CS 3413
Assignment 7

Due Date: November 18th, 2019 at 9:30 am

ASSIGNMENT IS TO BE COMPLETED INDIVIDUALLY BY ALL STUDENTS!

Your solution is to be written in C and submitted via D2L.

This question consists of writing a program that translates logical to physical addresses for a virtual address space of size 2^{32} bytes. Your program will read from stdin a list of logical addresses with access type (r – read or w – write), translate each logical address to a physical address and then print the physical address that would be accessed in memory. However, your simulation can only have n pages loaded in memory at any given time! The goal is to simulate the steps involved in translating logical to physical addresses.

Design your simulation with the following parameters:

- A page is 2^{12} = 4 kilobytes. (Note, this means a frame is also 2^{12} in size).
- Your process has been allocated *n* frames. *n* is a parameter to your program on the command line.

For the address translation you are to use the multiple levels of page directory, page table and offset for mapping (page directory is 10 bits, page table is 10 bits, page offset is 12 bits)

Your program should print for each logical address read:

```
logical address -> physical address
```

At the end of simulation you will print the total number of page faults that occur and the number of dirty pages that are paged out.

Notes:

- Example for format of the input is:

```
r 12345678
w 87654321
```

- Running your program with a sample file can be done as:

```
./a.out 10 < sample_logical_addr
```

- For this question you are to use the following page replacement policy:
  1. If an empty frame is available – you use it.
  2. If clean frames exist, then you choose one based on the Least Recently Used (LRU) page replacement policy.
  3. If only dirty pages exist, then you choose one based on the Least Recently Used page replacement policy.

- You do NOT have to worry about saving pages that are written too! You just need to keep track of whether the page is clean or dirty. Every time the page is paged-in, it is clean. But as soon as you write to it, it is dirty.
- Be aware of data type sizes and signed vs unsigned!