```c
// ECEN 3000 (Lab 2 Prelab)
// Austin Glaser and Jesse Ellison
// This is an implementation of a decimal-to-morse
// conversion function, with an integrated test.

#include <string.h>
#include <stdint.h>
#include <assert.h>
#include <stdio.h>

// Takes an unsigned integer input and converts it into
// the morse code equivalent (returning an array where
// 1 is a dash (long) and 0 is a dot (short).
//
// Uses a lookup table to speed up the conversion
int decimal_to_morse(uint8_t dec, uint8_t morse_out[])
{
  if (dec > 9) return -1;

  uint8_t morse_lookup[10][5] = {
                                  {1, 1, 1, 1, 1},  //0
                                  {0, 1, 1, 1, 1},  //1
                                  {0, 0, 1, 1, 1},  //2
                                  {0, 0, 0, 1, 1},  //3
                                  {0, 0, 0, 0, 1},  //4
                                  {0, 0, 0, 0, 0},  //5
                                  {1, 0, 0, 0, 0},  //6
                                  {1, 1, 0, 0, 0},  //7
                                  {1, 1, 1, 0, 0},  //8
                                  {1, 1, 1, 1, 0}   //9
                                 };

  memcpy(morse_out, morse_lookup[dec], 5);

  return 0;
}

int main(void)
{
  uint8_t morse_out[5];
  int32_t i, j;

  for (i = 0; i <= 9; i++) {
    assert(!decimal_to_morse(i, morse_out));

    printf("%d -> ", i);
    for (j = 0; j < 5; j++) printf("%d ", morse_out[j]);
    printf("\n");
  }

  assert(decimal_to_morse(50, morse_out));

  return 0;
}
```