

DAY-2 TASK(Team-D)

1. Given two arrays of integers, find the common elements between them:

main.c	Output
<pre>1 #include <stdio.h> 2 3 void findCommonElements(int arr1[], int size1, int arr2[], int size2 4) { 5 printf("Common elements are: "); 6 7 // Loop through each element of the first array 8 for (int i = 0; i < size1; i++) { 9 // Loop through each element of the second array 10 for (int j = 0; j < size2; j++) { 11 // Check if arr1[i] is present in arr2 12 if (arr1[i] == arr2[j]) { 13 printf("%d ", arr1[i]); 14 break; // Break once the common element is found 15 } 16 } 17 } 18 printf("\n"); 19 20 int main() { 21 int arr1[] = {1, 2, 3, 4, 5}; 22 int arr2[] = {3, 4, 5, 6, 7}; 23 24 int size1 = sizeof(arr1) / sizeof(arr1[0]); 25 int size2 = sizeof(arr2) / sizeof(arr2[0]);</pre>	<pre>Common elements are: 3 4 5 === Code Execution Successful ===</pre>

2. Write a program to add two 3x3 matrices using a two-dimensional array:

main.c	Output
<pre>1 #include <stdio.h> 2 3 int main() { 4 // Declare two 3x3 matrices 5 int matrix1[3][3], matrix2[3][3], result[3][3]; 6 7 // Input values for the first matrix 8 printf("Enter elements of the first matrix (3x3):\n"); 9 for (int i = 0; i < 3; i++) { 10 for (int j = 0; j < 3; j++) { 11 printf("Enter element [%d][%d]: ", i + 1, j + 1); 12 scanf("%d", &matrix1[i][j]); 13 } 14 } 15 16 // Input values for the second matrix 17 printf("\nEnter elements of the second matrix (3x3):\n"); 18 for (int i = 0; i < 3; i++) { 19 for (int j = 0; j < 3; j++) { 20 printf("Enter element [%d][%d]: ", i + 1, j + 1); 21 scanf("%d", &matrix2[i][j]); 22 } 23 } 24 25 // Add the matrices 26 for (int i = 0; i < 3; i++) {</pre>	<pre>Enter elements of the first matrix (3x3): Enter element [1][1]: 1 Enter element [1][2]: 2 Enter element [1][3]: 3 Enter element [2][1]: 4 Enter element [2][2]: 5 Enter element [2][3]: 6 Enter element [3][1]: 7 Enter element [3][2]: 8 Enter element [3][3]: 9 Enter elements of the second matrix (3x3): Enter element [1][1]: 9 Enter element [1][2]: 8 Enter element [1][3]: 7 Enter element [2][1]: 6 Enter element [2][2]: 5 Enter element [2][3]: 4 Enter element [3][1]: 3 Enter element [3][2]: 2 Enter element [3][3]: 1 Sum of the matrices: 10 10 10 10 10 10 10 10 10</pre>

3. Write a program to compute the transpose of a 4x4 matrix using two-dimensional arrays:

main.c	Output
<pre>1 #include <stdio.h> 2 3 int main() { 4 int matrix[4][4], transpose[4][4]; 5 6 // Input values for the 4x4 matrix 7 printf("Enter elements of the 4x4 matrix:\n"); 8 for (int i = 0; i < 4; i++) { 9 for (int j = 0; j < 4; j++) { 10 printf("Enter element [%d][%d]: ", i + 1, j + 1); 11 scanf("%d", &matrix[i][j]); 12 } 13 } 14 15 // Compute the transpose of the matrix 16 for (int i = 0; i < 4; i++) { 17 for (int j = 0; j < 4; j++) { 18 transpose[j][i] = matrix[i][j]; 19 } 20 } 21 22 // Display the original matrix 23 printf("\nOriginal Matrix:\n"); 24 for (int i = 0; i < 4; i++) { 25 for (int j = 0; j < 4; j++) { 26 printf("%d ", matrix[i][j]);</pre>	<pre>Enter elements of the 4x4 matrix: Enter element [1][1]: 1 Enter element [1][2]: 2 Enter element [1][3]: 3 Enter element [1][4]: 4 Enter element [2][1]: 5 Enter element [2][2]: 6 Enter element [2][3]: 7 Enter element [2][4]: 8 Enter element [3][1]: 9 Enter element [3][2]: 10 Enter element [3][3]: 11 Enter element [3][4]: 12 Enter element [4][1]: 13 Enter element [4][2]: 14 Enter element [4][3]: 15 Enter element [4][4]: 16 Original Matrix: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 Transposed Matrix: 1 5 9 13</pre>

4. Write a program to compute and display the sum of rows and columns of a 3x3 matrix:

main.c	Output
<pre>1 #include <stdio.h> 2 3 int main() { 4 int matrix[3][3]; 5 int rowSum[3] = {0}; // Array to store sum of rows 6 int colSum[3] = {0}; // Array to store sum of columns 7 8 // Input values for the 3x3 matrix 9 printf("Enter elements of the 3x3 matrix:\n"); 10 for (int i = 0; i < 3; i++) { 11 for (int j = 0; j < 3; j++) { 12 printf("Enter element [%d][%d]: ", i + 1, j + 1); 13 scanf("%d", &matrix[i][j]); 14 } 15 } 16 17 // Calculate sum of rows and columns 18 for (int i = 0; i < 3; i++) { 19 for (int j = 0; j < 3; j++) { 20 rowSum[i] += matrix[i][j]; // Sum for rows 21 colSum[j] += matrix[i][j]; // Sum for columns 22 } 23 } 24 25 // Display the matrix 26 printf("\nMatrix:\n");</pre>	<pre>Enter elements of the 3x3 matrix: Enter element [1][1]: 1 Enter element [1][2]: 2 Enter element [1][3]: 3 Enter element [2][1]: 4 Enter element [2][2]: 5 Enter element [2][3]: 6 Enter element [3][1]: 7 Enter element [3][2]: 8 Enter element [3][3]: 9 Matrix: 1 2 3 4 5 6 7 8 9 Sum of rows: Sum of row 1: 6 Sum of row 2: 15 Sum of row 3: 24 Sum of columns: Sum of column 1: 12 Sum of column 2: 15 Sum of column 3: 18</pre>

5. Write a program to determine if a given 3x3 matrix is a symmetric matrix:

main.c	Output
<pre>1 #include <stdio.h> 2 3 int main() { 4 int matrix[3][3]; 5 int isSymmetric = 1; // Assume the matrix is symmetric 6 // initially 7 8 // Input values for the 3x3 matrix 9 printf("Enter elements of the 3x3 matrix:\n"); 10 for (int i = 0; i < 3; i++) { 11 for (int j = 0; j < 3; j++) { 12 printf("Enter element [%d][%d]: ", i + 1, j + 1); 13 scanf("%d", &matrix[i][j]); 14 } 15 } 16 17 // Check if the matrix is symmetric 18 for (int i = 0; i < 3; i++) { 19 for (int j = i + 1; j < 3; j++) { // We only need to check 20 // the upper triangle (i < j) 21 if (matrix[i][j] != matrix[j][i]) { 22 isSymmetric = 0; // If any pair is not equal, the 23 // matrix is not symmetric 24 break; 25 } 26 } 27 } 28}</pre>	<pre>Enter elements of the 3x3 matrix: Enter element [1][1]: 1 Enter element [1][2]: 2 Enter element [1][3]: 3 Enter element [2][1]: 4 Enter element [2][2]: 5 Enter element [2][3]: 6 Enter element [3][1]: 7 Enter element [3][2]: 8 Enter element [3][3]: 9 The matrix is not symmetric. === Code Execution Successful ===</pre>

6. Write a program to use a three-dimensional array to store and display the marks of 3 students in 4 subjects for 2 classes:

main.c	Output
<pre>1 #include <stdio.h> 2 3 int main() { 4 // Declare a 3D array to store marks for 2 classes, 3 students, 5 // and 4 subjects 6 int marks[2][3][4]; 7 8 // Input marks for 2 classes, 3 students, and 4 subjects 9 printf("Enter marks for 3 students in 4 subjects for 2 classes 10 :\n"); 11 for (int i = 0; i < 2; i++) { // For each class 12 for (int j = 0; j < 3; j++) { // For each student 13 printf("Class %d, Student %d:\n", i + 1, j + 1); 14 for (int k = 0; k < 4; k++) { // For each subject 15 printf("Enter marks for Subject %d: ", k + 1); 16 scanf("%d", &marks[i][j][k]); 17 } 18 } 19 } 20 21 // Display the marks 22 printf("\nDisplaying marks for 2 classes:\n"); 23 for (int i = 0; i < 2; i++) { // For each class 24 printf("\nClass %d:\n", i + 1); 25 for (int j = 0; j < 3; j++) { // For each student 26 printf("Student %d: ", j + 1); 27 for (int k = 0; k < 4; k++) { // For each subject 28 printf("%d ", marks[i][j][k]); 29 } 30 printf("\n"); 31 } 32 } 33}</pre>	<pre>Enter marks for Subject 4: 92 Class 2, Student 2: Enter marks for Subject 1: 86 Enter marks for Subject 2: 85 Enter marks for Subject 3: 87 Enter marks for Subject 4: 90 Class 2, Student 3: Enter marks for Subject 1: 95 Enter marks for Subject 2: 86 Enter marks for Subject 3: 92 Enter marks for Subject 4: 98 Displaying marks for 2 classes: Class 1: Student 1: Subject 1: 85 Subject 2: 90 Subject 3: 88 Subject 4: 92 Student 2: Subject 1: 78 Subject 2: 82 Subject 3: 80 Subject 4: 85 Student 3: Subject 1: 92 Subject 2: 95 Subject 3: 90 Subject 4: 97 Class 2: Student 1: Subject 1: 75 Subject 2: 80 Subject 3: 88 Subject 4: 92 Student 2: Subject 1: 86 Subject 2: 85 Subject 3: 87 Subject 4: 90 Student 3: Subject 1: 95 Subject 2: 86 Subject 3: 92 Subject 4: 98 === Code Execution Successful ===</pre>

7. Write a program to check whether a matrix is sparse and display its non-zero elements:

main.c	Output
<pre>1 #include <stdio.h> 2 3 int main() { 4 int rows, cols; 5 6 // Input the dimensions of the matrix 7 printf("Enter number of rows and columns: "); 8 scanf("%d %d", &rows, &cols); 9 10 int matrix[rows][cols]; 11 int totalElements = rows * cols; 12 int zeroCount = 0; 13 14 // Input matrix elements 15 printf("Enter elements of the matrix:\n"); 16 for (int i = 0; i < rows; i++) { 17 for (int j = 0; j < cols; j++) { 18 scanf("%d", &matrix[i][j]); 19 if (matrix[i][j] == 0) { 20 zeroCount++; 21 } 22 } 23 } 24 }</pre>	<pre>Enter number of rows and columns: 3 3 Enter elements of the matrix: 0 0 0 5 0 0 0 0 3 The matrix is sparse. Non-zero elements in the matrix are: Element at position [1][0]: 5 Element at position [2][2]: 3 === Code Execution Successful ===</pre>

8. Write a program to rotate a 4x4 matrix 90 degrees in the clockwise direction:

main.c	Output
<pre>1 #include <stdio.h> 2 3 #define SIZE 4 // Define matrix size as 4x4 4 5 void rotateMatrix(int matrix[SIZE][SIZE]) { 6 // Temporary matrix to store the rotated matrix 7 int temp[SIZE][SIZE]; 8 9 // Rotate the matrix by 90 degrees clockwise 10 for (int i = 0; i < SIZE; i++) { 11 for (int j = 0; j < SIZE; j++) { 12 temp[j][SIZE - 1 - i] = matrix[i][j]; 13 } 14 } 15 16 // Copy the rotated matrix back to the original matrix 17 for (int i = 0; i < SIZE; i++) { 18 for (int j = 0; j < SIZE; j++) { 19 matrix[i][j] = temp[i][j]; 20 } 21 } 22 } 23 24 void displayMatrix(int matrix[SIZE][SIZE]) { 25 // Display the matrix</pre>	<pre>Enter elements of the 4x4 matrix: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 Original Matrix: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 Matrix after 90-degree rotation: 13 9 5 1 14 10 6 2 15 11 7 3 16 12 8 4 === Code Execution Successful ===</pre>

9. Given a matrix (2D array) of integers, find the saddle point(s) (an element that is the minimum in its row and maximum in its column):

main.c	Output
<pre>1 #include <stdio.h> 2 3 #define ROWS 3 // Number of rows in the matrix 4 #define COLS 3 // Number of columns in the matrix 5 6 void findSaddlePoints(int matrix[ROWS][COLS]) { 7 int saddlePointFound = 0; 8 9 // Iterate through each row 10 for (int i = 0; i < ROWS; i++) { 11 // Find the minimum element in the current row 12 int minRow = matrix[i][0]; 13 int minColIndex = 0; 14 15 for (int j = 1; j < COLS; j++) { 16 if (matrix[i][j] < minRow) { 17 minRow = matrix[i][j]; 18 minColIndex = j; 19 } 20 } 21 22 // Check if the minimum element is the maximum in its column 23 int isMaxInCol = 1; 24 for (int k = 0; k < ROWS; k++) { 25 if (matrix[k][minColIndex] > minRow) { 26 isMaxInCol = 0; </pre>	<pre>Matrix: 1 2 3 4 5 6 7 8 9 Saddle point found at [2][0] = 7 === Code Execution Successful ===</pre>

10. Write a program to print the upper triangular part of a 4x4 matrix:

main.c	Output
<pre>1 #include <stdio.h> 2 3 #define SIZE 4 // Define matrix size as 4x4 4 5 void printUpperTriangular(int matrix[SIZE][SIZE]) { 6 printf("Upper Triangular Matrix:\n"); 7 8 for (int i = 0; i < SIZE; i++) { 9 for (int j = 0; j < SIZE; j++) { 10 // Print only the elements in the upper triangular part 11 if (i <= j) { 12 printf("%d ", matrix[i][j]); 13 } else { 14 printf("0 "); // For elements below the diagonal 15 } 16 } 17 printf("\n"); 18 } 19 } 20 21 int main() { 22 int matrix[SIZE][SIZE]; 23 24 // Input elements for the 4x4 matrix 25 printf("Enter elements of the 4x4 matrix:\n"); 26 for (int i = 0; i < SIZE; i++) { </pre>	<pre>Enter elements of the 4x4 matrix: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 Upper Triangular Matrix: 1 2 3 4 0 6 7 8 0 0 11 12 0 0 0 16 === Code Execution Successful ===</pre>

