Results

GLOBAL FIFO LRU

```
Process:A, Effective Access Time:184.758
Process:A, Page Fault Rate:0.892
Process:B, Effective Access Time:183.709
Process:B, Page Fault Rate:0.856
```

GLOBAL FIFO RANDOM

```
Process:A, Effective Access Time:184.534
Process:A, Page Fault Rate:0.900
Process:B, Effective Access Time:182.373
Process:B, Page Fault Rate:0.904
```

GLOBAL CLOCK LRU

```
Process:A, Effective Access Time:184.758
Process:A, Page Fault Rate:0.730
Process:B, Effective Access Time:181.980
Process:B, Page Fault Rate:0.764
```

GLOBAL CLOCK RANDOM

```
Process:A, Effective Access Time:184.309
Process:A, Page Fault Rate:0.743
Process:B, Effective Access Time:182.568
Process:B, Page Fault Rate:0.746
```

LOCAL FIFO LRU

```
Process:A, Effective Access Time:184.758
Process:A, Page Fault Rate:0.937
Process:B, Effective Access Time:183.709
Process:B, Page Fault Rate:0.894
```

LOCAL FIFO RANDOM

```
Process:A, Effective Access Time:185.200
Process:A, Page Fault Rate:0.929
Process:B, Effective Access Time:183.709
Process:B, Page Fault Rate:0.886
```

LOCAL CLOCK LRU

```
Process:A, Effective Access Time:184.082
Process:A, Page Fault Rate:0.750
Process:B, Effective Access Time:182.568
Process:B, Page Fault Rate:0.667
```

LOCAL CLOCK RANDOM

```
Process:A, Effective Access Time:184.534
Process:A, Page Fault Rate:0.736
Process:B, Effective Access Time:182.373
Process:B, Page Fault Rate:0.672
```

Comparison

GLOBAL vs LOCAL:

GLOBAL method tends to perform faster than LOCAL does, since GLOBAL method doesn't have to check if the new page and evicted page are of same process. However, in reality, we mostly choose LOCAL method so that we could control how much memory a process consumes.(we don't see the difference from the experiment because given processes were too few)

FIFO vs CLOCK:

From my experiment, it is clearly that page-fault-rate of CLOCK was way less than that of FIFO. In most cases, there are pages that are used more frequently. By resetting the reference bit, we can keep the more frequent pages in the memory. Yet, FIFO should perform faster than CLOCK, since it doesn't have to check anything. (we don't see the difference from the experiment because given processes were too few)

RANDOM vs LRU:

As discussed above, it is obvious that RANDOM should run faster than LRU because it need not check anything. But when it comes to page-fault-rate, LRU tends to obtain less than RANDOM does. It is resulted from the fact that some pages are more likely to be requested more than once. Therefore by keeping those frequent pages in memory can we decrease the number of times that page faults occur.