Problem 1. Consider a virtual memory system that uses a single-level page map to translate virtual addresses into physical addresses. Each of the questions below asks you to consider what happens when one of the design parameters of the original system is changed.

increases by 1 bit. for addressing twice the memory addresses.

A. If the physical memory size (in bytes) is doubled, how does the number of bits in each entry of the page table change?

Does not change.

B. If the physical memory size (in bytes) is doubled, how does the number of entries in the page map change?

Does not change.

C. If the virtual memory size (in bytes) is doubled, how does the number of bits in each entry of the page table change?

Doubled each entry one address.

D. If the virtual memory size (in bytes) is doubled, how does the number of entries in the page map change?

One bit less. # physical pages halved

E. If the page size (in bytes) is doubled, how does the number of bits in each entry of the page table change?

halved # virtual pages halved.

F. If the page size (in bytes) is doubled, how does the number of entries in the page map change?

G. The following table shows the first 8 entries in the page map. Recall that the valid bit is 1 if the page is resident in physical memory and 0 if the page is on disk or hasn't been allocated.

3956 = 111101110100

V# offset

101101110100

$3956 - 2^{10} = 2932$

Virtual page	Valid bit	Physical page
0	0	7
1	1	9
2	0	3
3	1	2
4	1	5
5	0	5
6	0	4
7	1	1

■ If there are 1024 (2^{10}) bytes per page, what is the physical address corresponding to the decimal virtual address 3956? 2932

Problem 2. Consider two possible page-replacement strategies: LRU (the least recently used page is replaced) and FIFO (the page that has been in the memory longest is replaced). The merit of a page-replacement strategy is judged by its hit ratio.

Assume that, after space has been reserved for the page table, the interrupt service routines, and the operating-system kernel, there is only sufficient room left in the main memory for *four* user-program pages. Assume also that initially virtual pages 1, 2, 3, and 4 of the user program are brought into physical memory in that order. 4 3 2 1

A. For each of the two strategies, what pages will be in the memory at the end of the following sequence of virtual page accesses? Read the sequence from left to right: (6, 3, 2, 8, 4).

B. Which (if either) replacement strategy will work best when the machine accesses pages in the following (stack) order: (3, 4, 5, 6, 7, 6, 5, 4, 3, 4, 5, 6, 7, 6, ...)? LRU

C. Which (if either) replacement strategy will work best when the machine accesses pages in the following (repeated sequence) order: (3, 4, 5, 6, 7, 3, 4, 5, 6, 7, ...). both equally bad

D. Which (if either) replacement ~~strategy~~ will work best when the machine accesses pages in a randomly selected order, such as (3, 4, 2, 8, 7, 2, 5, 6, 3, 4, 8, ...). neither is better

A:

LRU		FIFO	
4321		4321	
6432	R	6432	R
3642	O	6432	
2364	O	6432	
8236	R	8693	R
4823	R	8643	

R— replace O— order changed.

B:

LRU		FIFO	
6543		6543	
7654	R	7654	R
6754	∅	7654	
5674		7654	
4567		7654	
3456	R	3765	R
4356		4376	R
5436		5437	R
6543		6543	R

2R out of 8 5R out of 8.

C:

LRU		FIFO	
6543		6543	
7654	R	7654	R
3765	R	3765	R
4376	R	4376	R
5437	R	5437	R
6543	R	6543	R

5R out of 5 5R out of 5