```c
/* File CExamples/arrays.c */
#include <stdio.h>
#include <stdlib.h> /* for calloc */
int main(void)
{int digits[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
 /* the dimension is optional and will be determined by the compiler
    note that char name[] = "Erich";      is a shorthand for
              char name[6] = {'E', 'r', 'i', 'c', 'h', '\0'}; */
 float* xptr[5]; /* array of pointers to float */
 float xy[5][5] = {{0.0, 0.1, 0.2, 0.3, 0.4},
                   {1.0, 1.1, 1.2, 1.3, 1.4},
                   {2.0, 2.1, 2.2, 2.3, 2.4},
                   {3.0, 3.1, 3.2, 3.3, 3.4},
                   {4.0, 4.1, 4.2, 4.3, 4.4}};
 int i, j;

 printf("digits[4] == %d, %d\n",   digits[4], *(digits+4)    );

 printf("xy[2][3] == %4.1f, %4.1f, %4.1f, %4.1f, %4.1f, %4.1f\n",
        xy[2][3], *(xy[2]+3), *(*(xy + 2) + 3), *((float*)xy + 2*5 + 3),
        (*(xy+2))[3], /* Wen-shin's way */
        *(&xy[0][0] + 2*5 + 3) /* the 6th version */ );

 /* initialize the array of pointers */
 for(i=0; i<5; i++)
    {xptr[i] = (float *)calloc(5, sizeof(float));
     for(j=0; j<5; j++)
        *(xptr[i] + j) = i + 0.1 * j;
    } /* end for i */

 printf("xptr[2][3] == %4.1f, %4.1f, %4.1f, %4.1f\n",
        xptr[2][3], *(xptr[2]+3), *(*(xptr + 2) + 3),
        (*(xptr+2))[3]); /* Wen-shin's way */
 for(i=0; i<5; i++) free(xptr[i]);
 return 0;
}
```