1. Let $A$ be a sparse matrix. We store $A$ in a sparse matrix format, involving vectors $r, c$ of integers and $v$ of real numbers, in the following way: for each nonzero entry in $A$, we assign a unique integer $k$ and have $(r_k, c_k)$ record the row and column position of the nonzero entry, and $v_k$ record the value of the nonzero entry. Thus, instead of inputing $A$, we input

- dimension $n$;
- number of nonzero elements $m$;
- column vectors $r, c, v$ of $m$ components.

Using basic programming (for loops, while loops, and if statements):

(a) Write a function that inputs the $n, m, r, c, v$ and outputs the actual $n \times n$ matrix $A$ being represented by the sparse matrix format. Print out or write out this function.

(b) Write a function that inputs the $n, m, r, c, v$ and additionally a vector $y$ of $n$ components, performs $Ay$ using only the sparse matrix format (do not generate $A$), and outputs the number of flops used. Print out or write out this function.

(c) Write a function that inputs the $n, m, r, c, v$ and additionally a vector $b$ of $n$ components. Assume, in the input, the sparse matrix $A$ being represented is lower triangular and $c$ is in non-decreasing order. Have the function perform column oriented forward substitution solving $Ax = b$ using only the sparse matrix format (do not generate $A$), and output the number of flops used. Print out or write out this function.

(d) Apply your three functions to the case $n = 8$, $m = 16$, and

$$r = [1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 8]^T,$$
$$c = [1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8]^T,$$
$$v = [2, -1, 2, -1, 2, -1, 2, -1, 2, -1, 2, -1, 2, -1, 2]^T,$$
$$y = [1, 2, 3, 4, 5, 6, 7, 8]^T,$$
$$b = [1, 2, 3, 4, 5, 6, 7, 8]^T.$$

Print out or write out the results of each function.