int data[3][4] = { {1,2,3,4}, {9,8,7,6}, {2,4,6,8}};
int total = sum(data, 3);

What should the prototype for sum() look like? And why does the function pass the number of rows (3) as an argument and not also the number of columns (4)?

Well, data is the name of an array with three elements. The first element is, itself, an array of 4 int values. Thus, the type of data is pointer-to-array-of-4-int, so an appropriate prototype would be this:

int sum(int (*ar2)[4], int size);

The parentheses are needed, for the declaration

int *ar2[4]

would declare an array of four pointers-to-int instead of a single pointer-to-array-of-4-int, and a function parameter cannot be an array. There's an alternative format that means exactly the same thing as the correct prototype, but, perhaps, is easier to read:

int sum(int ar2[][4], int size);

Either prototype states that ar2 is a pointer, not an array. Also note that pointer type specifically says it points to an array of 4 int. Thus, the pointer type specifies the number of columns, which is why the number of columns is not passed as a separate function argument.

Because the pointer type specifies the number of columns, the sum() function only works with arrays with four columns. But the number of rows is specified by the variable size, so sum() can work with a varying number of rows:

int a[100][4];
int b[6][4];
...
int total1 = sum(a, 100);  // sum all of a
int total2 = sum(b, 6);    // sum all of b
int total3 = sum(a, 10);   // sum first 10 rows of a
int total4 = sum(a+10, 20); // sum next 20 rows of a