Kevin Chevalier

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

The issue with 2D arrays are that pointers do not track the dimensions of the array, which is critical for accessing values within multiple rows. This is a problem especially if a pointer is needed to be passed into a function. One standard solution is to specify the dimensions in the array, which is also inconvenient if the dimensions are not known/fixed.

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

Both methods have the same concept to use a 1D array to store the contents of the array, but use different strategies to access the elements. The "Flat Array" method uses a simple formula to convert the row and column index to a single index. ex: arr[j + i*w]. The "Numerical Recipes Trick" method uses an auxiliary array of pointers to reference the start of each row within the 1D array. This method allows the user to continue to use a row and column index. ex: arr[i][j].

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

```
int main() {
  int x = 1;
  printf("%c\n", *(char *)&x + '0');
}
```

The output is not zero so my computer placed the first bit in the first byte so my computer is little endian.