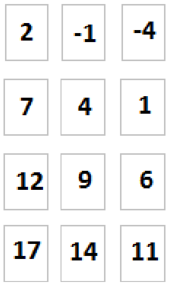
Matrix to List

We have seen and used 2 dimensional matrices such as this one



2	-1	-4
7	4	1
12	9	6
17	14	11

It is often convenient and appropriate to represent values in such a fashion. However, if truth be told, computer memory is organized sequentially, as a long list of locations. To store a matrix such as the one above the computer needs to “flatten” the matrix into a list of values. There are two general ways to do this: **row-major** order and **column-major** order. For **row-major** order we preserve the order in the rows by starting the list with the first row of values, following that by the second row of values, and that by the third row of values and so on. If our example matrix is stored in **row-major** order then it becomes

2 -1 -4 7 4 1 12 9 6 17 14 11

On the other hand, **column-major** order preserves the order in the columns. The values of the first column are followed by the values of the second column and so forth. If our example matrix is stored in **column-major** order then it becomes

2 7 12 17 -1 4 9 14 -4 1 6 11

Some computer languages use **row-major** order and others use **column major** order. There are times when it becomes essential to understand the real order of storing values when representing a matrix.