5 Diagonal Matrix.

6b. Addition of two matrices

#include <iostream>

using namespace std;

int main() {

int row, col;

int arr1[100][100], arr2[100][100];

int s1[100][3], s2[100][3], add[100][3];

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

cin >> row;

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

cin >> col;

cout << "Enter elements of Matrix 1:" << endl;

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

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

cin >> arr1[i][j];

}

}

cout << "Enter elements of Matrix 2:" << endl;

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

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

cin >> arr2[i][j];

}

}

// Display original matrices

cout << "\nMatrix 1:" << endl;

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

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

cout << arr1[i][j] << " ";

}

cout << endl;

}

cout << "\nMatrix 2:" << endl;

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

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

cout << arr2[i][j] << " ";

}

cout << endl;

}

// Convert Matrix 1 to sparse form

int k = 1;

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

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

if (arr1[i][j] != 0) {

s1[k][0] = i;

s1[k][1] = j;

s1[k][2] = arr1[i][j];

k++;

}

}

}

s1[0][0] = row;

s1[0][1] = col;

s1[0][2] = k - 1;

cout << "\nSparse Matrix 1 (Row Col Value):" << endl;

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

cout << s1[i][0] << " " << s1[i][1] << " " << s1[i][2] << endl;

}

// Convert Matrix 2 to sparse form

int z = 1;

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

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

if (arr2[i][j] != 0) {

s2[z][0] = i;

s2[z][1] = j;

s2[z][2] = arr2[i][j];

z++;

}

}

}

s2[0][0] = row;

s2[0][1] = col;

s2[0][2] = z - 1;

cout << "\nSparse Matrix 2 (Row Col Value):" << endl;

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

cout << s2[i][0] << " " << s2[i][1] << " " << s2[i][2] << endl;

}

// Add the sparse matrices

int m = 1, i = 1, j = 1;

add[0][0] = row;

add[0][1] = col;

while (i < k && j < z) {

if (s1[i][0] == s2[j][0]) {

if (s1[i][1] == s2[j][1]) {

int sum = s1[i][2] + s2[j][2];

if (sum != 0) {

add[m][0] = s1[i][0];

add[m][1] = s1[i][1];

add[m][2] = sum;

m++;

}

i++;

j++;

} else if (s1[i][1] < s2[j][1]) {

add[m][0] = s1[i][0];

add[m][1] = s1[i][1];

add[m][2] = s1[i][2];

i++;

m++;

} else {

add[m][0] = s2[j][0];

add[m][1] = s2[j][1];

add[m][2] = s2[j][2];

j++;

m++;

}

} else if (s1[i][0] < s2[j][0]) {

add[m][0] = s1[i][0];

add[m][1] = s1[i][1];

add[m][2] = s1[i][2];

i++;

m++;

} else {

add[m][0] = s2[j][0];

add[m][1] = s2[j][1];

add[m][2] = s2[j][2];

j++;

m++;

}

}

while (i < k) {

add[m][0] = s1[i][0];

add[m][1] = s1[i][1];

add[m][2] = s1[i][2];

i++;

m++;

}

while (j < z) {

add[m][0] = s2[j][0];

add[m][1] = s2[j][1];

add[m][2] = s2[j][2];

j++;

m++;

}

add[0][2] = m - 1;

cout << "\nAddition Result (Sparse Matrix Form):" << endl;

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

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

}

return 0;

}

OUTPUT :



