**Assignment Number: 3**

**Subject: Data Structure and Algorithms**

**Name: Shrirang Mhalgi**

**Roll No.:222006**

**Class: S.E**

**Division: B**

**Batch: B1**

**Title/Problem Statement:**

Write C/C++ program for storing matrix. Write functions for

a) Check whether given matrix is upper triangular or not

b) Compute summation of diagonal elements

c) Compute transpose of matrix d) Add, subtract and multiply two matrices

**CODE**

#include <iostream>

using namespace std;

class set

{

int M[30],cri[30],bad[30],i,j,k,con,cob,n,cn,bn,cab[30],temp;

int cv[30],play\_both[30],onc[30],onb[30],tmp[30],can[30];

int cuni,cint,conc,conb,cnne;

public:

void Str\_n()

{

cout<<"Enter the Class Strength :"<<endl;

for (;;)

{

if (cin >> n)

{

break;

}

else

{

cout << "Please enter a valid integer" << endl;

cin.clear();

//cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

cob=0;

con=0;

cuni=0;

cint=0;

conc=0;

conb=0;

cnne=0;

}

void cri\_str\_n()

{

cout<<"Number of Student playing Cricket :"<<endl;

for (;;)

{

if (cin >> cn)

{

break;

}

else

{

cout << "Please enter a valid integer" << endl;

cin.clear();

//cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

}

void bad\_str\_n()

{

cout<<"Number of Student playing Badminton :"<<endl;

for (;;)

{

if (cin >> bn)

{

break;

}

else

{

cout << "Please enter a valid integer" << endl;

cin.clear();

//cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

}

void Strength()

{

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

{

onc[i]=0;

onb[i]=0;

M[i]=0;

}

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

{

M[i]=i+1;

}

}

void Cricket()

{

con=0;

cout<<"Students playing Cricket :"<<endl;

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

{

cin>>cri[i];

con++;

cob++;

}

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

{

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

if (cri[i]<cri[j])

{

int tp=cri[i];

cri[i]=cri[j];

cri[j]=tp;

}

}

}

void Badminton()

{

cout<<"Students playing Badminton :"<<endl;

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

{

cin>>bad[i];

con++;

cob++;

}

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

{

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

if (bad[i]<bad[j])

{

int tp=bad[i];

bad[i]=bad[j];

bad[j]=tp;

}

}

}

void Play\_both()

{

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

{

for(k=0;k<bn;k++)

{

if (cri[j]==bad[k])

{

play\_both[cint]=cri[j];

cint++;

cob--;

}

}

}

}

void Play\_any()

{

i=0;

for(j=0;j<cn;j++,i++)

{

cab[i]=cri[j];

}

for(k=0;k<bn;k++,i++)

{

cab[i]=bad[k];

}

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

{

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

if (cab[i]<cab[j])

{

int tp=cab[i];

cab[i]=cab[j];

cab[j]=tp;

}

}

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

{

if (cab[j]!=cab[j+1])

{

cv[cuni]=cab[j];

cuni++;

}

}

}

void Play\_none()

{

i=0;

con=n+cuni;

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

{

cab[i]=M[j];

}

for(k=0;k<cuni;k++,i++)

{

cab[i]=cv[k];

}

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

{

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

if (cab[i]<cab[j])

{

int tp=cab[i];

cab[i]=cab[j];

cab[j]=tp;

}

}

j=0;

while(j<con)

{

if (cab[j]!=cab[j+1] && j<con-1)

{

can[cnne]=cab[j];

cnne++;

j++;

}

if(cab[j]==cab[j+1])

{

j++;

j++;

}

if(j==con-1 && cab[j]!=cab[j-1])

{

can[cnne]=cab[j];

cnne++;

j++;

}

}

}

void OnCri()

{

i=0;

con=cn+cint;

for(j=0;j<cn;j++,i++)

{

tmp[i]=cri[j];

}

for(k=0;k<cint;k++,i++)

{

tmp[i]=play\_both[k];

}

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

{

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

if (tmp[i]<tmp[j])

{

int tp=tmp[i];

tmp[i]=tmp[j];

tmp[j]=tp;

}

}

conc=0;

j=0;

while(j<con)

{

if (tmp[j]!=tmp[j+1] && j<con-1)

{

onc[conc]=tmp[j];

conc++;

j++;

}

if(tmp[j]==tmp[j+1])

{

j++;

j++;

}

if(j==con-1 && tmp[j]!=tmp[j-1])

{

onc[conc]=tmp[j];

conc++;

j++;

}

}

}

void OnBad()

{

i=0;

con=bn+cint;

for(j=0;j<bn;j++,i++)

{

tmp[i]=bad[j];

}

for(k=0;k<cint;k++,i++)

{

tmp[i]=play\_both[k];

}

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

{

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

if (tmp[i]<tmp[j])

{

int tp=tmp[i];

tmp[i]=tmp[j];

tmp[j]=tp;

}

}

conb=0;

j=0;

while(j<con)

{

if (tmp[j]!=tmp[j+1] && j<con-1)

{

onb[conb]=tmp[j];

conb++;

j++;

}

if(tmp[j]==tmp[j+1])

{

j++;

j++;

}

if(j==con-1 && tmp[j]!=tmp[j-1])

{

onb[conb]=tmp[j];

conb++;

j++;

}

}

}

void set\_M()

{

cout<<"Total Strength of Class:"<<endl<< "M = {"<<M[0];

for(i=1;i<n;i++)

{

cout<<","<<M[i];

}

cout<<"}"<<endl;

}

void set\_Cri()

{

cout<<"Students Playing Cricket :"<<endl<< "A = {"<<cri[0];

for(i=1;i<cn;i++)

{

cout<<","<<cri[i];

}

cout<<"}"<<endl;

}

void set\_Bad()

{

cout<<"Students Playing Badminton :"<<endl<< "B = {"<<bad[0];

for(i=1;i<bn;i++)

{

cout<<","<<bad[i];

}

cout<<"}"<<endl;

}

void set\_both()

{

cout<<"Students Playing both Cricket and Badminton :"<<endl<<"Z = {"<<play\_both[0];

for(i=1;i<cint;i++)

{

cout<<","<<play\_both[i];

}

cout<<"}"<<endl;

}

void set\_any()

{

cout<<"Students Playing either Cricket or Badminton or both :"<<endl<<"Y = {"<<cv[0];

for(i=1;i<cuni;i++)

{

cout<<","<<cv[i];

}

cout<<"}"<<endl;

}

void set\_none()

{

cout<<"Student playing Nothing : "<<endl<<"X = {"<<can[0];

for(i=1;i<cnne;i++)

{

cout<<","<<can[i];

}

cout<<"}"<<endl;

}

void set\_OnC()

{

cout<<"Student playing Only Cricket : "<<endl<<"C = {"<<onc[0];

for(i=1;i<conc;i++)

{

cout<<","<<onc[i];

}

cout<<"}"<<endl;

}

void set\_OnB()

{

cout<<"Student playing Only Badminton : "<<endl<<"D = {"<<onb[0];

for(i=1;i<conb;i++)

{

cout<<","<<onb[i];

}

cout<<"}"<<endl;

}

};

int main() {

int s,y;

set obj1;

obj1.Str\_n();

obj1.Strength();

obj1.cri\_str\_n();

obj1.Cricket();

obj1.bad\_str\_n();

obj1.Badminton();

obj1.Play\_any();

obj1.Play\_both();

obj1.Play\_none();

obj1.OnCri();

obj1.OnBad();

//Sets OUT;

obj1.set\_M(); //M={};

obj1.set\_Cri(); //C={};

obj1.set\_Bad(); //B={};\*/

cout<<endl;

do

{

cout<<"\t Enter the Operation you want to perform :"<<endl;

cout<<"1. Student Playing Only Cricket:"<<endl<<"2. Student Playing Only Badminton:"<<endl<<"3. Students Playing either cricket or badminton (Union):"<<endl<<"4. Student Playing Both cricket and badminton (Intersection):"<<endl<<"5. Student Playing Neither cricket or badminton:"<<endl;

cin>>s;

switch(s)

{

case 3: obj1.set\_any();

break;

case 1: obj1.set\_OnC();

break;

case 2: obj1.set\_OnB();

break;

case 4: obj1.set\_both();

break;

case 5: obj1.set\_none();

break;

default: cout<<"!!!INVALID OPERATION!!!"<<endl<<" Try Again......";

}

cout<<endl<<"Do You Wish To Continue This Program ??"<<endl<<"Enter any Natural number if Yes Else Enter 0 or any -ve Integer"<<endl;

cin>>y;

}while(y>0);

return 0;

}

/\*

output:

Enter the Class Strength :

7

Number of Student playing Cricket :

4

Students playing Cricket :

1

2

3

4

Number of Student playing Badminton :

4

Students playing Badminton :

3

4

5

6

Total Strength of Class:

M = {1,2,3,4,5,6,7}

Students Playing Cricket :

A = {1,2,3,4}

Students Playing Badminton :

B = {3,4,5,6}

Enter the Operation you want to perform :

1. Student Playing Only Cricket:

2. Student Playing Only Badminton:

3. Students Playing Something (Union):

4. Student Playing Both (Intersection):

5. Student Playing Nothing:

1

Student playing Only Cricket :

C = {1,2}

Do You Wish To Continue This Program ??

Enter any Natural number if Yes Else Enter 0 or any -ve Integer

1

Enter the Operation you want to perform :

1. Student Playing Only Cricket:

2. Student Playing Only Badminton:

3. Students Playing Something (Union):

4. Student Playing Both (Intersection):

5. Student Playing Nothing:

2

Student playing Only Badminton :

D = {5,6}

Do You Wish To Continue This Program ??

Enter any Natural number if Yes Else Enter 0 or any -ve Integer

3

Enter the Operation you want to perform :

1. Student Playing Only Cricket:

2. Student Playing Only Badminton:

3. Students Playing Something (Union):

4. Student Playing Both (Intersection):

5. Student Playing Nothing:

3

Students Playing either Cricket or Badminton or both :

Y = {1,2,3,4,5,6}

Do You Wish To Continue This Program ??

Enter any Natural number if Yes Else Enter 0 or any -ve Integer

4

Enter the Operation you want to perform :

1. Student Playing Only Cricket:

2. Student Playing Only Badminton:

3. Students Playing Something (Union):

4. Student Playing Both (Intersection):

5. Student Playing Nothing:

4

Students Playing both Cricket and Badminton :

Z = {3,4}

Do You Wish To Continue This Program ??

Enter any Natural number if Yes Else Enter 0 or any -ve Integer

5

Enter the Operation you want to perform :

1. Student Playing Only Cricket:

2. Student Playing Only Badminton:

3. Students Playing Something (Union):

4. Student Playing Both (Intersection):

5. Student Playing Nothing:

5

Student playing Nothing :

X = {7}

\*/