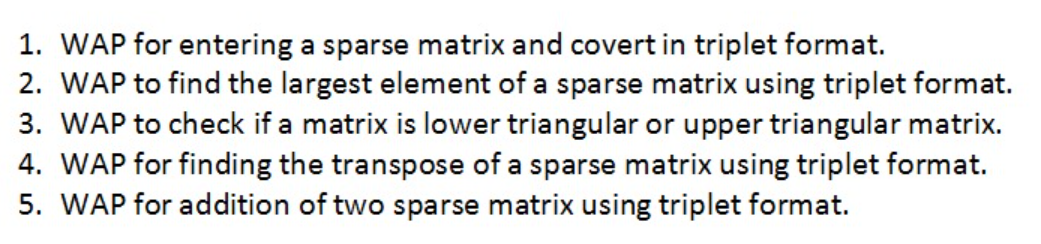
QuestionsAnswers

1.

#include <stdio.h>

int main()

{

int i,j,m,n,count=0,k=1;

printf("For a sparx matrix\n");

printf("Enter the row number and column number\n");

scanf("%d %d",&i,&j);

int a[i][j];

printf("Enter elements\n");

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

for(n=0;n<j;n++){

printf("[%d][%d] : ",m,n);

scanf("%d",&a[m][n]);

if(a[m][n]!=0)

{

count++;

}

}

printf("Showing Matrix\n");

for(m=0;m<i;m++){

for(n=0;n<j;n++){

printf("%d ",a[m][n]);

}

printf("\n");

}

int b[count+1][3];

b[0][0]=i;

b[0][1]=j;

b[0][2]=count;

for(m=0;m<i;m++){

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

{

if(a[m][n]!=0)

{

b[k][0]=m;

b[k][1]=n;

b[k][2]=a[m][n];

k++;

}

}}

printf("The triplet form is \n");

for(m=0;m<count+1;m++){

for(n=0;n<3;n++){

printf("%d ",b[m][n]);

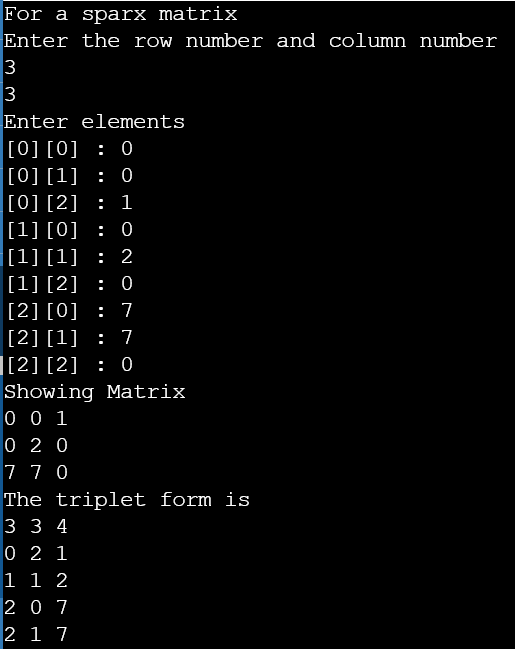
}

printf("\n");

}

return 0;

}

OUTPUT

2.

#include <stdio.h>

int main()

{

int i,j,m,n,count=0,k=1,x;

printf("To find the largest number in a sparx matrix\n");

printf("Enter the row number and column number\n");

scanf("%d %d",&i,&j);

int a[i][j];

printf("Enter elements\n");

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

for(n=0;n<j;n++){

printf("[%d][%d] : ",m,n);

scanf("%d",&a[m][n]);

if(a[m][n]!=0)

{

count++;

}

}

printf("Showing Matrix\n");

for(m=0;m<i;m++){

for(n=0;n<j;n++){

printf("%d ",a[m][n]);

}

printf("\n");

}

int b[count+1][3];

b[0][0]=i;

b[0][1]=j;

b[0][2]=count;

for(m=0;m<i;m++){

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

{

if(a[m][n]!=0)

{

b[k][0]=m;

b[k][1]=n;

b[k][2]=a[m][n];

k++;

}

}}

printf("The triplet form is \n");

for(m=0;m<count+1;m++){

for(n=0;n<3;n++){

printf("%d ",b[m][n]);

}

printf("\n");

}

x=b[1][2];

for(m=1;m<count;m++)

{

if(b[m+1][2]>x){

x=b[m+1][2];

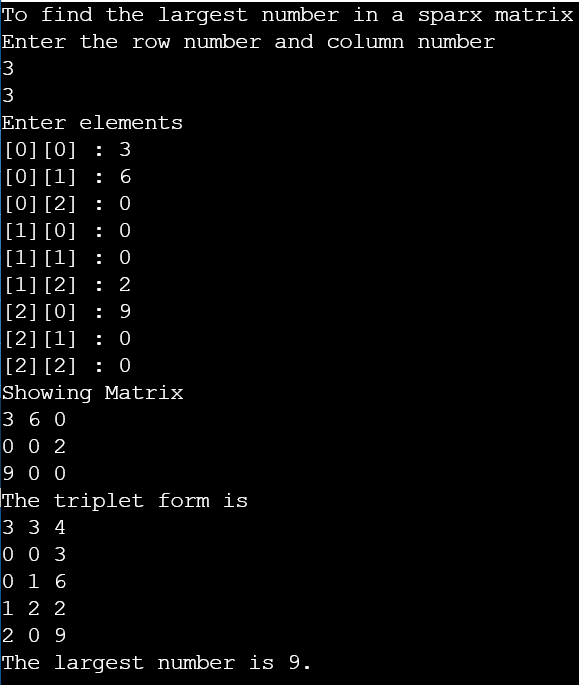
}

}

printf("The largest number is %d.",x);

return 0;

}

OUTPUT

3.

#include <stdio.h>

int main()

{

int n;

printf("To check which triangular matrix it is\nEnter the number of row or column ");

scanf("%d",&n);

int flag = 0;

int mat[n][n];

int i, j;

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

{

for(j = 0; j < n; j++){

printf("a[%d][%d] : ",i,j);

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

}

}

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

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

if (mat[i][j] != 0){

flag = 0;

break;

}

else

flag = 1;

}

if(flag == 0)

break;

}

if(flag==0){

for (i = 0; i < n-1; i++){

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

if (mat[i][j] != 0){

flag = 0;

break;

}

else

flag = 2;

}

if(flag == 0)

break;

}

}

if (flag == 1)

printf("Upper Triangular Matrix.");

else if(flag == 2)

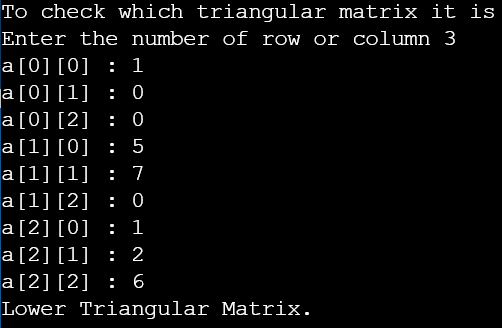
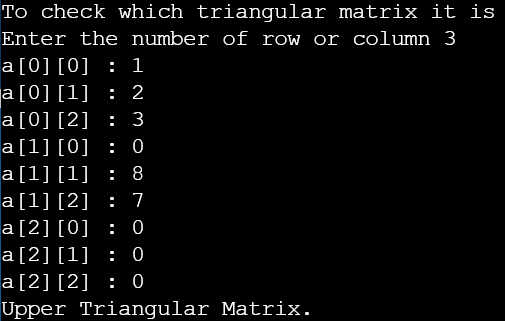
printf("Lower Triangular Matrix.");

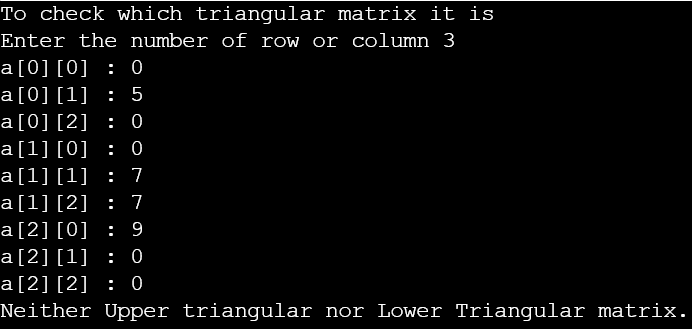
else if(flag == 0)

printf("Neither Upper triangular nor Lower Triangular matrix.");

return 0;

}

OUTPUT



4.

#include <stdio.h>

int main()

{

int i,j,m,n,count=0,k=1;

printf("To find Transpose for a sparx matrix\n");

printf("Enter the row number and column number\n");

scanf("%d %d",&i,&j);

int a[i][j];

printf("Enter elements\n");

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

for(n=0;n<j;n++){

printf("[%d][%d] : ",m,n);

scanf("%d",&a[m][n]);

if(a[m][n]!=0)

{

count++;

}

}

printf("Showing Matrix\n");

for(m=0;m<i;m++){

for(n=0;n<j;n++){

printf("%d ",a[m][n]);

}

printf("\n");

}

int b[count+1][3];

int c[count+1][3];

b[0][0]=i;

b[0][1]=j;

b[0][2]=count;

for(m=0;m<i;m++){

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

{

if(a[m][n]!=0)

{

b[k][0]=m;

b[k][1]=n;

b[k][2]=a[m][n];

k++;

}

}}

printf("The triplet form is \n");

for(m=0;m<count+1;m++){

for(n=0;n<3;n++){

printf("%d ",b[m][n]);

}

printf("\n");

}

for(m=0;m<count+1;m++)

{

c[m][0]=b[m][1];

c[m][1]=b[m][0];

c[m][2]=b[m][2];

}

printf("Printing Transpose of sparx matrix in triplet form is \n");

for(m=0;m<count+1;m++){

for(n=0;n<3;n++){

printf("%d ",c[m][n]);

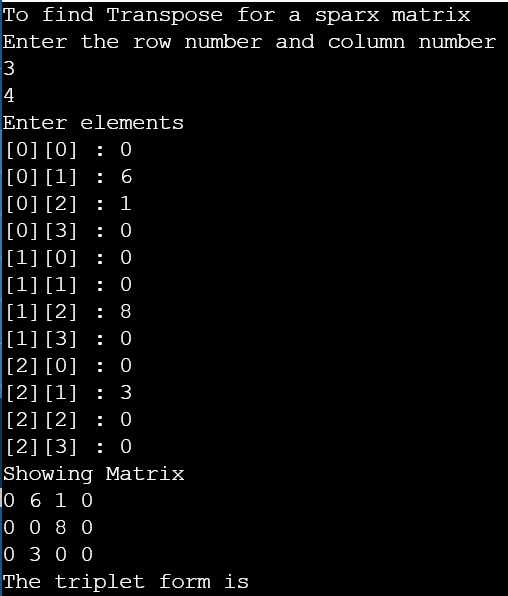
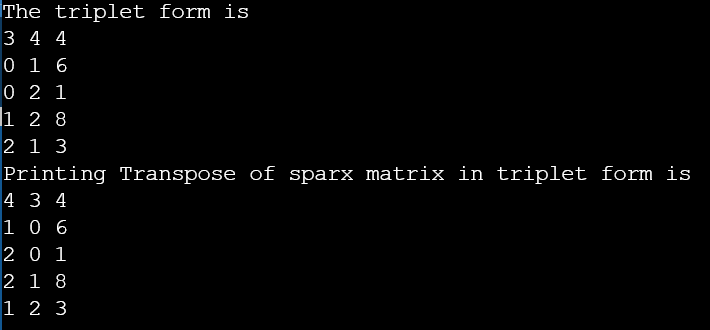
}

printf("\n");

}

return 0;

}

OUTPUT

5.

#include <stdio.h>

int main()

{

int i1,j1,i2,j2,m,n,count1=0,count2=0,k=1;

printf("For 1st sparx matrix\n");

printf("Enter the row number and column number\n");

scanf("%d %d",&i1,&j1);

int a[i1][j1];

printf("For 2nd sparx matrix\n");

printf("Enter the row number and column number\n");

scanf("%d %d",&i2,&j2);

if(i1 != i2 || j1 != j2){

printf("As the row and column no. are not same for both the matrix\nThe addition can't be performed.");

return 0;

}

int b[i2][j2];

printf("Enter elements of 1st matrix \n");

for(m=0;m<i1;m++)

for(n=0;n<j1;n++){

printf("[%d][%d] : ",m,n);

scanf("%d",&a[m][n]);

if(a[m][n]!=0)

{

count1++;

}

}

printf("Enter elements of 2nd matrix \n");

for(m=0;m<i2;m++)

for(n=0;n<j2;n++){

printf("[%d][%d] : ",m,n);

scanf("%d",&b[m][n]);

if(b[m][n]!=0)

{

count2++;

}

}

printf("Showing Matrix 1\n");

for(m=0;m<i1;m++){

for(n=0;n<j1;n++){

printf("%d ",a[m][n]);

}

printf("\n");

}

printf("\nShowing Matrix 2\n");

for(m=0;m<i2;m++){

for(n=0;n<j2;n++){

printf("%d ",b[m][n]);

}

printf("\n");

}

int at[count1+1][3];

at[0][0]=i1;

at[0][1]=j1;

at[0][2]=count1;

int l1=count1;

for(m=0;m<i1;m++){

for(n=0;n<j1;n++)

{

if(a[m][n]!=0)

{

at[k][0]=m;

at[k][1]=n;

at[k][2]=a[m][n];

k++;

}

}}

k=1;

int bt[count2+1][3];

bt[0][0]=i2;

bt[0][1]=j2;

bt[0][2]=count2;

int l2=count2;

for(m=0;m<i2;m++){

for(n=0;n<j2;n++)

{

if(b[m][n]!=0)

{

bt[k][0]=m;

bt[k][1]=n;

bt[k][2]=b[m][n];

k++;

}

}}

printf("For the 1st matrix \n");

printf("The triplet form is \n");

for(m=0;m<=l1;m++){

for(n=0;n<3;n++){

printf("%d ",at[m][n]);

}

printf("\n");

}

printf("For the 2nd matrix \n");

printf("The triplet form is \n");

for(m=0;m<=l2;m++){

for(n=0;n<3;n++){

printf("%d ",bt[m][n]);

}

printf("\n");

}

int temp=count1+count2;

int add[temp][3];

add[0][0]=at[0][0];

add[0][1]=at[0][1];

int x=1,y=1,z=1;

while(x<=l1 && y<=l2){

if(at[x][0]==bt[y][0]){

if(at[x][1]==bt[y][1]){

add[z][0]=at[x][0];

add[z][1]=at[x][1];

add[z][2]=at[x][2]+bt[y][2];

x++;y++;z++;

}

else if(at[x][1]<bt[y][1]){

add[z][0]=at[x][0];

add[z][1]=at[x][1];

add[z][2]=at[x][2];

x++;z++;

}

else

{

add[z][0]=bt[y][0];

add[z][1]=bt[y][1];

add[z][2]=bt[y][2];

y++;z++;

}

}

else if(at[x][0]<bt[y][0]){

add[z][0]=at[x][0];

add[z][1]=at[x][1];

add[z][2]=at[x][2];

x++;z++;

}

else

{

add[z][0]=bt[y][0];

add[z][1]=bt[y][1];

add[z][2]=bt[y][2];

y++;z++;

}

}

while((x<=l1) && (y>l2)){

add[z][0]=at[x][0];

add[z][1]=at[x][1];

add[z][2]=at[x][2];

x++;z++;

}

while((x>l1) && (y<=l2)){

add[z][0]=bt[y][0];

add[z][1]=bt[y][1];

add[z][2]=bt[y][2];

y++;z++;

}

add[0][2]=z-1;

printf("Addition Result \n");

for(m=0;m<z;m++){

for(n=0;n<3;n++){

printf("%d ",add[m][n]);

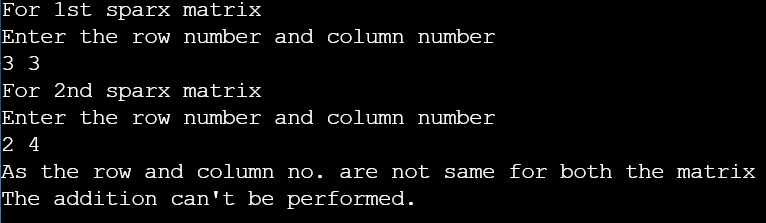
}

printf("\n");

}

return 0;

}

OUTPUT

