01. Transpose Sparse Matrix

spare matrix를 전치하는 코드다. 확인을 위해 dense matrix형태로 반환해 출력한다.

코드 설명

| transposeMatrix 함수

Array num_of_col은 전치를 하고자하는 행렬 B의 열에 몇 개의 요소가 들어있나 센다. 이때 열의 인덱스는 array의 인덱스와 동일하다.

Array row_idx는 전치 이후 행렬의 행이 Sparse matrix에서 몇 번째 요소인지를 알려준다. 각각의 인덱스는 이전 행의 시작 인덱스에 이전 행의 요소 개수를 더한 것과 같다.

첫 번째 for문은 num_of_col을 0으로 초기화 시켜준다.

두 번째 for문은 기존 SparseMatrix의 terms, 다시 말해 요소의 총 개수를 바탕으로 num_of_col의 요소를 채운다.

세 번째 for문은 num_of_col을 바탕으로 row_idx array를 채운다.

마지막 for문은 SparseMatrix B를 순서대로 훑으면서 B의 열을 기준으로 전치시킨다. 각 요소의 위치는 row_idx를 바탕으로 하며, 같은 행의 다음 요소를 위해 +1을 더해준다.

| printDenseMatrix 함수

Sparse Matrix의 terms가 전부 훑는 while문을 이용한다. While문 안에는 이중 for문과 if 문이 있다. 이중 for문은 미리 입력된 규격의 행렬을 전부 비교한다. if문을 이용해 SparseMatrix 안에 정의되어 있는 위치에서는 해당 요소값을, 아니라면 0을 반환한다.

입력 : main에서 Matrix를 입력한다.

출력 :

Matrix B:

000700

900008

000000

650000

000001

002000

Transpose of B:

090600

000500

000002

700000

000000

080010

02. Allocate, Add, and Deallocate Matrix

동적 메모리를 할당해 행렬을 표현한다. 이 방식을 사용한 두 개의 행렬을 더하고, 동적 메모리 할당을 끝낸다.

| mem_alloc_3D_double() 함수

3D 배열에 동적 메모리를 할당한다. For문의 중첩을 통해 각 차원에 동적 메모리를 할당한다.

| addition_3D(input matrix1, input matrix2, output matrix)

For문의 중첩을 통해 같은 자리에 있는 행렬 요소를 더하고 값을 프린트한다.

| deallocate_3D_double(matrix의 시작 주소)

For문의 중첩을 통해 각 차원에 접근하고, 메모리를 거둬간다.

| assignMatrix (matrix의 시작 주소)

For문의 중첩을 통해 각 각의 요소들에 접근하게 해주며, 요소들은 scanf로 입력 받는다.

입력과 출력

- 입력단

input x, y, z (format = x y z): 2 2 2
Matrix A
matrix[0][0][0] = 1
matrix[0][0][1] = 2
matrix[0][1][0] = 3
matrix[0][1][1] = 4
matrix[1][0][0] = 3
matrix[1][0][1] = 1
matrix[1][1][0] = 2
matrix[1][1][1] = 2
Matrix B
matrix[0][0][0] = 4
matrix[0][0][1] = 2
matrix[0][1][0] = 3
matrix[0][1][1] = 4
matrix[1][0][0] = 5
matrix[1][0][1] = 4
matrix[1][1][0] = 2
matrix[1][1][1] = 3

- 출력단

matrix[0][0][0] = 5.000000
matrix[0][0][1] = 4.000000
matrix[0][1][0] = 6.000000
matrix[0][1][1] = 8.000000
matrix[1][0][0] = 8.000000
matrix[1][0][1] = 5.000000
matrix[1][1][0] = 4.000000
matrix[1][1][1] = 5.000000