**Experiment 2**

**AIM:** Write a program to convert a given matrix into sparse matrix.

**Algo:**

Initialize r,c,m[10][10],k=1

read r

read c

for i=0 to r-1 do:

for j=0 to c-1 do:

read m[i][j];

endfor

endfor

for id=0 to y-1 do:

read b[id];

endfor

define structure sparse:

int r,c,v

Initialize sparse s[100]

for i=0 to r-1 do:

for j=0 to c-1 do:

if m[i][j] != 0 then:

s[k] = sparse(i,j,m[i][j])

k=k+1

endif

endfor

endfor

s[0]=sparse(r,c,k-1)

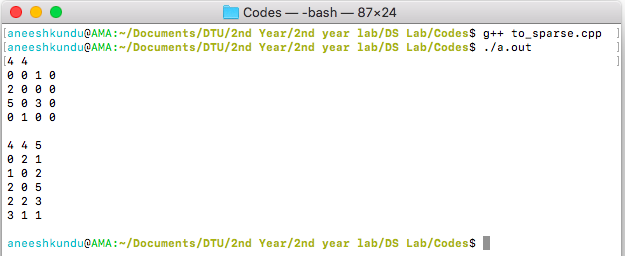
for i=0 to s[0].v do:

write s[i].r, s[i].c, s[i].v

**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];
13. int m[10][10], r, c;
14. void create\_sparse(sparse \* s) {
15. s[0] = sparse(r, c, 0);
16. int k = 1;
17. for (int i = 0; i < r; i++) {
18. for (int j = 0; j < c; j++) {
19. if (m[i][j])
20. s[k++] = sparse(i, j, m[i][j]);
21. }
22. }
23. s[0].val = k - 1;
24. }
25. void print\_sparse(sparse \* s) {
26. cout << endl;
27. int k = s[0].val;
28. for (int i = 0; i <= k; i++)
29. cout << s[i].row << " " << s[i].col << " " << s[i].val << endl;
30. cout << endl;
31. }
32. int main() {
33. cin >> r >> c;
34. for (int i = 0; i < r; i++) {
35. for (int j = 0; j < c; j++)
36. cin >> m[i][j];
37. }
38. create\_sparse(a);
39. print\_sparse(a);
40. return 0;
41. }

**Output:**



**Discussion:**

In this program, we iterate over every column of every row, if the value in that cell is non zero we store the row, column, and value in the sparse table.

**Conclusion:**

The program converts a matrix, into its sparse form with time complexity .