## The tables prepared in Task 3:

simpleloop.c size = 50

| | fifo | lru | clock | rand |
|---|---|---|---|---|
| Hit rate | 71.0639 | 72.8468 | 72.7592 | 70.7034 |
| Hit count | 7294 | 7477 | 7468 | 7257 |
| Miss count | 2970 | 2787 | 2796 | 3007 |
| Overall eviction count | 2920 | 2737 | 2746 | 2957 |
| Clean eviction count | 203 | 90 | 97 | 251 |
| Dirty eviction count | 2717 | 2647 | 2649 | 2706 |

simpleloop.c size = 100

| | fifo | lru | clock | rand |
|---|---|---|---|---|
| Hit rate | 73.1099 | 73.8406 | 73.8016 | 73.0417 |
| Hit count | 7504 | 7579 | 7575 | 7497 |
| Miss count | 2760 | 2685 | 2689 | 2767 |
| Overall eviction count | 2660 | 2585 | 2589 | 2667 |
| Clean eviction count | 46 | 1 | 4 | 55 |
| Dirty eviction count | 2614 | 2584 | 2585 | 2612 |

matmul.c size = 50

| | fifo | lru | clock | rand |
|---|---|---|---|---|
| Hit rate | 60.9664 | 63.9458 | 63.9447 | 65.5416 |
| Hit count | 1760685 | 1846728 | 1846696 | 1892816 |
| Miss count | 1127275 | 1041232 | 1041264 | 995144 |
| Overall eviction count | 1127225 | 1041182 | 1041214 | 995094 |
| Clean eviction count | 1083231 | 1040070 | 1040098 | 955961 |
| Dirty eviction count | 43994 | 1112 | 1116 | 39133 |

matmul.c size = 100

| | fifo | lru | clock | rand |
|---|---|---|---|---|
| Hit rate | 62.4805 | 65.1499 | 63.9500 | 88.8185 |
| Hit count | 1804411 | 1881504 | 1846850 | 2565043 |
| Miss count | 1083549 | 1006456 | 1041110 | 322917 |
| Overall eviction count | 1083449 | 1006356 | 1041010 | 322817 |
| Clean eviction count | 1061226 | 1005275 | 1039927 | 315339 |
| Dirty eviction count | 22223 | 1081 | 1083 | 7478 |

## Comparing the various algorithms:

From the results we get in the tables, we may find that the hit rate from highest to lowest of the various algorithms is always OTP, LRU, Clock, FIFO, and Rand. For OTP, it is the optimal algorithm, and it is always finding and replace the page that is not used in the future, so the hit rate is the highest. The LRU replace the page used least often, and Clock replace the first page whose ref bit is 0. The rate of LRU and Clock is close, while LRU is always a little higher. The hit rate of FIFO is lower, because it just replace the page by the order of 'First in first out', which doesn't consider later used or not. The Rand has the worst result since it just choose page randomly. We may also notice that, the hit rate also increases slightly as the memory size increases with the same program and algorithm. For the running time, it first depends on the memory size. Also for each algorithm, the runtime order from fastest to slowest is Rand, FIFO, Clock, LRU, and OTP, which is the inverse order of hit rate. This is because Rand and FIFO just choose the page randomly or oldest page, so they are fast. For Clock, LRU and OTP, they all need to store some information of each page, like time, counter, and ref bit. In order to make an efficient choice, they need to trace each page according the stored information, so these three algorithms takes more time and spaces.

## My three traces:

tr-trace1.ref  size = 8

| | fifo | lru | clock | opt(by hand) |
|---|---|---|---|---|
| Hit rate | 47.0588 | 55.8824 | 50.0000 | 67.6471 |
| Hit count | 16 | 19 | 17 | 23 |
| Miss count | 18 | 15 | 17 | 11 |
| Overall eviction count | 10 | 7 | 9 | |
| Clean eviction count | 0 | 0 | 0 | |
| Dirty eviction count | 10 | 7 | 9 | |

tr-trace2.ref  size = 8

| | fifo | lru | clock |
|---|---|---|---|
| Hit rate | 80.0000 | 80.0000 | 80.0000 |
| Hit count | 32 | 32 | 32 |
| Miss count | 8 | 8 | 8 |
| Overall eviction count | 0 | 0 | 0 |
| Clean eviction count | 0 | 0 | 0 |
| Dirty eviction count | 0 | 0 | 0 |

tr-trace3.ref  size = 8

| | fifo | lru | clock |
|---|---|---|---|
| Hit rate | 0 | 0 | 0 |
| Hit count | 0 | 0 | 0 |
| Miss count | 36 | 36 | 36 |
| Overall eviction count | 28 | 28 | 28 |
| Clean eviction count | 19 | 19 | 19 |
| Dirty eviction count | 9 | 9 | 9 |