San José State University
Department of Computer Engineering

CMPE 142
Operating Systems
Section 1

Spring 2021
Instructor: Ron Mak

Assignment #6

Assigned: Friday, March 12
Due: Friday, March 26 at 11:59 PM
Team assignment, 120 points

Page replacement algorithms

In this assignment, your team will simulate page replacement algorithms operating systems use to implement virtual memory, **first-in first-out (FIFO)**, **least recently used (LRU)**, **least frequently used (LFU)**, **most frequently used (MFU)**, **random pick**, and the theoretical **optimal**. You may write your programs in either C or C++.

Reference string

You are simulating running a process that consists of **eight pages** numbered 0 through 7. Your simulated physical memory has **three page frames**, and eight page frames are always available on the backing store. (You are not simulating what happens on the disk for this assignment.)

Generate a string of 100 page references. Due to locality of reference, after referencing a page i, there should be a 60% probability that the next reference is page i-1, i, or i+1.

TIP: Given a page reference i, to compute the next page reference, generate a random integer r from 0 through 9:
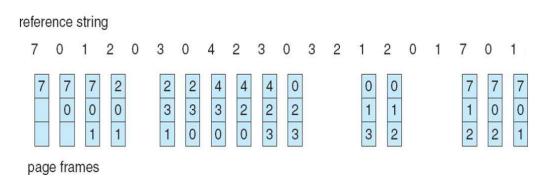
- If r is 0 or 1, then the next reference is i-1.
- If r is 2 or 3, then the next reference is i again.
- If r is 4 or 5, then the next reference is i+1.
- If r is 6, 7, 8, or 9, then randomly make the next reference other than i-1, i, or i+1.
- The value of i should wrap around from 7 to 0.

Run the algorithms

Run all six algorithms, each using the <u>same</u> reference string. For each algorithm, compute the **hit ratio** of pages already in memory (pages in the three page frames), and **count page swaps** (a page evicted and another page brought in).

When the simulated process starts, the page frames are empty, so none of the process's pages are in memory. For each subsequent reference in the generated string of page references, print the page numbers of the <u>pages in memory</u> and (if any) which page needed to be <u>paged in</u> and which page was <u>evicted</u>.

Use the examples in the lecture slides as a guide. It will be easier to print the page references down the page and the corresponding page frames horizontally. Therefore, instead of:

reference string

page frames

Print:

```
Ref     Frames     Action
 7      7
 0      7 0
 1      7 0 1
 2      2 0 1       7 evicted, 2 entered
 0
 3      2 3 1       0 evicted, 3 entered
etc.
```

<u>Repeat five times</u> with a different reference string each time to get averages for each algorithm.

What to submit

Submit the following to Canvas, **Assignment #6: Page Replacement Algorithms**.

- Source files (either C or C++) of your program.
- A text file of your program's output.

Rubric

Your submission will be graded according to these criteria:

Criteria	Max points
First-in first-out (FIFO)	20
Least recently used (LRU)	20
Least frequently used (LFU)	20
Most frequently used (MFU)	20
Random pick	20
Optimal	20