**ASSIGNMENT 2 – 2D ARRAY**

ARGHA MALLICK – 11500122014

Q7. Write a C program to store the elements in a 2D array and display it & represent it in row major order.

Source Code:

#include<stdio.h>

int main(){

int m, n;

printf("Enter number of rows: ");

scanf("%d",&m);

printf("Enter number of cols: ");

scanf("%d", &n);

int arr[m][n];

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

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

printf("Enter number for arr[%d][%d]: ",i,j);

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

}

}

printf("Your 2D matrix as follows\n");

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

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

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

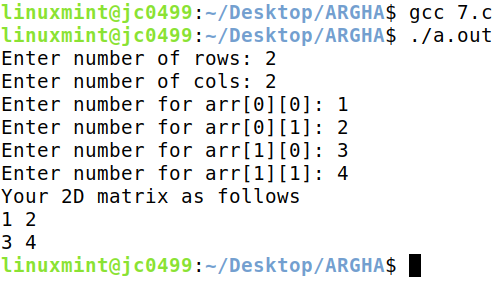
}

printf("\n");

}

}

Output:



Q8. Write a program to test a given matrix is parse or not. If it is parse then represent it in 3-tuple format.

Source Code:

#include<stdio.h>

int main(){

int m, n;

printf("Enter number of rows: ");

scanf("%d",&m);

printf("Enter number of cols: ");

scanf("%d", &n);

int arr[m][n];

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

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

printf("Enter number for arr[%d][%d]: ",i,j);

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

}

}

int isSparse = 0, nonZeroCount = 0, zeroCount = 0;

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

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

if(arr[i][j] != 0) nonZeroCount++;

else zeroCount++;

}

}

if(zeroCount > nonZeroCount) {

isSparse = 1;

}

if(isSparse == 1) {

// represent it as 3 - tuple format

int sparse[3][nonZeroCount], k = 0;

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

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

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

sparse[0][k] = i;

sparse[1][k] = j;

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

k++;

}

}

}

for(int i=0; i<3; i++) {

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

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

}

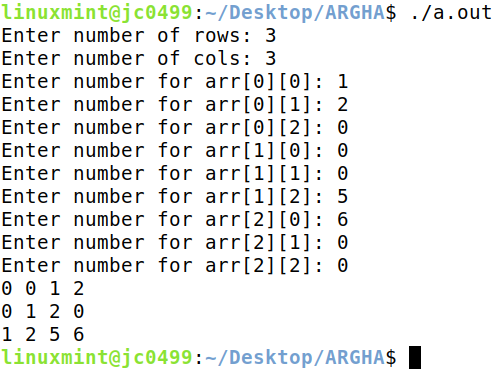
printf("\n");

}

}

}

Output:



Q9. Write a C program to find the Tranpose the given matrix.

Source Code:

#include <stdio.h>

int main() {

int m, n;

printf("Enter number of rows: ");

scanf("%d", &m);

printf("Enter number of cols: ");

scanf("%d", &n);

int arr[m][n];

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

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

printf("Enter number for arr[%d][%d]: ", i, j);

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

}

}

int isSparse = 0, nonZeroCount = 0, zeroCount = 0;

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

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

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

nonZeroCount++;

else

zeroCount++;

}

}

if (zeroCount > nonZeroCount) {

isSparse = 1;

}

if (isSparse == 1) {

// convert to 3 - tuple format

int sparse[3][nonZeroCount], k = 0;

int transposeSparse[nonZeroCount][3];

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

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

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

sparse[0][k] = i;

sparse[1][k] = j;

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

k++;

}

}

}

// print the sparse matrix

printf("The Sparse Matrix is as follows\n");

for (int i = 0; i < 3; i++) {

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

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

}

printf("\n");

}

// find the transpose of the sparse matrix

for (int i = 0; i < 3; i++) {

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

transposeSparse[j][i] = sparse[i][j];

}

}

// print the transpose of the sparse matrix

printf("The Transpose Sparse Matrix is as follows\n");

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

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

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

}

printf("\n");

}

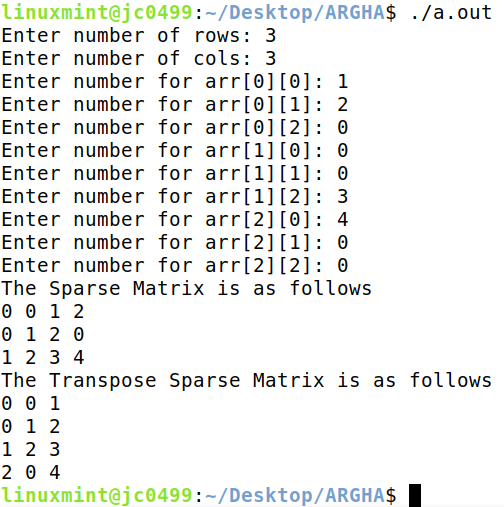
} else {

printf("This is not a Sparse Matrix.\n");

}

}

Output:



Q10. Write a C program to find the matrix multiplication of two given matrix.

Source Code:

#include <stdio.h>

int main() {

// input first matrix

int m1, n1;

printf("Enter number of rows: ");

scanf("%d", &m1);

printf("Enter number of cols: ");

scanf("%d", &n1);

int arr1[m1][n1];

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

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

printf("Enter number for arr1[%d][%d]: ", i, j);

scanf("%d", &arr1[i][j]);

}

}

// input second matrix

int m2, n2;

printf("Enter number of rows: ");

scanf("%d", &m2);

printf("Enter number of cols: ");

scanf("%d", &n2);

int arr2[m2][n2];

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

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

printf("Enter number for arr1[%d][%d]: ", i, j);

scanf("%d", &arr2[i][j]);

}

}

if (n2 == m1) {

// do multiplication

int mulArr[m1][n2];

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

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

mulArr[i][j] = 0;

for (int k = 0; k < m1; k++) {

mulArr[i][j] += arr1[i][k] \* arr2[k][j];

}

}

}

// print the matrix

printf("The multiplication is as follows\n");

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

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

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

}

printf("\n");

}

} else {

printf("Multiplication Not Possible.\n");

}

}

Output:

