LAB NASM data and bss examples and problems (NASM/Linux Platform)

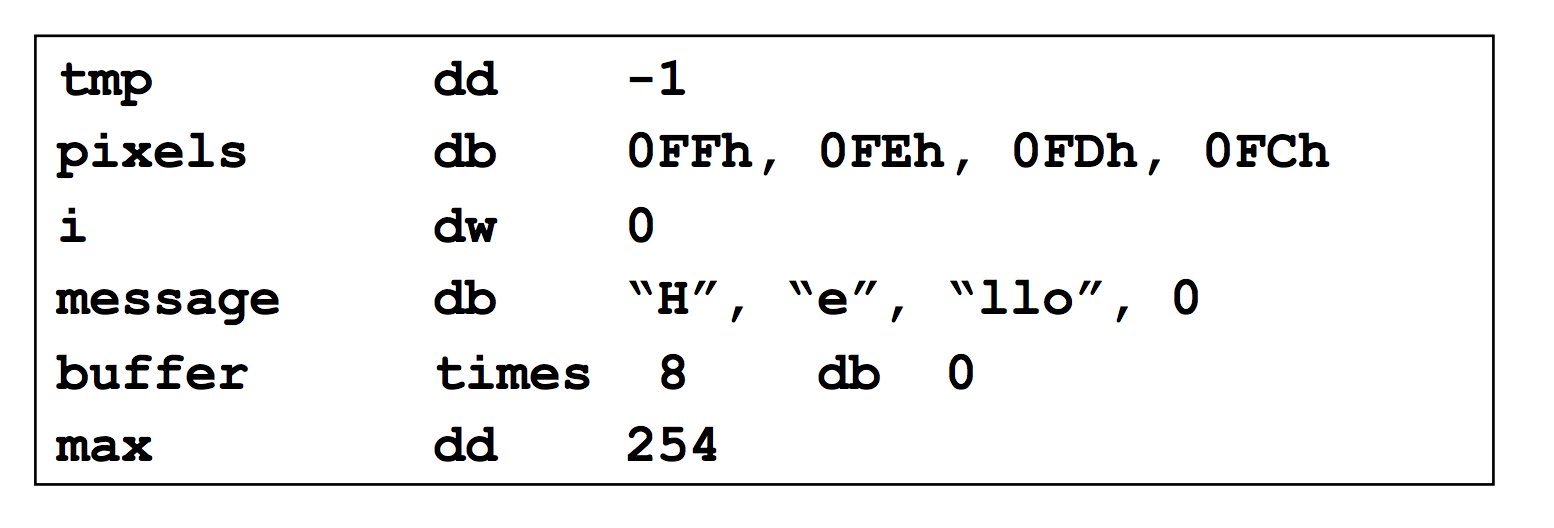
# Part One: Data section declaration, and how it is stored in memory

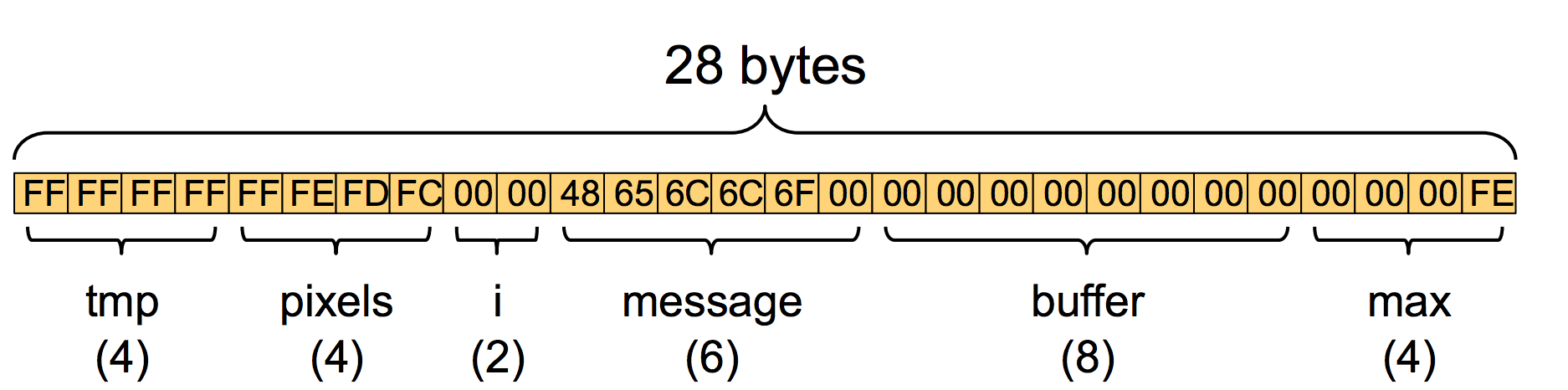
DB is 8 bits

DW is 16 bits

DD is 32 bits

### Exercise #1a

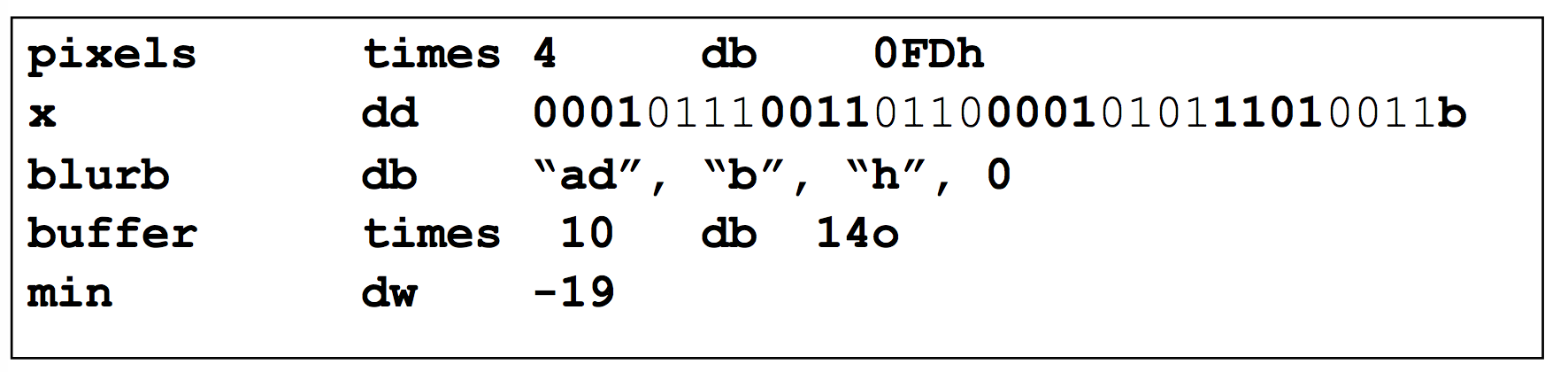




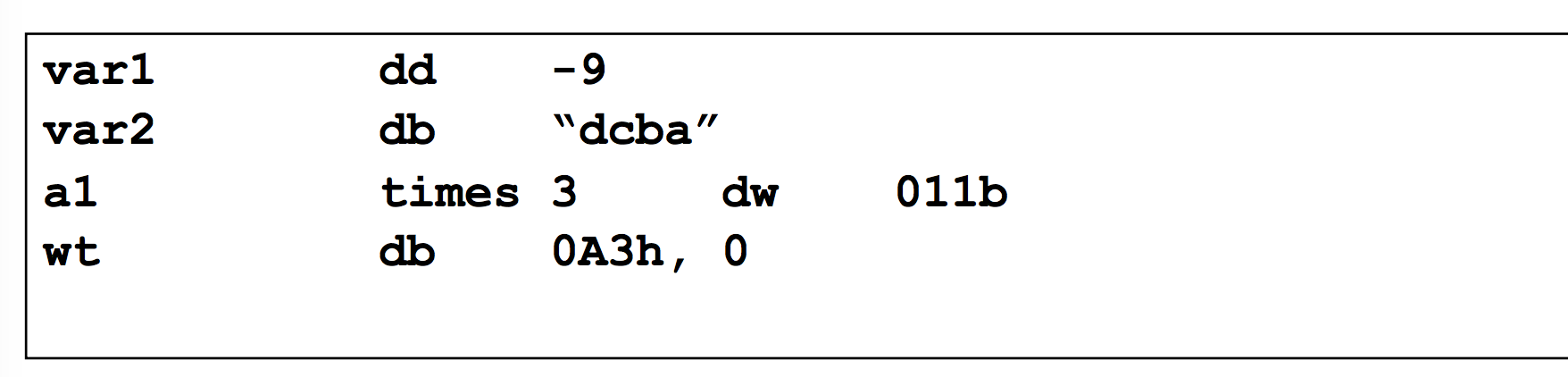
* nasm -felf64 ex1a.asm -l ex1a.lst

### Exercise #1b

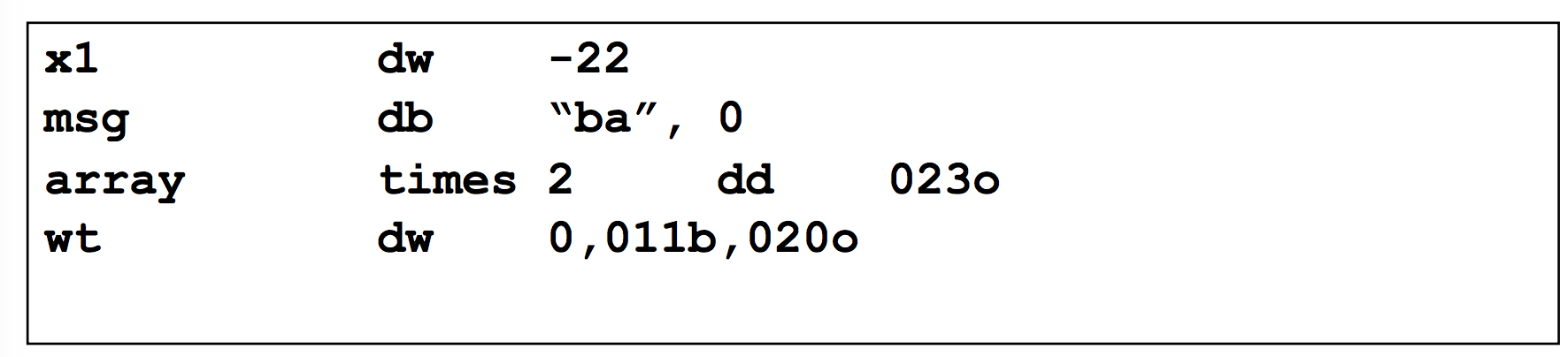
What is the layout and content of the data memory segment on a Little Endian Machine? Byte per byte, in hex.



### Exercise #1c



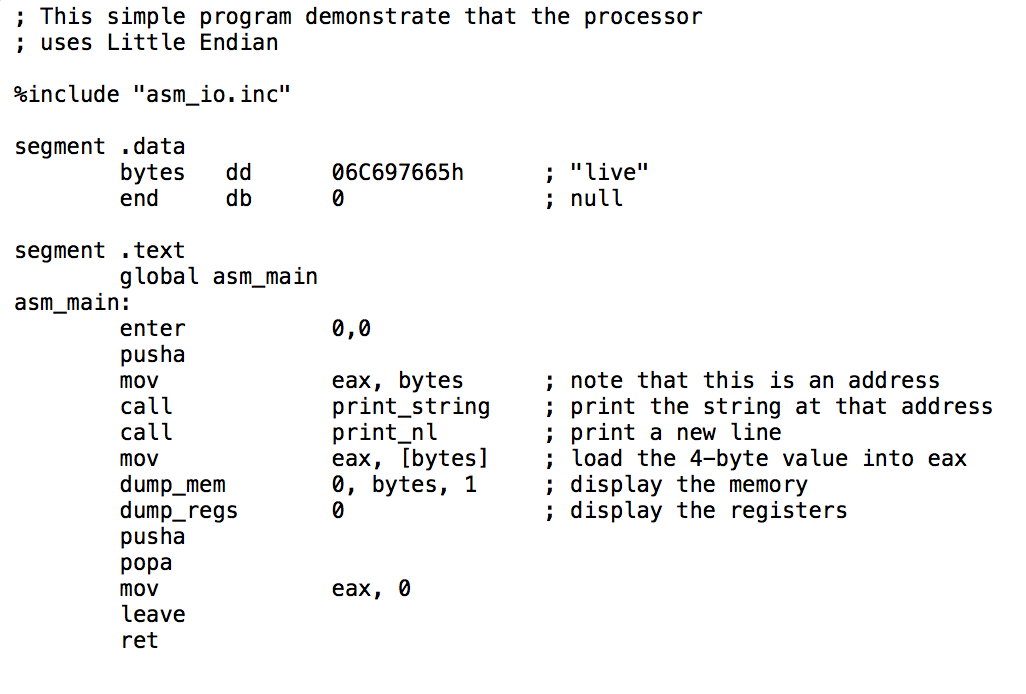
### Exercise #1d



# Part Two: Involving register indirect, or memory indirect

These exercises use a nasm library that contain useful functions from “PC Assembly Language” textbook resource; so we can “dump\_mem”, and “print\_string” for example. The library files you need are asm\_io.inc, and asm\_io.asm; I have put those files in D2L.

### Exercise #2a

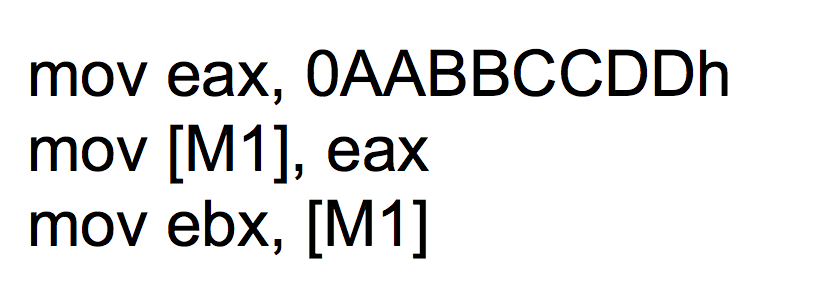


* nasm -felf32 -d ELF\_TYPE asm\_io.asm
* nasm -felf32 -d ELF\_TYPE ex2a.asm
* gcc -m32 driver.c ex2a.asm asm\_io.o
* ./a.out

### Exercise #2b

What is the in-register byte-order and the in-memory byte order?

First line involves immediate data addressing mode. No memory access is involved.



### c. Exercise #2c

What is the in-register byte-order and the in-memory byte order?

