5 Diagonal Matrix.

6a. Transpose of a matrix.

#include <iostream>

using namespace std;

int main() {

int row, col;

int trans[100][3]; // Transposed sparse matrix

int s[100][3]; // Sparse matrix

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

cin >> row;

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

cin >> col;

int arr[row][col];

cout << "Enter the elements of array:" << endl;

for (int i = 0; i < row; i++) {

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

cin >> arr[i][j];

}

}

// Display original matrix

cout << "\nOriginal Matrix:" << endl;

for (int i = 0; i < row; i++) {

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

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

}

cout << endl;

}

// Convert to sparse matrix

int k = 1;

for (int i = 0; i < row; i++) {

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

if (arr[i][j] != 0) {

s[k][0] = i;

s[k][1] = j;

s[k][2] = arr[i][j];

k++;

}

}

}

s[0][0] = row;

s[0][1] = col;

s[0][2] = k - 1;

// Display sparse matrix

cout << "\nSparse Matrix (Triplet form):\n";

cout << "Row Column Value" << endl;

for (int i = 0; i < k; i++) {

cout << s[i][0] << " " << s[i][1] << " " << s[i][2] << endl;

}

// Transpose sparse matrix

int x = 1;

for (int i = 0; i < s[0][1]; i++) { // loop over columns

for (int j = 1; j <= s[0][2]; j++) {

if (s[j][1] == i) {

trans[x][0] = s[j][1];

trans[x][1] = s[j][0];

trans[x][2] = s[j][2];

x++;

}

}

}

trans[0][0] = s[0][1];

trans[0][1] = s[0][0];

trans[0][2] = s[0][2];

// Display transposed sparse matrix

cout << "\nTranspose of Sparse Matrix:\n";

cout << "Row Column Value" << endl;

for (int i = 0; i < k; i++) {

cout << trans[i][0] << " " << trans[i][1] << " " << trans[i][2] << endl;

}

return 0;

}

OUTPUT :

