```c
/*****************************************************************************
 File name:   pointers.c
 Author:      $Author: chadd $
 Date:        Sep 7, 2011
 Class:       CS300
 Assignment:  In Class Example
 Purpose:     Demonstrate passing an array to a function and accessing the
              array elements through array notation and pointer arithmetic.
              Demonstrate passing a pointer to an int to a function and
              updating that int inside the function.
 RevisionID:  $Id: pointers.c 37 2011-09-12 15:22:04Z chadd $
 *****************************************************************************/

#include <stdio.h>
#include <stdlib.h>

/*****************************************************************************
 Function:    printIt
 Description: For each element of an array this function prints:
              Memory address, value, value

 Parameters:  pInt - the int array
              size - the size of the array
 Returned:    None
 *****************************************************************************/
void printIt (int *pInt, int size)
// int pInt[]
{
  int i;
  for (i = 0; i < size; i++)
  {

    /* Notice that the pointer notation, pInt[i], is much more
     * concise and clear than the pointer arithmetic, *(pInt + i).
     */
    printf ("%p %d %d \n", (void*) (pInt + i),
    pInt[i], *(pInt + i));
  }
  return;
}

/*****************************************************************************
 Function:    updateInt
 Description: Read an int from input and store twice that int into the
              memory location provided as an argument.

 Parameters:  pInt - the pointer to an int int
 Returned:    None
```

```c
 ***************************************************************************/
void updateInt (int *pInt)
{
  int input;
  scanf ("%d", &input);
  *pInt = input * 2;
  printf ("Double (in function): %d\n", *pInt);
}

/***************************************************************************
 Function:    main
 Description: Declares a static array and calls printIt.
              Also calls updateInt
 Parameters:  None
 Returned:    None
 ***************************************************************************/
int main ()
{
  int array[4] = { 1, 2, 3, 4 };

  int getDouble;

  printIt (array, 4);

  updateInt (&getDouble);
  printf ("Double (in main): %d\n", getDouble);

  return EXIT_SUCCESS;
}
```