CSCI 3310 – Address Translation Exercises

1. Consider a machine with a total memory size of 256 bytes, a page size of 16 bytes, and the (partial) page table shown below:

Page	Frame
0	2
1	6
2	11
3	9
4	5
5	0
6	4
…	…

(1a) How many possible pages can a process have?

(1b) How many bits are in a virtual address (assuming a process can access all memory)?

(1c) How many bits of the virtual address are for **p** (page number) and for **d** (page offset)?

(1d) Translate virtual address 24 to its physical memory address. Do this in two ways: using the computationally expensive non-binary method, and then using the efficient binary method.

(1e) Translate virtual address 82 to its physical memory address.

2. On most machines, each address corresponds to a specific byte of memory -- such machines are called **byte-addressable**. Some special-purpose machines, on the other hand, use a different model where each address corresponds to a **word** of memory (the 'natural' data size of the processor) -- these machines are called **word-addressable**, and on such machines, a word is the smallest addressable quantity instead of a byte. Suppose we have the same memory system described previously (256 total bytes with 16-byte pages) running on a word-addressable machine with a 32-bit word size.

(2a) – (2d) Repeat questions (1a) through (1d) given this memory architecture. Remember that addresses are now in terms of words, not bytes. Each virtual address will thus specify a **set of bytes** (comprising a word) in physical memory rather than a single byte.

(2e) Translate virtual address 13 to a set of physical bytes on this machine.