Program 2.9: Fast transpose of a sparse matrix

```
1. void fastTranspose(term a[], term b[])
2. { /* the transpose of a is placed in b */
3.   int rowTerms[MAX_COL], startingPos[MAX_COL];
4.   int i,j, numCols= a[0].col, numTerms= a[0].value;
5.   b[0].row = numCols; b[0].col= a[0].row;
6.   b[0].value= numTerms;
7.   if(numTerms >0) { /* nonzero matrix */
8.     for(i=0; i<numCols; i++)
9.       rowTerms[i]=0;
10.  for(i=1; i<= numTerms; i++)
11.      rowTerms[a[i].col]++;
12.  startingPos[0]= 1;
13.  for(i=1; i<numCols; i++)
14.     startingPos[i]= startingPos[i-1]+ rowTerms[i-1];
15.  for(i=1; i<= numTerms; i++) {
16.     j=startingPos[a[i].col]++;
17.     b[j].row=a[i].col; b[j].col= a[i].row;
18.     b[j].value= a[i].value;
19.  }
20. }
21. }
```

rowTerms[ ] stores the no. of non-zero elements corresponding to each column i, where 0≤i<a[0].col.

startingPos[ ] stores the start indexes of b[ ] corresponding to each entry beginning a new row in b[ ], in ascending order, i.e. ∀i, 0≤i<numCols, startingPos[i] stores the index of the first non-zero element of row i of matrix b[ ].

Loop invariant of lines 8-9: Just before the ith iteration of the loop, for all indices K, where 0≤K<i, the no. of non-zero elements of column K of a[ ], i.e. rowTerms[K] has been initially initialized to 0.

Initialization: At the start of the 1st iteration of the loop, i=0. Just before the start of the iteration, the range of indices 0≤K<i=0