**Assignment - 16 A Job Ready Bootcamp in C++, DSA and IOT MySirG**

**Multi-Dimensional Array in C Language**

1. Write a program to calculate the sum of two matrices each of order 3x3.

#include<stdio.h>

int main()

{

int A[3][3], B[3][3], C[3][3], i, j;

printf("Enter nine numbers for first Matrix: \n");

for(i=0; i<=2; i++)

{

for(j=0; j<=2; j++)

{

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

}

}

printf("Enter nine numbers for second Matrix: \n");

for(i=0; i<=2; i++)

{

for(j=0; j<=2; j++)

{

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

}

}

printf("Sum of Matrix is \n");

for(i=0; i<=2; i++)

{

for(j=0; j<=2; j++)

{

C[i][j]=A[i][j]+B[i][j];

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

}

printf("\n");

}

return 0;

}

2. Write a program to calculate the product of two matrices each of order 3x3.

#include<stdio.h>

int main()

{

int A[3][3], B[3][3], C[3][3], i, j;

printf("Enter nine numbers for first Matrix: \n");

for(i=0; i<=2; i++)

{

for(j=0; j<=2; j++)

{

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

}

}

printf("Enter nine numbers for second Matrix: \n");

for(i=0; i<=2; i++)

{

for(j=0; j<=2; j++)

{

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

}

}

printf("Product of Matrix is \n");

for(i=0; i<=2; i++)

{

for(j=0; j<=2; j++)

{

C[i][j]=A[i][j] \* B[i][j];

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

}

printf("\n");

}

return 0;

}

3. Write a program in C to find the transpose of a given matrix.

#include <stdio.h>

int main()

{

int matrix[10][10];

int transpose[10][10];

int i, j, rows, columns;

printf("Enter the number of rows and columns of the matrix: ");

scanf("%d %d", &rows, &columns);

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

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

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

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

}

}

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

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

transpose[j][i] = matrix[i][j];

}

}

printf("\nOriginal matrix:\n");

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

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

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

}

printf("\n");

}

printf("\nTranspose of the matrix:\n");

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

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

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

}

printf("\n");

}

return 0;

}

4. Write a program in C to find the sum of right diagonals of a matrix.

#include <stdio.h>

int main()

{

int matrix[10][10];

int i, j, size, sum = 0;

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

scanf("%d", &size);

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

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

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

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

}

}

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

sum += matrix[i][size-i-1];

}

printf("\nMatrix:\n");

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

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

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

}

printf("\n");

}

printf("\nSum of right diagonals: %d\n", sum);

return 0;

}

5. Write a program in C to find the sum of left diagonals of a matrix.

#include <stdio.h>

int main()

{

int matrix[10][10];

int i, j, size, sum = 0;

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

scanf("%d", &size);

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

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

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

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

}

}

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

sum += matrix[i][i];

}

printf("\nMatrix:\n");

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

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

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

}

printf("\n");

}

printf("\nSum of left diagonals: %d\n", sum);

return 0;

}

6. Write a program in C to find the sum of rows and columns of a Matrix.

#include <stdio.h>

int main() {

int matrix[10][10], row\_sum, col\_sum;

int rows, cols;

printf("Enter the number of rows and columns in the matrix: ");

scanf("%d %d", &rows, &cols);

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

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

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

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

}

}

printf("The matrix is:\n");

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

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

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

}

printf("\n");

}

printf("The sum of each row is:\n");

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

row\_sum=0;

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

row\_sum =row\_sum + matrix[i][j];

}

printf("%d ",row\_sum);

}

printf("\n");

printf("The sum of each column is:\n");

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

col\_sum=0;

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

col\_sum = col\_sum + matrix[j][i];

}

printf("%d ",col\_sum);

}

printf("\n");

return 0;

}

7. Write a program in C to print or display the lower triangular of a given matrix.

#include <stdio.h>

int main() {

int matrix[10][10];

int rows, cols;

printf("Enter the number of rows and columns in the matrix: ");

scanf("%d %d", &rows, &cols);

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

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

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

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

}

}

printf("The matrix is:\n");

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

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

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

}

printf("\n");

}

printf("Lower triangle is:\n");

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

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

if(i>j){

matrix[i][j]=0;

}

else{

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

}

}

printf("\n");

}

return 0;

}

8. Write a program in C to print or display an upper triangular matrix.

#include <stdio.h>

int main() {

int matrix[10][10];

int rows, cols;

printf("Enter the number of rows and columns in the matrix: ");

scanf("%d %d", &rows, &cols);

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

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

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

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

}

}

printf("The matrix is:\n");

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

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

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

}

printf("\n");

}

printf("Upper triangle is:\n");

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

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

if(i<j){

matrix[i][j]=0;

}

else{

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

}

}

printf("\n");

}

return 0;

}

9. Write a program in C to accept a matrix and determine whether it is a sparse matrix.

#include <stdio.h>

int main() {

int matrix[10][10];

int num\_rows, num\_cols, num\_zero = 0;

printf("Enter the number of rows and columns in the matrix: ");

scanf("%d %d", &num\_rows, &num\_cols);

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

for(int i = 0; i < num\_rows; i++) {

for(int j = 0; j < num\_cols; j++) {

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

}

}

printf("The matrix is:\n");

for(int i = 0; i < num\_rows; i++) {

for(int j = 0; j < num\_cols; j++) {

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

}

printf("\n");

}

for(int i = 0; i < num\_rows; i++) {

for(int j = 0; j < num\_cols; j++) {

if(matrix[i][j] == 0) {

num\_zero++;

}

}

}

if(num\_zero > (num\_rows \* num\_cols) / 2) {

printf("The matrix is sparse.\n");

} else {

printf("The matrix is not sparse.\n");

}

return 0;

}

10. Write a program in C to find the row with maximum number of 1s.

#include <stdio.h>

int main() {

int arr[10][10];

int rows, cols, i, j, max\_row, max\_count = 0;

printf("Enter the number of rows and columns: ");

scanf("%d %d", &rows, &cols);

printf("Enter the elements of the array:\n");

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

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

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

}

}

printf("The matrix is:\n");

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

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

printf("%d ", arr

[i][j]);

}

printf("\n");

}

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

int count = 0;

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

if(arr[i][j] == 1) {

count++;

}

}

if(count > max\_count) {

max\_count = count;

max\_row = i;

}

}

printf("Row with maximum number of 1s is %d", max\_row);

return 0;

}