```c
// lab00, lab procedure, tasks 1 & 3

// source file

/*
 * this file contains the implementation of the functions relevant for
 * lab00.
 * The declarations of the functions are in the corresponding header
 * file (functions.h).
*/

// the header file needs to be included
#include "functions.h"

/* task 1
 * The function print_bits() accepts an input of type uint16_t
 * (arg_word) and has no return value (void).
 * The function simply writes the binary and hexadecimal number of the
 * input to the terminal.
*/
void print_bits(uint16_t arg_word)
{
  // only a simple implementation of a for loop is given
  // you are free to use it or solve the problem in another way
  uint16_t one = 1;
  uint16_t array[16];
  uint16_t arg_word2 = arg_word;
  int i;
  for (i = 0; i <= 15; i++)
  {
      if (arg_word & one == 1)
      {
        array[15-i] = 1;
      }
      else
      {
        array[15-i] = 0;
      }

      arg_word = arg_word >> 1;
  }

  printf("hex: 0x%x, bin: ",arg_word2);

  for(i=0;i<4;i++)
  {
    int k = 4*i;
    printf("%d%d%d%d ", array[k],array[k+1],array[k+2],array[k+3]);
  }


  /*int i;
  for(i = 0; i <= 0; i++)
  {
    printf("i = 0x%x\n", i);
  }*/
}

/* task 3
 * The function bit_merge() accepts two uint16_t as inputs (lsb and msb)
```

```
61  * and combines them to a uint32_t number by merging them.
62  * The return value is a uint32_t number.
63  */
64  uint32_t bit_merge(uint16_t lsb, uint16_t msb)
65  {
66    // for the moment, the function only returns 0
67    uint32_t lsbb = lsb;
68    uint32_t msbb = msb;
69    msbb = msbb << 16;
70    uint32_t result = lsbb | msbb ;
71    return result;
72  }
73
```