```
#include<iostream>
using namespace std;
#include<stdlib.h>
#define MAX 100

class Sparse
{
	int a[MAX][MAX],b[MAX][MAX],r,c,k=1,nz=0;
	public :
		void read();
		void convert();
		void display();
		void addition(Sparse &S1,Sparse &S2);
		void transdis();
	    void fast_trans();
		void multi(Sparse &S1,Sparse &S2);
};

void Sparse::read()
{
	cout<<"\nEnter number of rows and columns:";
	cin>>r>>c;
	cout<<"\nEnter "<<r*c<<" elements:";
	for (int i=0;i<r;i++)
	{
		for (int j=0;j<c;j++)
		{
			cout<<"("<<i<<", "<<j<<"):";
			cin>>a[i][j];
		}
	}
}

void Sparse::convert()
{
	b[0][0]=r;
	b[0][1]=c;
	for (int i=0;i<r;i++)
	{
		for (int j=0;j<c;j++)
		{
			if (a[i][j]!=0)
			{
				b[k][0]=i;
				b[k][1]=j;
				b[k][2]=a[i][j];
				k++;
				nz++;
			}
		}
```

```
        }
        b[0][2]=nz;
}

void Sparse::display()
{
        cout<<"\nrows\tcol\tvalues\n";
        for (int i=0;i<k;i++)
        {
                for (int j=0;j<3;j++)
                {
                        cout<<b[i][j];
                        cout<<"\t";
                }
                cout<<"\n";
        }
}
void Sparse::addition(Sparse &S1,Sparse &S2)
{
        int sum[100][3];
        int A,B,i=1,j=1,k=1;
        A=S1.b[0][2];
        B=S2.b[0][2];

        sum[0][0]=S1.b[0][0];
        sum[0][1]=S1.b[0][1];

        while (i<=A && j<=B)
        {
                if (S1.b[i][0]==S2.b[j][0])
                {
                        if (S1.b[i][1]==S2.b[j][1])
                        {
                                sum[k][0]=S1.b[i][0];
                                sum[k][1]=S1.b[i][1];
                                sum[k][2]=S1.b[i][2]+S2.b[j][2];
                                i++;
                                j++;
                                k++;
                        }
                        else if (S1.b[i][1]<S2.b[j][1])
                        {
                                sum[k][0]=S1.b[i][0];
                                sum[k][1]=S1.b[i][1];
                                sum[k][2]=S1.b[i][2];
                                i++;
                                k++;
                        }
                        else
                        {
```

```
                        sum[k][0]=S2.b[j][0];
                        sum[k][1]=S2.b[j][1];
                        sum[k][2]=S2.b[j][2];
                        j++;
                        k++;
                }
        }
        else if (S1.b[i][0]>S2.b[j][0])
        {
                sum[k][0]=S2.b[j][0];
                sum[k][1]=S2.b[j][1];
                sum[k][2]=S2.b[j][2];
                j++;
                k++;
        }
        else if (S1.b[i][0]<S2.b[j][0])
        {
                sum[k][0]=S1.b[i][0];
                sum[k][1]=S1.b[i][1];
                sum[k][2]=S1.b[i][2];
                i++;
                k++;
        }
   }
   while (i<=A && B<=j)
   {
        sum[k][0] = S1.b[i][0];
sum[k][1] = S1.b[i][1];
sum[k][2] = S1.b[i][2];
i++;
k++;
   }
   while (i>=A && B>=j)
   {
        sum[k][0] = S2.b[j][0];
sum[k][1] = S2.b[j][1];
sum[k][2] = S2.b[j][2];
j++;
k++;
   }
   sum[0][2]=k-1;
   cout<<"Addition of Matrices is:\n";
   cout<<"\nrows\tcol\tvalues\n";
   for (int i=0;i<k;i++)
   {
        for (int j=0;j<3;j++)
        {
                cout<<sum[i][j];
                cout<<"\t";
        }
```

```
                cout<<"\n";
        }
}

void Sparse::transdis()
{
        int k=1;
        int trans[100][3];
        trans[0][0]=b[0][1];
        trans[0][1]=b[0][0];
        trans[0][2]=b[0][2];
        for (int i=0;i<b[0][1];i++)
        {
                for (int j=1;j<=b[0][2];j++)
                {
                        if (b[j][1]==i)
                        {
                                trans[k][0]=b[j][1];
                                trans[k][1]=b[j][0];
                                trans[k][2]=b[j][2];
                                k++;
                        }
                }
        }
        cout<<"\nrows\tcol\tvalues\n";
        for (int i=0;i<k;i++)
        {
                for (int j=0;j<3;j++)
                {
                        cout<<trans[i][j];
                        cout<<"\t";
                }
                cout<<"\n";
        }
}
void Sparse::fast_trans()
{
int loc,i,col_no;
int result[MAX][3];
int total[b[0][1]], index[b[0][1]+1];

for(i=0;i<b[0][1];i++) {

        total[i]=0;
                        }

for(i=1;i<=b[0][2];i++) {

        col_no = b[i][1];
        total[col_no]++;
```

```
                        }

index[0] = 1;

for(i=1;i<=b[0][1];i++) {

index[i] = index[i-1]+ total[i-1];

                        }

result[0][0] = b[0][1];
result[0][1] = b[0][0];
result[0][2] = b[0][2];

for(i=1;i<=b[0][2];i++) {

        col_no = b[i][1];
        loc = index[col_no];
        result[loc][0] = b[i][1];
        result[loc][1] = b[i][0];
        result[loc][2] = b[i][2];
        index[col_no]++;
        }
cout<<"Fast Transpose of Matrix is:\n";
        cout<<"\nrows\tcol\tvalues\n";

for(int i=0;i<=result[0][2];i++) {

cout<<"\n";
for(j=0;j<3;j++) {

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

}
        }

}


void Sparse::multi(Sparse &S1,Sparse &S2)
{
        int m=1,k=1;
        int tr[100][3];
        tr[0][0]=S2.b[0][1];
        tr[0][1]=S2.b[0][0];
        tr[0][2]=S2.b[0][2];
        for (int i=0;i<S2.b[0][1];i++)
        {
                for (int j=1;j<=S2.b[0][2];j++)
                {
```

```
                if (S2.b[j][1]==i)
                {
                        tr[m][0]=S2.b[j][1];
                        tr[m][1]=S2.b[j][0];
                        tr[m][2]=S2.b[j][2];
                        m++;
                }
        }
}
if (S1.b[0][1]!=tr[0][1])
{
        cout<<"Multiplication not possible\n";
}
else
{
int j=1;
int multi[100][3];
multi[0][0]=S1.b[0][0];
multi[0][1]=tr[0][0];
multi[0][2]=0;

for (int i=1;i<=S1.b[0][2];i++)
{
        for (int j=1;j<=tr[0][2];j++)
        {
                if (S1.b[i][1]==tr[j][1])
                {
                        multi[k][0]=S1.b[i][0];
                        multi[k][1]=tr[j][0];
                        multi[k][2]=S1.b[i][2]*tr[j][2];
                        k++;
           }
        }
}
for (int i=1;i<k-1;i++)
{
        for (int j=i+1;j<=k-1;j++)
        {
                if (multi[i][0]>multi[j][0])
                {
                        int temp1,temp2,temp3;
                        temp1=multi[i][0];
                        multi[i][0]=multi[j][0];
                        multi[j][0]=temp1;
                        temp2=multi[i][1];
                        multi[i][1]=multi[j][1];
                        multi[j][1]=temp2;
                        temp3=multi[i][2];
                        multi[i][2]=multi[j][2];
                        multi[j][2]=temp3;
```

```
                }
                else if (multi[i][0]==multi[j][0])
                {
                        if (multi[i][1]>multi[j][1])
                        {
                                int temp1,temp2,temp3;
                                temp1=multi[i][0];
                                multi[i][0]=multi[j][0];
                                multi[j][0]=temp1;
                                temp2=multi[i][1];
                                multi[i][1]=multi[j][1];
                                multi[j][1]=temp2;
                                temp3=multi[i][2];
                                multi[i][2]=multi[j][2];
                                multi[j][2]=temp3;
                        }
                }
        }
}
int o=0;
for (int i=1;i<k-1;i++)
{
        if (multi[i][0]==multi[i+1][0] && multi[i][1]==multi[i+1][1])
        {
                multi[i][0]=multi[i][0];
                multi[i][1]=multi[i][1];
                multi[i][2]+=multi[i+1][2];
                int p=i+1;
                for (int l=p;l<k-1;l++)
                {
                        multi[p][0]=multi[p+1][0];
                        multi[p][1]=multi[p+1][1];
                        multi[p][2]=multi[p+1][2];
                }
                k--;
                o++;
        }
}
k=k-o;
multi[0][2]=k;
cout<<"\nrows\tcol\tvalues\n";
for (int i=0;i<=k;i++)
{
        for (int j=0;j<3;j++)
        {
                cout<<multi[i][j];
                cout<<"\t";
        }
        cout<<"\n";
}
```

```
        }
}

int main()
{
        Sparse S1;
        cout<<"Elements of First Matrix:"<<endl;
        S1.read();
        S1.convert();
        Sparse S2;
        cout<<"Elements of Second Matrix:"<<endl;
        S2.read();
        S2.convert();
        cout<<"Matrix 1:";
        S1.display();
        cout<<"Matrix 2:";
        S2.display();

        Sparse S3,S4;
        int ch;

        cout<<"\n1.Addition\n2.Transpose\n3.Multiplication\n4.Fast Transpose\n ";
        cout<<"Enter your choice:";
        cin>>ch;
        switch (ch)
        {
                case 1:
                        cout<<"\nAddition of matrix 1 and matrix 2:";
                        S3.addition(S1,S2);
                        break;
                case 2:
                        cout<<"\nTranspose of matrix 1:\n";
                        S1.transdis();
                        cout<<"Transpose of matrix 2:";
                        S2.transdis();
                        break;
                case 3:
                        cout<<"\nMultiplication of matrix 1 and matrix 2:\n";
                        S3.multi(S1,S2);
                        break;
                case 4:
                        cout<<"\nFast Transpose of matrix 1:\n";
                        S1.fast_trans();
                   cout<<"\nFast Transpose of matrix 2:\n";
                        S2.fast_trans();
                        break;
                default:
                        cout<<"Invalid Choice \n";

        }
```

}