1. Write a program that reads integers between 0 and 100 from the user, after that the program should print out the numbers in ascending order. Also print how many times the number was given.

2. Matrices are two dimensional arrays, for example

   $$A=(a_{ij})_{3\times 3}=\begin{pmatrix} 1 & 2 & 3 \\ 1 & 0 & 1 \\ -1 & 5 & 0 \end{pmatrix}$$

   is some 3x3 matrix. The elements of a matrix are referred to using two indices, from which the first one tells the row and the second the column. For example, above $a_{12}=2$ and $a_{31}=-1$.

   If $A=(a_{ij})_{3\times 3}$ and $B=(b_{ij})_{3\times 3}$ are matrices, their sum is $(c_{ij})_{3\times 3}=(a_{ij}+b_{ij})_{3\times 3}$. For example,

   $$\begin{pmatrix} 1 & 2 & 3 \\ -1 & 0 & 1 \\ 3 & 1 & 0 \end{pmatrix}+\begin{pmatrix} 2 & 1 & 0 \\ 4 & -7 & 1 \\ 0 & 4 & 1 \end{pmatrix}=\begin{pmatrix} 3 & 3 & 3 \\ 3 & -7 & 2 \\ 3 & 5 & 1 \end{pmatrix} .$$

   Similarly for difference of matrices, $A-B=C=(c_{ij})_{3\times 3}=(a_{ij}-b_{ij})_{3\times 3}$.

   Write functions that implement the addition and substraction of two 3x3-matrices (3 rows and 3 columns). You can initialize two matrices in the main function in any way you'd like to. The prototypes for the two functions could be

   ```
   void msum(double a[][3], double b[][3],double sum[][3]);

   void mdiff(double a[][3], double b[][3],double diff[][3]);
   ```

   The first and second arguments are the matrices to be summed / substracted, and the third is the matrix where the result is to be stored.

   You'll need two (one nested) for loop. The first should go through the the row indices of the matrices, and the second should scan the column indices. Remember that the indices start from zero in C. Print the results in the main function.

3. Extra: Write a function that sorts any real number array to ascending order. Write a main function that reads the numbers from the user and then uses your sort function, and then prints the results to the screen.