Day 14: Symmetric Matrix

Gati Goyal

"Mathematics is the music of reason."
— James Joseph Sylvester

---

1 Introduction

A matrix is said to be symmetric if it is equal to its transpose. Symmetric matrices are square matrices, and their properties make them important in linear algebra, optimization, and other computational applications.

2 Problem Statement

Problem: Check if a given matrix is symmetric. **Hint:** Compare the elements of the matrix with its transpose. **Edge Case:** Ensure the matrix is square.

3 Properties of Symmetric Matrices

- A symmetric matrix is always square.
- The element at position (i, j) is equal to the element at position (j, i).
- Symmetric matrices have applications in graph theory, physics, and optimization problems.

4 Algorithm

1. Check if the matrix is square. If not, it cannot be symmetric.
2. Compare each element `mat[i][j]` with `mat[j][i]` for all i and j.
3. If all elements match, the matrix is symmetric.

5 Code

```c
#include <stdio.h>

// Function to check if a matrix is symmetric
int isSymmetric(int n, int matrix[n][n]) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (matrix[i][j] != matrix[j][i]) {
                return 0; // Not symmetric
            }
        }
    }
    return 1; // Symmetric
}

int main() {
    int n;

    printf("Enter the size of the square matrix: ");
    scanf("%d", &n);

    int matrix[n][n];

    printf("Enter elements of the %dx%d matrix:\n", n, n);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }

    printf("Matrix:\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }

    if (isSymmetric(n, matrix)) {
        printf("The matrix is symmetric.\n");
    } else {
        printf("The matrix is not symmetric.\n");
    }

    return 0;
}
```

6 Complexity Analysis

- **Time Complexity:** $O(n^2)$ The matrix elements are compared in a nested loop.
- **Space Complexity:** $O(1)$ No additional space is required beyond the input matrix.

7 Examples and Edge Cases

Matrix	Symmetric?
$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$	Yes
$\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$	No
$\begin{bmatrix} 1 \end{bmatrix}$	Yes (trivially symmetric)

8 Output

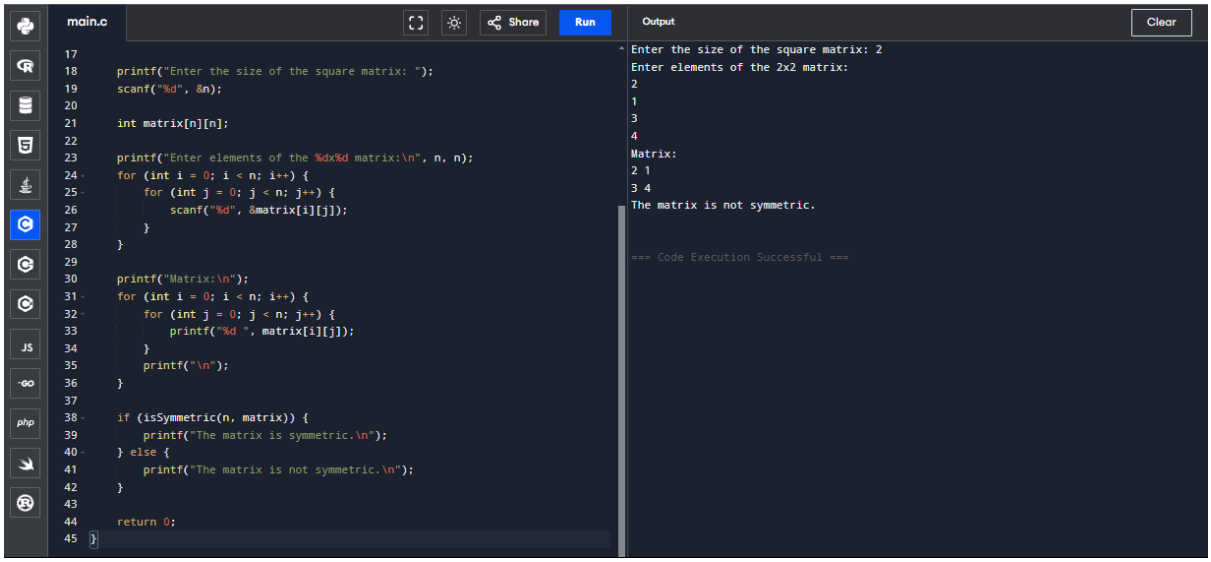

Figure 1: Program Output Screenshot

9 Conclusion

A symmetric matrix is a powerful mathematical concept with wide applications. This implementation checks symmetry efficiently with $O(n^2)$ complexity. The program handles all edge cases, including non-square matrices and single-element matrices.