How do you find size of every dimension in a multi-dimensional array?

```
#include <stdio.h>
int main (void)
{

    int array[11][22][33];
    int x1 = sizeof(array) / sizeof(*array);  /* = 11 */
    int x2 = sizeof(array) / sizeof(array[0]);  /* = 11 */
    int y1 = sizeof(*array) / sizeof(**array);  /* = 22 */
    int y2 = sizeof(array[0]) / sizeof(array[0][0]);  /* = 22 */
    int z1 = sizeof(**array) / sizeof(***array);  /* = 33 */
    int z2 = sizeof(array[0][0]) / sizeof(array[0][0][0]);  /* = 33 */

   printf ("%d  %d  %d\n", x1, y1, z1);
   printf ("%d  %d  %d\n", x2, y2, z2);
   return 0;
}
```

Accessing Array Elements

Table (next slide) lists different ways of accessing your **board** array. The left column contains row index values to the **board** array, and the top row contains column index values. The entry in the table corresponding to a given row index and column index shows the various possible expressions for referring to that element.

```
char board[3][3] =
   {
       {'1','2','3'},
       {'4','5','6'},
       {'7','8','9'}
   };
```

Table - Pointer Expressions for Accessing Array Elements

board	0	1	2
0	board[0][0] *board[0] **board	board[0][1] *(board[0]+1) *(*board+1)	board[0][2] *(board[0]+2) *(*board+2)
1	board[1][0] *(board[0]+3) *board[1] *(*board+3)	board[1][1] *(board[0]+4 *(board[1]+1) *(*board+4)	board[1][2] *(board[0]+5) *(board[1]+2) *(*board+5)
2	board[2][0] *(board[0]+6) *(board[1]+3) *board[2] *(*board+6)	board[2][1] *(board[0]+7) *(board[1]+4) *(board[2]+1) *(*board+7)	board[2][2] *(board[0]+8) *(board[1]+5) *(board[2]+2) *(*board+8)