🔢 Matrix Operations Documentation

🎯 Program Overview

This C++ program performs various matrix operations including input, display, transpose, and zigzag traversal of a 2D array.

🔑 Key Components

1. 1️⃣ Matrix Input

```
cout<<"Enter the number of rows: ";

cin>>m;

cout<<"Enter the number of columns: ";

cin>>n;

int arr[m][n];
```

The program starts by taking user input for matrix dimensions and elements.

2. 2️⃣ Matrix Display

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

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

        cout<<arr[i][j]<<" ";

    }

    cout<<endl;

}
```

Uses nested loops to display the matrix elements in a grid format.

3. 3️⃣ Matrix Transpose

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

    for(int j=i+1;j<m;j++){

       int temp=arr[i][j];

        arr[i][j]=arr[j][i];

        arr[j][i]=temp;

    }

}
```

- 🔄 Swaps elements across the main diagonal
- 📝 Uses temporary variable for swapping
- 📐 Only traverses upper triangular matrix to avoid double swapping

4. 4️⃣ Zigzag Traversal

```
for(int i=0;i<m;i++){
  if(i%2==0){
     for(int j=0;j<n;j++){
       cout<<arr[i][j]<<" ";
     }
  }
  else{
     for(int j=n-1;j>=0;j--){
       cout<<arr[i][j]<<" ";
     }
  }
}
```

Implements alternating row traversal:

- ➡️ Even rows: Left to right
- ⬅️ Odd rows: Right to left

⏱️ Time Complexity Analysis

Operation	Time Complexity
📥 Matrix Input	O(m×n)
📤 Matrix Display	O(m×n)
🔄 Matrix Transpose	O(m^2)
↔️ Zigzag Traversal	O(m×n)

💡 Potential Improvements

- ✅ Add input validation for matrix dimensions
- 💾 Implement dynamic memory allocation instead of fixed arrays
- ⚠️ Add error handling for invalid inputs
- ⚡ Optimize transpose operation for non-square matrices