CSCD 340
Lab 9

Write a C program that translates virtual address to physical addresses. The program will read from a file containing virtual addresses and will translate each virtual address to its corresponding physical address.

There are 2 files required for this program.

- File 1 – setup.txt – a text file that contains in this order
  - An int that represents the total size of the virtual address space (Example: 1048576 would represent a virtual address space that is 1 MB in size or 20 bits)
  - An int that represent the size of a single page/frame (Example: 256 would be a page is 256 bytes in size or 8 bits)
  - An int that represents the total size of the physical address space (Example: 65536 would represent a physical memory space that is 64 KB in size or 16 bits)
  - You will have to do the math. In this case we have you have 4096 pages, each page is 256 bytes in size, and you have 256 frames each frame is 256 in size)

- File 2 – the virtual addresses to be translated. This file contains several base ten, 32-bit integers that represent virtual addresses.

- You may hardcode setup.txt, you must prompt for the name of File 2.

NOTE: you are guaranteed valid powers of 2 numbers and **the Happy Part of Stuland**

TO DO

1. Create a dynamic array of the size that is equal to the number of pages in the virtual address space.
   - This array will be an array of structures. The structure will be named Page Table Entry (PTE).
   - The PTE will contain
     - int the page frame it is mapped to as a decimal number
     - int the present bit

2. Create a dynamic array of the size that is equal to the number of page frames in the physical address space
   - This array will be an array of structures. The structure will be named Page Frame (PF).
   - The PF will contain
     - int the page it is mapped to

3. Read the virtual address from the file, determine the page for that address
   - If the page is mapped, the frame number is obtained from the page table entry within the page and the frame is updated with the page number.
   - If the page is not mapped (a miss) then a page fault happens and the page table entry must be loaded with a page frame number that is not currently being used. This should be linear start at zero and go to 1 then 2 then 3 etc.
   - If there is no available page frame number, then evict page 0, then page 1, then page 2, etc.

EXAMPLE

- For example page 3 is referenced as the first page
  - This reference will cause a miss
  - Page 3 will then be mapped to page frame 0
  - The present bit will be set
  - The physical address will be displayed

- Later on page 3 is referenced again
    - This reference will be a hit – pull the frame from the PTE and calculate the physical address
    - The physical address will be displayed

TEST FILE

- Provided is test.txt, which contains a limited set of ints representing virtual addresses ranging from 0 through the size of the virtual address space.

OUTPUT

- Your program is to report the following per address
    - Virtual Address: the virtual address value
    - Page Number: the page number of the virtual address
    - Page Frame Number: the page frame number
    - Physical Address: the physical address

NOTES AND FUTURE WORK:

- The number is in binary so use bitwise operations you will receive 0 points if you convert it to a character string
- I have provided a small testing file my real file has 10000 addresses in it. Make sure you test with more than my basic testing file. Also hand calculate just to be sure.

TO TURN IN

A single zip file (named your last name first letter of your first name lab9.zip) containing:

- All C files
    - ensure your program compiles and runs
    - The one that contains main will be called cscd340Lab9.c
- All input files
    - setup.txt
    - test.txt/addresses.txt or whatever you called it
- A makefile with the target of lab9

SAMPLE RUN

```
Virtual Address: 16916
Page Number: 66
Page Frame Number: 0
Physical Address: 20
========================
Virtual Address: 62493
Page Number: 244
Page Frame Number: 1
Physical Address: 285
========================
Virtual Address: 16915
Page Number: 66
Page Frame Number: 0
Physical Address: 19
========================
```