CSCI 3310 – Address Translation Exercises

1. Consider a machine using paging with a total physical memory size of 256 bytes, a page size of 16 bytes, and the partially filled in page table shown below:

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

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

(b) How many bits do we need for a virtual address so a process can access all memory?

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

(d) 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.

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

2. On most machines, each address corresponds to one byte of memory -- such machines are called **byte-addressable**. Some special-purpose machines use a different model where each address corresponds to a complete **word** of memory. These machines are called **word-addressable**, and here, 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.

(a) – (d) Repeat questions (1a) through (1d) given this memory architecture. Remember that addresses are now in terms of words, not bytes. Each virtual address thus specifies a **set of bytes** (comprising a word) in physical memory rather than one byte. Note that this means there will be fewer virtual addresses!

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