Task 3

1. page-simpleloop.ref

./sim -f page-simpleloop.ref -m 50/100 -s 2518 -a *

Memory size: 50

	hit rate	hit count	miss count	overall eviction	clean eviction	dirty evection
FIFO	22.7206	770	2619	2569	45	2524
clock	25.3762	860	2529	2479	0	2479
LRU	25.4352	862	2527	2477	0	2477
MRU	1.4163	48	3341	3291	410	2881

Memory size: 100

	hit rate	hit count	miss count	overall eviction	clean eviction	dirty evection
FIFO	24.0189	814	2575	2475	23	2452
clock	25.3172	858	2531	2431	1	2430
LRU	25.4352	862	2527	2427	0	2427
MRU	1.9475	66	3323	3223	401	2822

2. page-repeatloop.ref

./sim -f page-repeatloop.ref -m 50/100 -s 99 -a *

Memory size: 50

	hit rate	hit count	miss count	overall eviction	clean eviction	dirty evection
FIFO	33.4507	190	378	328	241	87
clock	34.3310	195	373	323	236	87
LRU	34.5070	196	372	322	236	86
MRU	49.2958	280	288	238	164	74

Memory size: 100

	hit rate	hit count	miss count	overall eviction	clean eviction	dirty evection
FIFO	82.7465	470	98	0	0	0
clock	82.7465	470	98	0	0	0
LRU	82.7465	470	98	0	0	0
MRU	82.7465	470	98	0	0	0

3. page-blocked.ref

./sim -f page-blocked.ref -m 50/100 -s 993 -a *

Memory size: 50

	hit rate	hit count	miss count	overall eviction	clean eviction	dirty evection
FIFO	99.6911	1939658	6011	5961	4946	1015
clock	99.7491	1940787	4882	4832	3848	984
LRU	99.7477	1940760	4909	4859	3875	984
MRU	13.1380	255622	1690047	1689997	1632627	57370

Memory size: 100

	hit rate	hit count	miss count	overall eviction	clean eviction	dirty evection
FIFO	99.7881	1941547	4122	4022	3025	997
clock	99.8015	1941806	3863	3763	2792	971
LRU	99.8141	1942052	3617	3517	2569	948
MRU	21.6730	421684	1523985	1523885	1471265	52620

4. page-matmul.ref

./sim -f page-matmul.ref -m 50/100 -s 993 -a *

Memory size: 50

	hit rate	hit count	miss count	overall eviction	clean eviction	dirty evection
FIFO	52.4492	1217013	1103352	1103302	1102309	993
clock	55.1394	1279436	1040929	1040879	1039918	961
LRU	55.1396	1279440	1040925	1040875	1039914	961
MRU	15.9283	369595	1950770	1950720	1931220	19500

Memory size: 100

	hit rate	hit count	miss count	overall eviction	clean eviction	dirty evection
FIFO	53.7949	1248239	1072126	1072026	1071052	974
clock	55.1721	1280194	1040171	1040071	1039111	960
LRU	56.6329	1314090	1006275	1006175	1005215	960
MRU	22.3449	518483	1801882	1801782	1783832	17950

Comparisons of the four algorithms:

Page-simpleloop.ref has the lowest overall hit rate for all algorithms because the loop accesses different page frames each time, and it is more expensive than other programs. Page-blocked.ref has the highest hit rate for all programs except for the MRU because the algorithms take advantage of the spatial locality of the pages in page-blocked.ref. Page-simpleloop.ref causes the four algorithms to have more dirty evictions than other programs, this is because there are more instructions that modify the pages. Looking at the performance of the four algorithms across different programs, LRU and clock have almost identical hit rates. This is because clock is an approximation of LRU with less computational overhead for choosing the next page to evict. FIFO is a little bit worse than LRU and clock for these traces, because FIFO is a more naive algorithm that always evicts the oldest page regardless of how often or recently this page has been referenced. MRU is usually worse than all other algorithms in all the programs except for page-repeatloop. This is probably because the traces in page-repeatloop tend to reference the least recently used pages. However, most code has temporal locality when it comes to memory accesses. Thus, the performance of different algorithms depends largely on the traces, but for a random trace file, clock tends to perform better than the other three algorithms.

In addition, when memory size increases from 50 to 100, all algorithms tend to have similar performances except for MRU. MRU has a relatively significant increase in hit rate when the memory size increases. This is probably because when the memory size increases, we are able to keep more recently used pages. Also, when memory size increases from 50 to 100, page-repeatloop tends to cause all algorithms to have a significant increase in performance. This is probably because although traces in page-repeatloop accesses a variety of different pages, the range is not very large.

Trace 1

./sim -f trace1.ref -m 8 -s 32 -a *

	hit rate	hit count	miss count
FIFO	37.5	12	20
clock	6.25	2	30
LRU	3.125	1	31
Optimal	43.75	14	18

Trace 2

./sim -f trace2.ref -m 8 -s 32 -a *

	hit rate	hit count	miss count
FIFO	43.75	14	18
clock	43.75	14	18
LRU	65.625	21	11
Optimal	65.625	21	11

Trace 3

./sim -f trace3.ref -m 8 -s 32 -a *

	hit rate	hit count	miss count
FIFO	0	0	31
clock	0	0	31
LRU	0	0	31
Optimal	64.516	20	11