**Experiment 1**

**AIM:** Write a program to add 2 sparse matrices

**THEORY:** A matrix is a two-dimensional data object made of m rows and n columns, therefore having total m x n values. If most of the elements of the matrix have 0 value, then it is called a sparse matrix.  
Why to use Sparse Matrix instead of simple matrix?

* Storage: There are lesser non-zero elements than zeros and thus lesser memory can be used to store only those elements.
* Computing time: Computing time can be saved by logically designing a data structure traversing only non-zero elements.

**Code:**

1. #include < iostream >
2. using namespace std;
3. struct sparse {
4. int row, col, val;
5. sparse() {}
6. sparse(int r, int c, int v) {
7. row = r;
8. col = c;
9. val = v;
10. }
11. };
12. sparse a[20], b[20], ans[20];
13. void read\_sparse(sparse \* s) {
14. cin >> s[0].row >> s[0].col >> s[0].val;
15. for (int i = 1; i <= s[0].val; i++)
16. cin >> s[i].row >> s[i].col >> s[i].val;
17. }
18. void print\_sparse(sparse \* s) {
19. cout << endl;
20. int k = s[0].val;
21. for (int i = 0; i <= k; i++)
22. cout << s[i].row << " " << s[i].col << " " << s[i].val << endl;
23. cout << endl;
24. }
25. void add\_sparse(sparse \* a, sparse \* b, sparse \* c) {
26. int i = 1, j = 1, k = 1;
27. int i\_max = a[0].val, j\_max = b[0].val;
28. while (i <= i\_max && j <= j\_max) {
29. if (a[i].row < b[j].row)
30. c[k++] = a[i++];
31. else if (a[i].row == b[j].row) {
32. if (a[i].col < b[j].col)
33. c[k++] = a[i++];
34. else if (a[i].col == b[j].col) {
35. c[k] = sparse(a[i].row, a[i].col, a[i].val + b[j].val);
36. i++;
37. j++;
38. k++;
39. }
40. else
41. c[k++] = b[j++];
42. }
43. else
44. c[k++] = b[j++];
45. }
46. while (i <= i\_max)
47. c[k++] = a[i++];
48. while (j <= j\_max)
49. c[k++] = b[j++];
50. c[0] = sparse(a[0].row, a[0].col, k - 1);
51. }
52. int main() {
53. read\_sparse(a);
54. read\_sparse(b);
55. add\_sparse(a, b, ans);
56. cout << "ans:\n";
57. print\_sparse(ans);
58. return 0;
59. }

**Output:**

