPA:

->Search by CGPA:

->DELETE BY NAME:

->Display after delete by name,insert after name,insert after cgpa.