6] (a)

#include<stdio.h>

struct Triple{ int row,column,value; };

int main(){

int n;

printf("Enter number of elements: ");

scanf("%d",&n);

struct Triple a[20],t[20];

printf("Enter row,column,value:\n");

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

scanf("%d %d %d",&a[i].row,&a[i].column,&a[i].value);

int k=0;

for(int c=0;c<10;c++)

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

if(a[i].column==c){

t[k].row=a[i].column;

t[k].column=a[i].row;

t[k].value=a[i].value;

k++;

}

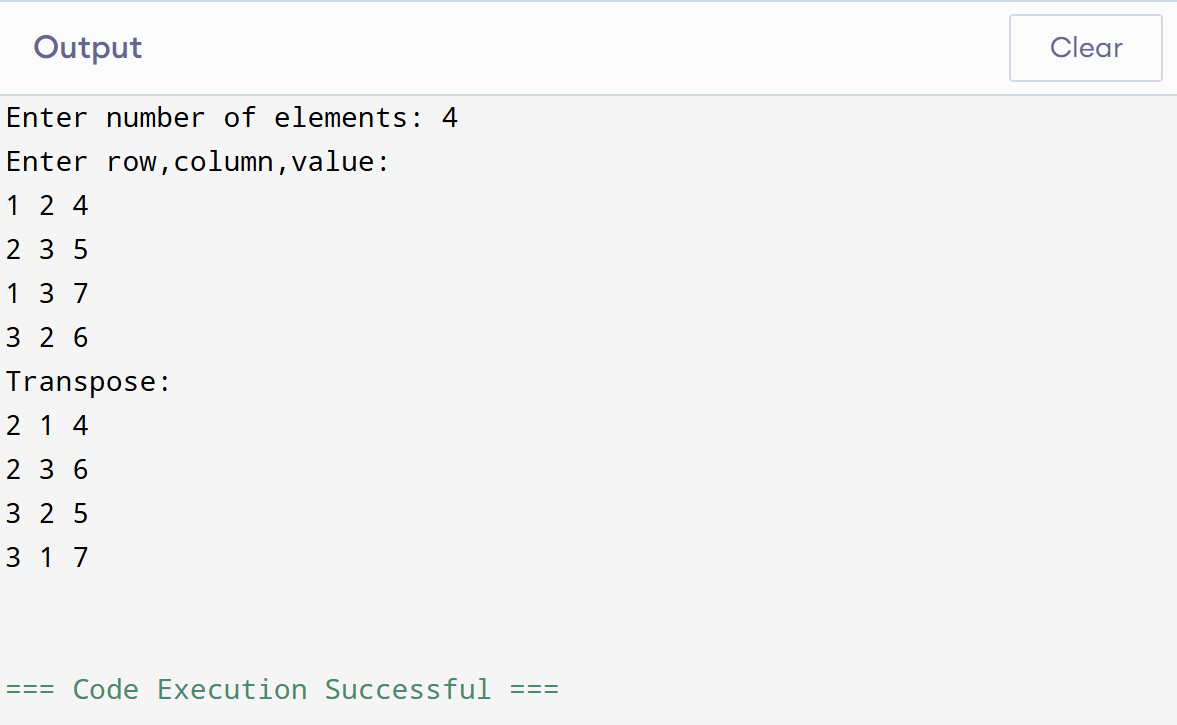
printf("Transpose:\n");

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

printf("%d %d %d\n",t[i].row,t[i].column,t[i].value);

return 0;

}



(b)

#include<stdio.h>

typedef struct{

int row, col, val;

} Triplet;

int main(){

Triplet a[20],b[20],sum[40];

int r,c,mat[10][10],count = 0;

printf("Enter rows and columns for Matrix A & B: ");

scanf("%d%d",&r,&c);

printf("Enter elements of Matrix A:\n");

for(int i=0;i<r;i++){

for(int j=0;j<c;j++){

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

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

a[count+1].row=i;

a[count+1].col=j;

a[count+1].val=mat[i][j];

count++;

}

}

}

a[0].row=r;a[0].col=c;a[0].val=count;

count=0;

printf("Enter elements of Matrix B:\n");

for(int i=0;i<r;i++){

for(int j =0;j<c;j++){

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

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

b[count+1].row=i;

b[count+1].col=j;

b[count+1].val=mat[i][j];

count++;

}

}

}

b[0].row=r;b[0].col=c;b[0].val=count;

int i=1,j=1,k=1;

sum[0].row=r;sum[0].col=c;

while(i<=a[0].val&&j<=b[0].val){

if(a[i].row<b[j].row || (a[i].row==b[j].row&&a[i].col< b[j].col))

sum[k++]=a[i++];

else if(b[j].row<a[i].row || (b[j].row==a[i].row&&b[j].col<a[i].col))

sum[k++]=b[j++];

else{

sum[k]=a[i];

sum[k].val=a[i].val+b[j].val;

k++;i++;j++;

}

}

while(i<=a[0].val) sum[k++] = a[i++];

while(j<=b[0].val) sum[k++] = b[j++];

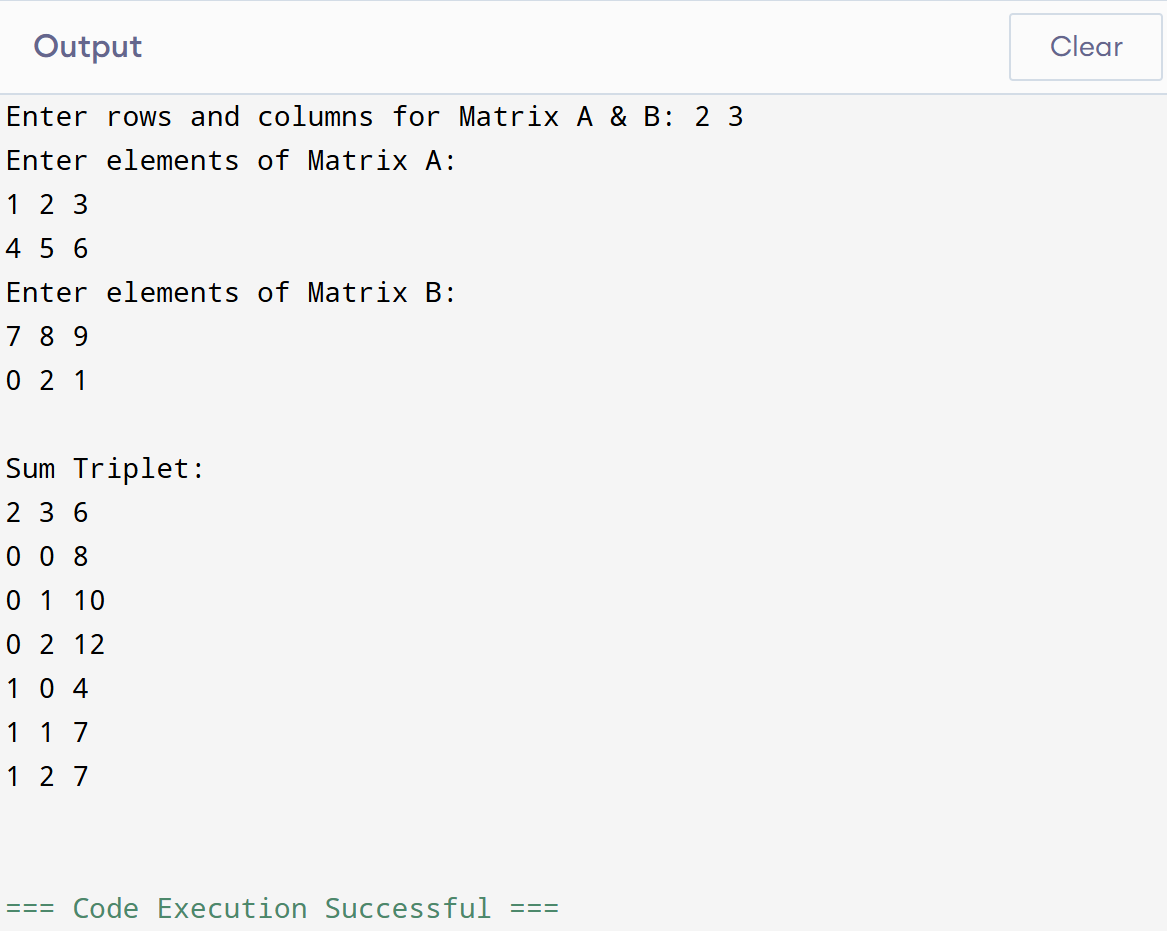
sum[0].val = k-1;

printf("\nSum Triplet:\n");

for (int x=0;x<=sum[0].val;x++)

printf("%d %d %d\n",sum[x].row,sum[x].col,sum[x].val);

return 0;}



(c)

#include<stdio.h>

typedef struct{

int row,col,val;

} Triplet;

void transpose(Triplet a[],Triplet b[]){

int k=1;

b[0].row = a[0].col;

b[0].col = a[0].row;

b[0].val = a[0].val;

for(int col=0;col<a[0].col;col++){

for(int i=1;i<=a[0].val;i++){

if(a[i].col == col){

b[k].row = a[i].col;

b[k].col = a[i].row;

b[k].val = a[i].val;

k++;

}

}

}

}

int main(){

Triplet a[20],b[20],bt[20],prod[40];

int r,c,mat[10][10],count=0;

printf("Enter rows and columns for Matrix A: ");

scanf("%d%d",&r,&c);

printf("Enter elements of Matrix A:\n");

for(int i=0;i<r;i++){

for(int j=0;j<c;j++){

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

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

a[count + 1].row = i;

a[count + 1].col = j;

a[count + 1].val = mat[i][j];

count++;

}

}

}

a[0].row=r;a[0].col=c;a[0].val=count;

count=0;

int r2,c2;

printf("Enter rows and columns for Matrix B: ");

scanf("%d%d",&r2,&c2);

printf("Enter elements of Matrix B:\n");

for(int i=0;i<r2;i++){

for(int j=0;j<c2;j++){

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

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

b[count + 1].row = i;

b[count + 1].col = j;

b[count + 1].val = mat[i][j];

count++;

}

}

}

b[0].row=r2;b[0].col=c2;b[0].val=count;

if(a[0].col != b[0].row){

printf("Multiplication not possible!\n");

return 0;

}

transpose(b,bt);

int k=1;

prod[0].row = a[0].row;

prod[0].col = b[0].col;

for(int i=0;i<a[0].row;i++){

for(int j=0;j<b[0].col;j++){

int sum=0;

for(int p=1;p<=a[0].val;p++){

if(a[p].row==i){

for(int q=1;q<=bt[0].val;q++){

if(bt[q].row == j && bt[q].col == a[p].col){

sum += a[p].val \* bt[q].val;

}

}

}

}

if(sum != 0){

prod[k].row = i;

prod[k].col = j;

prod[k].val = sum;

k++;

}

}

}

prod[0].val=k-1;

printf("\nProduct Triplet:\n");

for(int x=0;x<=prod[0].val;x++)

printf("%d %d %d\n",prod[x].row,prod[x].col,prod[x].val);

return 0;

}

