**SPARSE MATRIX – PROGRAM**

/\*Jans Johnson

S3 CSE Beta

Rollno: 24\*/

#include<stdio.h>

//Function for creating the triplet matrix

void triplet(int arr[][100],int trip[][3],int l,int m)

{

int i,j;

trip[0][0]=l;

trip[0][1]=m;

int k=1;

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

{

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

{

if(arr[i][j]!=0)

{

trip[k][0]=i;

trip[k][1]=j;

trip[k][2]=arr[i][j];

k++;

}

}

}

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

}

//function for displaying the triplet matrix

void disptriplet(int trip[][3])

{

int j,i=trip[0][2]+1;

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

{

printf("%d %d %d\n",trip[j][0],trip[j][1],trip[j][2]);

}

}

void main()

{

int a[100][100],b[100][100];

int tripa[100][3],tripb[100][3],tripsum[100][3];

int j,i,k;

int l,m;

int resp;

int lim;

do

{

printf("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?? MENU DRIVEN PROGRAM ??\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n1. Sparse matrix Addition\n2. Sparse matrix Transpose\n3.Exit\nYour Choice:");

scanf("%d",&resp);

if(resp==1)

{

//inserting the first matrix

printf("Enter the dimensions of the matrices:");

scanf("%d%d",&l,&m);

printf("Enter the first matrix:\n");

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

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

scanf("%d",&a[i][j]);

//inserting second matrix

printf("Enter the second matrix:\n");

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

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

scanf("%d",&b[i][j]);

triplet(a,tripa,l,m);

triplet(b,tripb,l,m);

//displaying the triplet representation

printf("\n");

disptriplet(tripa);

printf(" +\n");

disptriplet(tripb);

printf(" =\n");

//finding and displaying the sum

tripsum[0][0]=l;

tripsum[0][1]=m;

k=i=j=1;

while(i<=tripa[0][2]&&j<=tripb[0][2])

{

if(tripa[i][0]<tripb[j][0])

//row numbers are not equal

{

tripsum[k][0]=tripa[i][0];

tripsum[k][1]=tripa[i][1];

tripsum[k][2]=tripa[i][2];

k++;i++;

}

else if(tripb[j][0]<tripa[i][0])

//row numbers are not equal

{

tripsum[k][0]=tripb[j][0];

tripsum[k][1]=tripb[j][1];

tripsum[k][2]=tripb[j][2];

k++;j++;

}

else if(tripa[i][1]<tripb[j][1])

//row numbers are equal, compare column

{

tripsum[k][0]=tripa[i][0];

tripsum[k][1]=tripa[i][1];

tripsum[k][2]=tripa[i][2];

k++;i++;

}

else if(tripb[j][1]<tripa[i][1])

//row numbers are equal, compare column

{

tripsum[k][0]=tripb[j][0];

tripsum[k][1]=tripb[j][1];

tripsum[k][2]=tripb[j][2];

k++;j++;

}

else

{

tripsum[k][0]=tripa[i][0];

//row and column numbers are equal

tripsum[k][1]=tripa[i][1];

tripsum[k][2]=tripa[i][2]+tripb[j][2];

k++;

i++;

j++;

}

}

while(i<=tripa[0][2])

//copy remaining terms from tripa

{

tripsum[k][0]=tripa[i][0];

tripsum[k][1]=tripa[i][1];

tripsum[k][2]=tripa[i][2];

i++;

k++;

}

while(j<=tripb[0][2])

//copy remaining terms from tripb

{

tripsum[k][0]=tripb[j][0];

tripsum[k][1]=tripa[j][1];

tripsum[k][2]=tripa[j][2];

j++;

k++;

}

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

disptriplet(tripsum);

}

else if(resp==2)

{

printf("Enter the dimensions of the matrix:");

scanf("%d %d",&l,&m);

printf("Enter the matrix:\n");

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

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

scanf("%d",&a[i][j]);

printf("\nThe triplet representaion is:\n");

triplet(a,tripa,l,m);;

disptriplet(tripa);

printf("\n\nTranspose:\n");

for(i=0;i<=tripa[0][2];i++)

{

tripb[i][0]=tripa[i][1];

tripb[i][1]=tripa[i][0];

tripb[i][2]=tripa[i][2];

}

//Sorting the elements

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

for(j=1;j<tripb[0][2];j++)

{

if(tripb[j][0]>tripb[j+1][0])

{

k=tripb[j][0];

tripb[j][0]=tripb[j+1][0];

tripb[j+1][0]=k;

k=tripb[j][1];

tripb[j][1]=tripb[j+1][1];

tripb[j+1][1]=k;

k=tripb[j][2];

tripb[j][2]=tripb[j+1][2];

tripb[j+1][2]=k;

}

else if(tripb[i][0]==tripb[i+1][0] && tripb[i][1]>tripb[i+1][1])

{

k=tripb[i][0];

tripb[i][0]=tripb[i+1][0];

tripb[i+1][0]=k;

k=tripb[i][1];

tripb[i][1]=tripb[i+1][1];

tripb[i+1][1]=k;

k=tripb[i][2];

tripb[i][2]=tripb[i+1][2];

tripb[i+1][2]=k;

}

}

disptriplet(tripb);

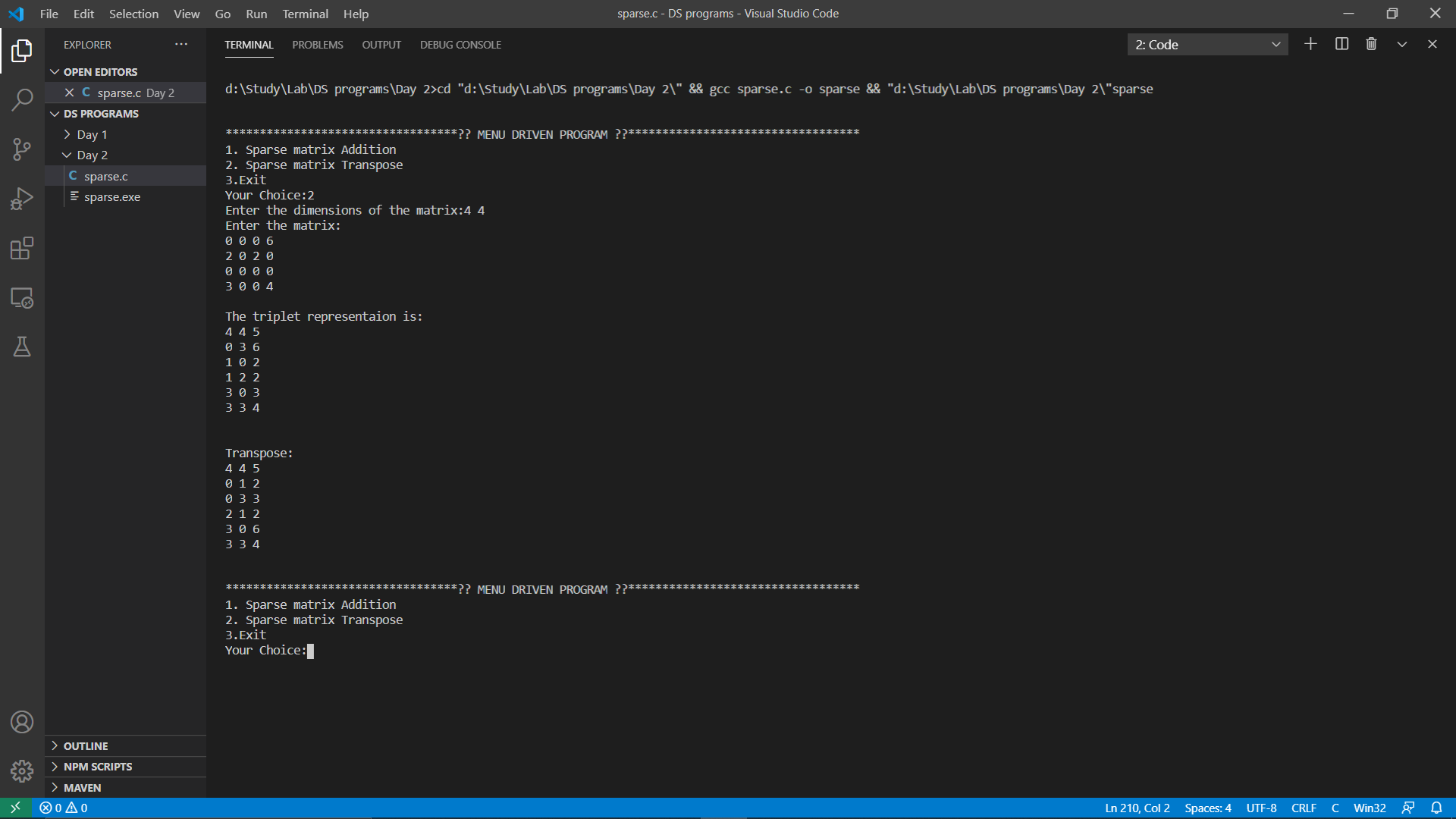
}

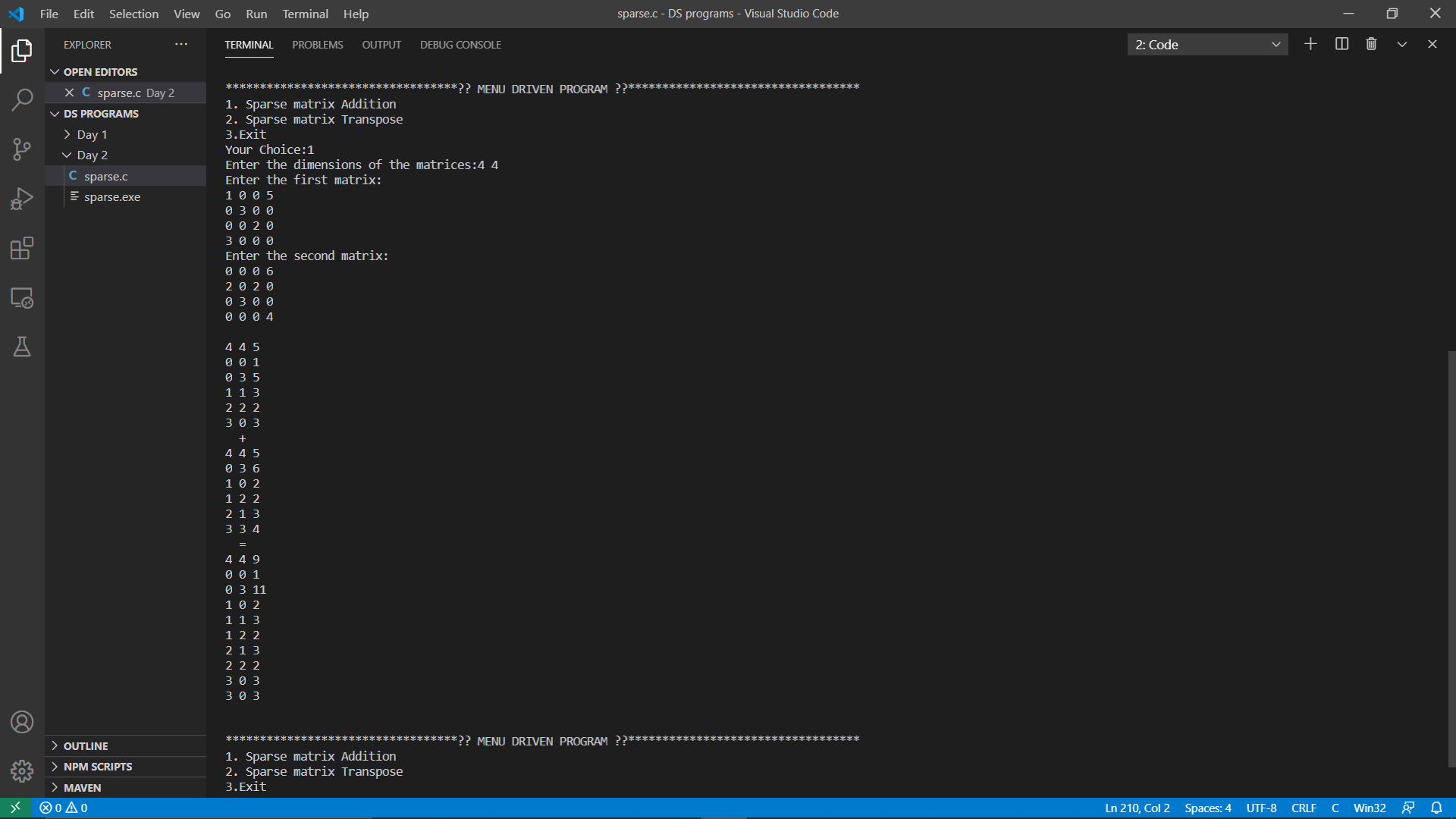
}while(resp==1 || resp==2);

}

**OUTPUT**

**Test case 1:**

****

**Test Case 2:**