**A11:sparse matrix transpose, fast transpose and addition**

**#include**<iostream>

**#define** MAX 20

**using** **namespace** std;

**int** **main**()

{

**int** ch, m1,n1,m2,n2,a[10][10],b[10][10],sp1[MAX][3],sp2[MAX][3], sp3[MAX][3],c[10][10],trans[MAX][3],sum=0, count1=0,count2=0, k1=1,k2=1;

**char** ans;

cout<<"enter size of first matrix";

cin>>m1>>n1;

cout<<"enter first matrix"<<**endl**;

**for**(**int** i=0;i<m1;i++)

{

**for**(**int** j=0;j<n1;j++)

{

cin>>a[i][j];

}

}

cout<<"first matrix converted to sparse matrix"<<**endl**;

sp1[0][0]=m1;

sp1[0][1]=n1;

**for**(**int** i=0;i<m1;i++)

{

**for**(**int** j=0;j<n1;j++)

{

**if**(a[i][j]!= 0)

{

sp1[k1][0]=i;

sp1[k1][1]=j;

sp1[k1][2]=a[i][j];

count1+=1;

k1++;

}

}

}

cout<<"total number of non zero elements : "<<count1<<**endl**;

sp1[0][2]=count1;

cout<<"first sparse matrix"<<**endl**;

**for**(**int** i=0;i<=count1;i++)

{

**for**(**int** j=0;j<3;j++)

{

cout<<"\t"<<sp1[i][j];

}

cout<<**endl**;

}

cout<<"enter size of second matrix";

cin>>m2>>n2;

cout<<"enter second matrix"<<**endl**;

**for**(**int** i=0;i<m2;i++)

{

**for**(**int** j=0;j<n2;j++)

{

cin>>b[i][j];

}

}

cout<<"Second matrix converted to sparse matrix"<<**endl**;

sp2[0][0]=m2;

sp2[0][1]=n2;

**for**(**int** i=0;i<m2;i++)

{

**for**(**int** j=0;j<n2;j++)

{

**if**(b[i][j]!= 0)

{

sp2[k2][0]=i;

sp2[k2][1]=j;

sp2[k2][2]=b[i][j];

count2+=1;

k2++;

}

}

}

cout<<"total number of non zero elements : "<<count2<<**endl**;

sp2[0][2]=count2;

cout<<"second sparse matrix"<<**endl**;

**for**(**int** i=0;i<=count2;i++)

{

**for**(**int** j=0;j<3;j++)

{

cout<<"\t"<<sp2[i][j];

}

cout<<**endl**;

}

**do**

{

cout<<"\n 1. Transpose \n 2. fast transpose \n 3. addition of sparse matrix \n enter your choice"<<**endl**;

cin>>ch;

**switch**(ch)

{

**case** 1:

cout<<"\n transpose of first sparse matrix \n";

trans[0][0]=sp1[0][1];

trans[0][1]=sp1[0][0];

trans[0][2]=sp1[0][2];

**if**(sp1[0][2]>1)//if non zero matrix then

{

**int** k=1;

**for**(**int** c=0;c<=sp1[0][1];c++)//for column values

{

**for**(**int** t=1;t<=sp1[0][2];t++)

{

**if**(sp1[t][1]==c)

{

trans[k][0]=sp1[t][1];

trans[k][1]=sp1[t][0];

trans[k][2]=sp1[t][2];

k++;

}

}

}

}

**for**(**int** c=0;c<=trans[0][2];c++)

{

cout<<trans[c][0]<<"\t"<<trans[c][1]<<"\t"<<trans[c][2]<<"\n";

}

**break**;

**case** 2:

**int** rterm[MAX],rpos[MAX],ft[MAX][3],i,j,row,col,num;

row=sp1[0][0];

col=sp1[0][1];

num=sp1[0][2];

ft[0][0]=col;

ft[0][1]=row;

ft[0][2]=num;

**if**(num>0)

{

**for**(i=0;i<=col;i++)

rterm[i]=0;

**for**(i=1;i<=num;i++)

rterm[sp1[i][1]]++;

rpos[0]=1; //setting row wise position

**for**(i=1;i<=col;i++)

rpos[i]=rpos[i-1]+rterm[i-1];

**for**(i=1;i<=num;i++)

{

j=rpos[sp1[i][1]];

ft[j][0]=sp1[i][1];

ft[j][1]=sp1[i][0];

ft[j][2]=sp1[i][2];

rpos[sp1[i][1]]=j+1;

}

}

**for**(**int** c=0;c<=num;c++)

{

cout<<ft[c][0]<<"\t"<<ft[c][1]<<"\t"<<ft[c][2]<<"\n";

}

**break**;

**case** 3:

**int** k=1;

i=1,j=1;

**if**(sp1[0][0]>=sp2[0][0])

sp3[0][0]=sp1[0][0];

**else**

sp3[0][0]=sp2[0][0];

**if**(sp1[0][1]>=sp2[0][1])

sp3[0][1]=sp1[0][1];

**else**

sp3[0][1]=sp2[0][1];

**while**(i<=sp1[0][2] && j<= sp2[0][2] )

{

**if**(sp1[i][0]==sp2[j][0])

{

**if**(sp1[i][1]==sp2[j][1])

{

sp3[k][2]=sp1[i][2]+sp2[j][2];

sp3[k][1]=sp1[i][1];

sp3[k][0]=sp1[i][0];

i++;

j++;

k++;

}

**else** **if**(sp1[i][1]<sp2[j][1])

{

sp3[k][2]=sp1[i][2];

sp3[k][1]=sp1[i][1];

sp3[k][0]=sp1[i][0];

i++;

k++;

}

**else**

{

sp3[k][2]=sp2[i][2];

sp3[k][1]=sp2[i][1];

sp3[k][0]=sp2[i][0];

j++;

k++;

}

}

**else** **if**(sp1[i][0]<sp2[j][0])

{

sp3[k][2]=sp1[i][2];

sp3[k][1]=sp1[i][1];

sp3[k][0]=sp1[i][0];

i++;

k++;

}

**else**

{

sp3[k][2]=sp2[i][2];

sp3[k][1]=sp2[i][1];

sp3[k][0]=sp2[i][0];

j++;

k++;

}

}//end of while

//copingremainingelements

**while**(i<=sp1[0][2])

{

sp3[k][2]=sp1[i][2];

sp3[k][1]=sp1[i][1];

sp3[k][0]=sp1[i][0];

i++;

k++;

}

**while**(j<=sp2[0][2])

{

sp3[k][2]=sp2[j][2];

sp3[k][1]=sp2[j][1];

sp3[k][0]=sp2[j][0];

j++;

k++;

}

sp3[0][2]=k-1;

cout<<"Rowno colno value\n";

**for**(**int** i=0;i<=sp3[0][2];i++)

{

cout<<sp3[i][0]<<"\t"<<sp3[i][1]<<"\t"<<sp3[i][2]<<"\n";

}

**break**;

}

cout<<"\n do you want to continue?(y/n)";

cin>>ans;

}**while**(ans=='y' || ans=='Y');

**return** 0;

}

//output

enter size of first matrix2

2

enter first matrix

10

20

30

0

first matrix converted to sparse matrix

total number of non zero elements : 3

first sparse matrix

2 2 3

0 0 10

0 1 20

1 0 30

enter size of second matrix2

2

enter second matrix

0

0

5

40

Second matrix converted to sparse matrix

total number of non zero elements : 2

second sparse matrix

2 2 2

1 0 5

1 1 40

1. Transpose

2. fast transpose

3. addition of sparse matrix

enter your choice

1

transpose of first sparse matrix

2 2 3

0 0 10

0 1 30

1 0 20

do you want to continue?(y/n)y

1. Transpose

2. fast transpose

3. addition of sparse matrix

enter your choice

2

2 2 3

0 0 10

0 1 30

1 0 20

do you want to continue?(y/n)y

1. Transpose

2. fast transpose

3. addition of sparse matrix

enter your choice

3

Rowno colno value

2 2 4

0 0 10

0 1 20

1 0 35

1 1 40

do you want to continue?(y/n)n