simpleloop

	OPT	LRU	CLOCK	FIFO	RAND		50
Hit rate	74.8117	73.814	73.7481	72.0162	71.6303		
Hit count	7948	7842	7835	7651	7610		
Miss count	2676	2782	2789	2873	3014		
Overall eviction count	2626	2732	2739	2923	2964		
Clean eviction count	26	87	90	205	259		
Dirty eviction count	2600	2645	2649	2718	2705		

	OPT	LRU	CLOCK	FIFO	RAND	100
Hit rate	75.16	74.6894	74.6611	74.0211	73.9081	
Hit count	7985	7935	7932	7864	7852	
Miss count	2639	2689	2692	2760	2772	
Overall eviction count	2539	2589	2592	2660	2672	
Clean eviction count	0	2	3	44	60	
Dirty eviction count	2539	2587	2589	2616	2612	

	OPT	LRU	CLOCK	FIFO	RAND	150
Hit rate	75.16	74.7176	74.7082	74.3976	74.3223	
Hit count	7985	7938	7937	7904	7896	
Miss count	2639	2686	2687	2720	2728	
Overall eviction count	2489	2536	2537	2570	2578	
Clean eviction count	0	0	0	16	20	
Dirty eviction count	2489	2536	2537	2554	2558	

	OPT	LRU	CLOCK	FIFO	RAND	200
Hit rate	75.16	74.7176	74.7176	74.4729	74.4917	
Hit count	7985	7938	7938	7912	7914	
Miss count	2639	2686	2686	2712	2710	
Overall eviction count	2439	2486	2486	2512	2510	
Clean eviction count	0	0	0	12	16	
Dirty eviction count	2439	2486	2486	2500	2494	

matmul

	OPT	LRU	CLOCK	FIFO	RAND		50
Hit rate	79.6598	63.9487	63.9486	60.97	65.5515	60.97	
Hit count	2300702	1846942	1846938	1760911	1893232		
Miss count	587458	1041218	1041222	1127249	994928		
Overall eviction count	587408	1041168	1041172	1127199	994878		
Clean eviction count	586322	1040063	1040065	1083211	955867		
Dirty eviction count	1086	1105	1107	43988	39011		

	OPT	LRU	CLOCK	FIFO	RAND	100

Hit rate	96.787	65.1524	65.3138	62.4832	88.7987
Hit count	2795364	1881706	1886366	1804615	2564648
Miss count	92796	1006454	1001794	1083545	323512
Overall eviction count	92696	1006354	1001694	1083445	323412
Clean eviction count	91612	1005274	1000612	1061221	315976
Dirty eviction count	1084	1080	1082	22224	7436

	OPT	LRU	CLOCK	FIFO	RAND	150
Hit rate	99.0785	98.8614	98.6045	98.8087	96.6819	
Hit count	2861547	2855275	2847855	2853752	2792328	
Miss count	26613	32885	40305	34408	95832	
Overall eviction count	26463	32735	40155	34258	95682	
Clean eviction count	25379	31656	39075	32943	93388	
Dirty eviction count	1084	1079	1080	1315	2294	

	OPT	LRU	CLOCK	FIFO	RAND	200
Hit rate	99.333	98.8618	98.8613	98.8267	98.0458	
Hit count	2868897	2855287	2855273	2854273	2831720	
Miss count	19263	32873	32887	33887	56440	
Overall eviction count	19063	32673	32687	33687	56240	
Clean eviction count	17979	31594	31608	32433	54605	
Dirty eviction count	1084	1079	1079	1254	1635	

tr-blacked.ref

	OPT	LRU	CLOCK	FIFO	RAND	50
Hit rate	99.8472	99.7879s	99.763	99.7346	99.6605	
Hit count	2414561	2413127	2412525	2411839	2410045	
Miss count	3695	5129	5731	6417	8211	
Overall eviction count	3645	5079	5681	6367	8161	
Clean eviction count	2560	2745	3247	4099	5660	
Dirty eviction count	1085	2334	2434	2268	2501	

	OPT	LRU	CLOCK	FIFO	RAND	100
Hit rate	99.8761	99.8435	99.829	99.822	99.7839	
Hit count	2415259	2414472	2414120	2413952	2413029	
Miss count	2997	3784	4136	4304	5227	
Overall eviction count	2897	3684	4036	4204	5127	
Clean eviction count	1825	2603	2610	2726	3384	
Dirty eviction count	1072	1081	1426	1478	1743	

	OPT	LRU	CLOCK	FIFO	RAND	150
Hit rate	99.8957	99.8442	99.8437	99.826	99.8201	
Hit count	2415734	2414488	2414476	2414048	2413905	
Miss count	2522	3768	3780	4208	4351	

Overall eviction count	2372	3618	3630	4058	4201	
Clean eviction count	1297	2558	2570	2636	2739	
Dirty eviction count	1075	1060	1060	1422	1462	
	OPT	LRU	CLOCK	FIFO	RAND	200
Hit rate	99.906	99.8472	99.8676	99.8692	99.8426	
Hit count	2415984	2414561	2415055	2415094	2414450	
Miss count	2272	3695	3201	3162	3806	
Overall eviction count	2072	3495	3001	2962	3606	
Clean eviction count	1007	2435	1941	1865	2281	
Dirty eviction count	1065	1060	1060	1097	1325	
						tr-map.ref
	OPT	LRU	CLOCK	FIFO	RAND	50
Hit rate	99.7197	99.6256	99.6278	99.492	99.4504	
Hit count	45544	45501	45502	45440	45421	
Miss count	128	171	170	232	251	
Overall eviction count	81	121	120	182	201	
Clean eviction count	7	28	24	56	73	
Dirty eviction count	74	93	96	126	128	
	OPT	LRU	CLOCK	FIFO	RAND	
Hit rate	99.7438	99.7416	99.7307	99.7132	99.7263	100
Hit count	45555	45554	45549	45541	45547	
Miss count	117	118	123	131	125	
Overall eviction count	17	18	23	31	25	
Clean eviction count	0	0	0	0	0	
Dirty eviction count	17	18	23	31	25	
	OPT	LRU	CLOCK	FIFO	RAND	150
Hit rate	99.7438	99.7438	99.7438	99.7438	99.7438	
Hit count	45555	45555	45555	45555	45555	
Miss count	117	117	117	117	117	
Overall eviction count	0	0	0	0	0	
Clean eviction count	0	0	0	0	0	
Dirty eviction count	0	0	0	0	0	
	OPT	LRU	CLOCK	FIFO	RAND	200
Hit rate	99.7438	99.7438	99.7438	99.7438	99.7438	
Hit count	45555	45555	45555	45555	45555	
Miss count	117	117	117	117	117	
Overall eviction count	0	0	0	0	0	
Clean eviction count	0	0	0	0	0	
Dirty eviction count	0	0	0	0	0	

Question 2:

Explanation:

Observations within the same program :
-When physical memory size is large(like 200 frames), OPT and clock algorithm generally generate very high hit rate, and OPT's will be just slightly higher than clock's.
-When physical memory size if large(like 200 frames), OPT and clock have significant higher hitting rates than LRU and FIFO
-When physical memory size is small(like 50 frames), OPT, clock, FIFO, LRU will have the moderate strong and similar hitting rates(60% to 70% +)
-Generally speaking, the physical memory size has very small effect on hitting rate for FIFO and LRU algorithm.
-Generally speaking, the physical memory size has relatively large effect on hitting rate for OPT and clock algorithm.
-For the same program/process ,the smaller the physical memory size is, the lower the hitting rate will each page replacement algorithm generate
Reasons for the above observations :
OPT is the best algorithm for sure, since it rationally analyzes the least referenced physical frame, and it assumes that it knows about the future situations of paging, thus selecting the longest-time-used-again one to be evicted, while FIFO is just generally removing the oldest one, LRU is replacing the frame that is used the least in the past(past is really a bad prediction for the future), and clock is basically looping over frames as circular buffer, replacing the frame once finding it unreferenced(indicated by reference bit of the corresponding pte_t).
Since size of VIRTUAL ADDRESS spaces for the same program is generally fixed, thus the larger the size of physical frames => the less likely that the process will need some physical frames swapped into the disk => less page faults => higher hitting rate
Explanations of four Programs :
By analyzing the simple loop program, none of the algorithm has high hitting rate, since the program is continuously 'appending' value to an array(or continuously allocating new memory space) , and because every increment is 128 * sizeof(double) which is 128 * 8, thus it is very unlikely that the program will reference the same memory address more than one time, and it requires continuously new physical frames(need to get the unused one swapped in), which causes page faults => low hitting rates.
By analyzing the matmul program, which is the computing multiplication of matrix. It is convinced that both OPT and Clock page replacement algorithm will have the high hitting rate. Since matrix multiplication involves reusing the same variable again and again(based on Math common sense as well as reading code in the program). Thus no matter predicting the likelihood of page's being referenced by analyzing the future or simply analyzing the reference bit, it BOTH benefit choosing the correct victim to evict.
=> OPT and clock algorithm have much higher hitting rate than the other two(much fewer page faults)
By analyzing the blocked program, the four algorithm both generate high hitting rate. This is because that the programs involve matrix adding and multiplication, thus it is very likely that 1, the recently 'come' memory will will be referenced again 2, the referenced memory is likely to be referenced again 3, the least recently used is actually the most likely to be referenced in the future 4, It is no doubt that OPT is the best algorithm. Thus, all four algorithm will have high hitting rate.
By analyzing the map program, since 1, the program is relatively simple and contain few lines of code 2, the program is continuously accessing or checking one(s) variables, like generic integer variables that store int value or conditional variables. Then, it may seldom trigger page replacement approach, which means => no page faults => high hitting rate

Question 3:

The hit count for each trace file increases a little bit or merely stays the same after each memory size increase. This means that, the accuracy of finding the best evicted page increases when memorize size goes up. It's because a larger memory size means we have a longer list of reference to check which page has the longest "wait time" since the last usage. And this implies we have a better comparison before evicting a page. And this makes the hit rate increases, at least staying the same. During processing the tr-matmul.ref file, once the mem size goes from 100 to 150, the hit rate raised up 30+%. This demonstrated the statement I made above.