CSL333 | Assignment-5 | Due 10/Nov/2018 11:59 PM | 100 points

- Important instructions for coding submission are here: https://goo.gl/IMWvdF
- Grading scheme to be followed is available here: https://goo.gl/52D82g
- Assignment description may be underspecified to allow some room for exploration and creativity.
- Your submission should be packaged as a zip file named **exactly** in this format: CSL333-[your entry no.]-[assignment no.].zip.

We need to write a module in Python which can simulate the virtual memory with demand paging. This module should implement the paging mechanism using inverted page tables. The reference input addresses should be input/printed in decimal. This module should expose a function that has the following signature:

```
simulate(virtual_address_size, page_size, ram_size, page_rep_algo, ref_addresses)
```

Input to the function:

1. Size of virtual addresses in bits (e.g. 32 bits, 64 bits, etc.).
2. A set of referenced addresses.
3. RAM size in MB
4. Page size in kB. Assume physical memory frame size equal to page size.
5. Choice of page replacement algorithm to use for a given run of simulation. Choices include:
    a. FIFO
    b. LFU
    c. MFU

Expected output:

1. For each address in the input reference addresses set, compute and print:
    a. Page number and offset corresponding to the input address.
    b. Frame number and offset corresponding to the input address.
    c. Page fault status (yes/no).
2. Rate of page faults. It can be computed as the fraction of referenced addresses for which page fault occurred.

You should supply a test case to verify the correctness of your program. Assume that the referenced addresses:

1. are randomly generated
2. belong to same process
3. none of these addresses are in memory at the beginning