

Program

```
#include <stdio.h>
int sp[100][3], t[100][3];
void transpose(int m[][3])
{
    int r, c, k = 1;
    t[0][0] = m[0][1];
    t[0][1] = m[0][0];
    t[0][2] = m[0][2];
    for (r = 0; r < m[0][1]; r++)
    {
        for (c = 1; c <= m[0][2]; c++)
        {
            if (m[c][1] == r)
            {
                t[k][0] = m[c][1];
                t[k][1] = m[c][0];
                t[k][2] = m[c][2];
                k++;
            }
        }
    }
    printf("\nTranspose of the sparse matrix:\n");
    for (r = 0; r < k; r++)
    {
        printf("%d\t%d\t%d\n", t[r][0], t[r][1], t[r][2]);
    }
}

int main()
{
    printf("Enter the number of rows and columns of the sparse matrix representation:\n");
    scanf("%d %d", &sp[0][0], &sp[0][1]);
    printf("Enter the number of non-zero elements:\n");
    scanf("%d", &sp[0][2]);
    printf("Enter the sparse matrix representation (row, column, value):\n");
    for (int i = 1; i <= sp[0][2]; i++)
    {
        scanf("%d %d %d", &sp[i][0], &sp[i][1], &sp[i][2]);
    }
    transpose(sp);
    return 0;
}
```

Output

```
Enter the number of rows and columns of the sparse matrix representation:
34
Enter the number of non-zero elements:
6
Enter the sparse matrix representation (row, column, value):
0
0
1
1
2
6
1
3
7
2
```

SPARSE MATRIX TRANSPOSE

Aim:

To write a program to find out the sparse matrix transpose of a given matrix

Algorithm:

1. Start
2. Declare global variables sp[100][3],t[100][3]
3. Define function transpose(int m[][3])

```
    Declare integer variables r, c and initialise k = 1
    t[0][0] = m[0][1]
    t[0][1] = m[0][0]
    t[0][2] = m[0][2]
```

```
    For r=0 to r<m[0][1] do
        For c=1 to c<=m[0][2] do
            If m[c][1]==r
                t[k][0] = m[c][1]
                t[k][1] = m[c][0]
                t[k][2] = m[c][2]
                k++
```

4. Print "Transpose of sparse matrix"

```
    For r=0 to k
        Print value of t[r][0],t[r][1],t[r][2]
```

5. In main function ()

```
    Read number of rows and columns and non-zero elements of the matrix
    Read sparse matrix
    Call function transpose(sp)
    End main function
```

6. Stop

1
8
2
2
4
2
3
10

Transpose of the sparse matrix:

4 3 6
0 0 1
1 2 8
2 1 6
2 2 4
3 1 7
3 2 10

Result:

Program has been executed successfully and obtained the output