Multidimensional Arrays

Enrich your knowledge with multidimensional arrays which will prove to be very useful in solving programming problems.

We'll cover the following

- Matrix Calculations

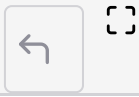
We have seen single-dimensional arrays. C provides for multi-dimensional arrays as well. Here is how we can define a two-dimensional array of integers, initialize values, and index into it to read off values:

```
#include <stdio.h>

int main ()
{
  int grades[2][2] = {1,2,3,4};
  int i,j;
  for (i=0; i<2; i++) {
    for (j=0; j<2; j++) {
      printf("grades[%d][%d]=%d\n", i, j, grades[i][j]);
    }
  }
  return 0;
}
```

Here is how we would define a three-dimensional array:

```
#include <stdio.h>

int main ()
{
  int grades[2][2][2] = {1,2,3,4,5,6,7,8};
  int i,j,k;
  for (i=0; i<2; i++) {
    for (j=0; j<2; j++) {
      for (k=0; k<2; k++) {
    printf("grades[%d][%d][%d]=%d\n", i, j, k, grades[i][j][k]);
      }
    }
  }
  return 0;
```

```
}
```



Matrix Calculations

Although you can use multidimensional arrays to represent matrices, if you are going to be doing a lot of matrix calculation in your C code, it will probably be better to make use of one of the pre-existing APIs for matrix algebra, rather than coding up this stuff yourself. Two common choices are:

1. The GNU Scientific Library Vectors and Matrices
2. LAPACK (and BLAS) libraries

In the next lesson, we'll deal with arrays that don't have to be defined before runtime. It might sound a bit confusing, but bear with me. It's a useful tool to have.