Program 2.8: Transpose of a sparse matrix

```
1.  void transpose (term a[], term b[])
2.  { /* b is set to the transpose of a */
3.    int n,i,j, currentb;
4.    n=a[0].value; /* total number of elements */
5.    b[0].row= a[0].col; /* rows in b= columns in a */
6.    b[0].col= a[0].row; /* columns in b= rows in a */
7.    b[0].value=n;
8.    if(n>0) { /* non zero matrix */
9.      currentb=1;
10.     for(i=0; i<a[0].col; i++)
        /* transpose by the columns in a */
11.       for(j=1; j<=n; j++)
          /* find elements from the current column */
12.       if(a[j].col==i) {
          /* element is in current column, add it to b */
13.         b[currentb].row= a[j].col;
14.         b[currentb].col=a[j].row;
15.         b[currentb].value=a[j].value;
16.         currentb++;
17.       }
18.   }
19. }
```

. The matrix b stores the transpose of the sparse matrix a. Since b is of type term, it follows that b[0].row stores the no. of rows of b, b[0].col stores the no. of columns of b and b[0].value stores the no. of non-zero elements in b.

. It's easy to see that b[0].value= a[0].value=n, since b is only the transpose of a. By defn of transpose matrix, b[0].row=a[0].col and b[0].col =a[0].row. We have done just that in lines 4-7.