

```
def reorder_elements(arr, index):
    # Create a new list to store reordered elements
    reordered_arr = [0] * len(arr)

    # Reorder elements according to index
    for i in range(len(arr)):
        reordered_arr[index[i]] = arr[i]

    return reordered_arr

# Input the number of elements
n = int(input("Enter the number of elements: "))

# Input the elements of the array
arr = []
for i in range(n):
    arr.append(int(input(f"Enter element {i+1} of the array: ")))

# Input the elements of the index array
index = []
for i in range(n):
    index.append(int(input(f"Enter index for element {i+1}: ")))

# Reorder the array based on the index
reordered_arr = reorder_elements(arr, index)

# Output the reordered array
print("Reordered array:", reordered_arr)
```

Enter the number of elements: 3  
Enter element 1 of the array: 3  
Enter element 2 of the array: 2  
Enter element 3 of the array: 1  
Enter index for element 1: 2  
Enter index for element 2: 1  
Enter index for element 3: 0  
Reordered array: [1, 2, 3]

```
def calculate_trace(matrix):
    # Ensure the matrix is square
    n = len(matrix)
    if not all(len(row) == n for row in matrix):
        raise ValueError("The matrix must be square.")

    # Calculate the trace by summing up the diagonal elements
    trace = sum(matrix[i][i] for i in range(n))
    return trace

# Input Matrix
matrix = [[1,2,3],
          [4,5,6],
          [7,8,9]]

calculate_trace(matrix)
```

15

```
def print_matrix(matrix):
    # Iterate over each row in the matrix
    for row in matrix:
        # Print each row as a space-separated string of numbers
        # print(" ".join(map(str, row)))
        print(row)

def transpose_matrix(matrix):
    # Get the number of rows and columns
    rows = len(matrix)
    cols = len(matrix[0])


    # Initialize the transpose matrix with switched dimensions
    transpose = [[0] * rows for _ in range(cols)]

    # Fill the transpose matrix by swapping rows with columns
    for i in range(rows):
        for j in range(cols):
            transpose[j][i] = matrix[i][j]

    return transpose
```

```
# Input matrix
matrix = [[1,2,3],
          [4,5,6]]

transpose_mat = transpose_matrix(matrix)
print_matrix(transpose_mat)
```



```
[1, 4]
[2, 5]
[3, 6]
```