1. What is the problem with two-dimensional arrays in C/C++?

Trying to pass a 2D array to a function is difficult, as when you want to compile you have to specify the array dimensions. Once your program is called, it can only handle that one size specified. For example, if you want to manipulate an image, you can only handle an image that has one certain pixel size which makes the complicated program only useful for one image.

2. Describe two ways to work around C/C++'s problems with two dimensional arrays.

Two ways to work around this problem are using a flat array and the numerical recipes trick. To flatten an array, the data points are stored in one large array one by one. If you know the width and the height of the array you are working in, you can access each element by computing the offset. For the numerical recipes trick, you add an additional array which stores pointers to the start of each "row" in the flat array. This allows you to access any point in the array by typing in [x][y] which will give you the data point at row x and column y.

3. Is your computer big endian or little endian? Hint: write a small program to find out.

My computer is a little endian. When I write a program that can get the individual bytes of an integer, it starts with a 1 rather than 0 indicating that its putting the larger byte values at the beginning.