COMP 3659 – Operating Systems
Memory Management Lab, Part 1: Paging Basics (Exercises)

Instructions:

- **These exercises are to be completed on paper, on a whiteboard, etc.**
- Working at separate lab workstations (physically distanced), each student must complete each lab activity below. However, students are encouraged to form learning partnership groups of up to three members. Group members should communicate as they work and should help each other as they engage with and make sense of each activity. This can include:
    - Developing a common understanding of the purpose of each activity;
    - Articulating uncertainty and questions;
    - Sharing insights and suggestions for additional experimentation and exploration;
    - Helping each other to understand, explain, and make sense of exercises and candidate solutions; and
    - Articulating and summarizing what you have discovered or learned.
- Complete each of the following lab activities.

Lab Activities:

1. Define the terms *logical address space* and *physical address space*. As part of your answer, describe the role of a computer system's *memory management unit* (MMU).

2. Imagine a simple MMU that implements a base/limit register approach to memory protection. Explain precisely how it cooperates with the kernel such that errant processes (that try to access physical addresses outside of their designated address spaces) are terminated. Also, what information must be added to each process's PCB in order to support this scheme?

3. Consider a simple paging system with the following features:

    - 2^{14} memory locations in physical memory; each holds one byte
    - page size of 2^{8} bytes
    - 2^{4} pages in each process's logical address space

    Answer the following questions. Justify each answer.

    a. Draw a picture showing the layout / meanings of the bits in a physical address.
    b. Draw a picture showing the layout / meanings of the bits in a logical address.
    c. In what page does the logical address $\texttt{42A}_{16}$ reside? At what offset within that page?
    d. Assuming the page from (c) is mapped to frame $\texttt{3D}_{16}$, at what physical address is the logical address from (c) located? Show your answer in hex.
    e. How many frames are in physical memory? Show your answer in decimal.
    f. How many entries are in the page table? Show your answer in decimal.
    g. How many bits are in a page table entry? Show your answer in decimal.

4. Consider a simple paging system with the following features:

   - 2^{16} memory locations in physical memory; each holds one byte
   - page size of 2^{12} bytes
   - There may be *up to* 2^{2} pages in each process's logical address space

   Further, assume three processes (P_0 through P_2) are currently running:

   - P_0 requires 6051 bytes of memory for all of its code and data
   - P_1 requires 1024 bytes of memory for all of its code and data
   - P_2 requires 8192 bytes of memory for all of its code and data

   Ignoring the kernel's space requirements (for simplicity), answer the following questions:

   a. Draw a picture illustrating how all three processes' pages *might* be allocated across physical frames (you may scatter them throughout physical memory however you like)
   b. Draw the page table for each process, according to your previous answer
   c. Draw the free frame list, according to your first answer
   d. Calculate the amount of internal fragmentation in this example, as a percentage of frame size
   e. Would smaller page sizes increase or decrease the amount of internal fragmentation? Justify your answer

5. What is the role of the MMU in a simple paging system like the ones described above?

6. Paging avoids the possibility of external fragmentation. Explain why.