1. Answer:

a) A frame has the same size as a page, 2^{10} bytes.

b) There is one entry for each page in the logical address space. Therefore, there are 2^{16} entries.

c) The number of bytes in the logical address space is (2^{16} pages) × (2^{10} bytes/page) = 2^{26} bytes.

d) The number of frames needed to store the largest page table is (2^{16} entries) × (2^{2} bytes/entry) / 2^{10} bytes/frame = 2^{8} frames. = no. of entries x size of entries / size of frame

2. Answer:

a) 660+198=858

b) 222+156=378

c) Invalid address because the length of segment 1 is 422 <530.

3. Answer:

a) 5 mod 1024 = 379 -> 5, 379

- Relative address is 5499 = 5 × 1024 + 379, i.e., virtual address is 5, 379
- Virtual address in binary is 0000 0001 0101 0111 1011 from table:
- Map to frame number 2 20 bits 5 379 page number = 5 -> frame number = 2
- The physical address is 0000 1001 0111 1011

b) 16 bits 2 379 2221 mod 1024 = 173 -> 2, 173

- Relative address is 2221 = 2 × 1024 + 173, i.e., virtual address is 2, 173
- Virtual address in binary is 0000 0000 1000 1010 1101
- The page has not been loaded into memory yet, resulting in a page fault

from table:

page number = 2 -> no frame number

4. Answer:

a) OPT: no 7 no 1 0 -> 4 frame no 4

7	0	1	2	0	3	0	4	2	3	0	3	2
7	7	7	2	2	2	2	2	2	2	2	2	2
	0	0	0	0	0	0	4	4	4	0	0	0
		1	1	1	3	3	3	3	3	3	3	3
F	F	F	F		F		F			F		

Number of page faults=7

b) FIFO:

7	0	1	2	0	3	0	4	2	3	0	3	2
7	7	7	2	2	2	2	4	4	4	0	0	0
	0	0	0	0	3	3	3	2	2	2	2	2
		1	1	1	1	0	0	0	3	3	3	3
F	F	F	F		F	F	F	F	F	F		

Number of page faults=10

c) LRU: not last two

7	0	1	2	0	3	0	4	2	3	0	3	2
7	7	7	2	2	2	2	4	4	4	0	0	0
	0	0	0	0	0	0	0	0	3	3	3	3
		1	1	1	3	3	3	2	2	2	2	2
F	F	F	F		F		F	F	F	F		

Number of page faults=9

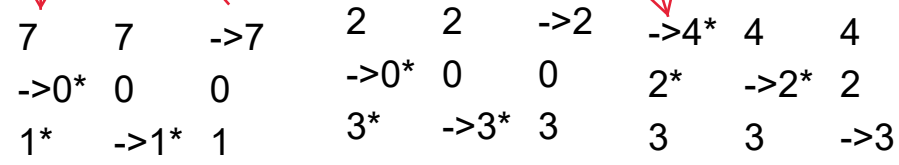
not 0 or 1

not 0 or 3

d) Clock:

7	0	1	2	0	3	0	4	2	3	0	3	2
7*	7*	→7*	2*	2*	→2*	→2*	4*	4*	4*	→4	3*	3*
→	0*	0*	→0	→0*	0	0*	→0	2*	2*	2	→2	→2*
	→	1*	1	1	3*	3*	3	→3	→3*	0*	0*	0*
F	F	F	F		F		F	F		F	F	

Number of page faults=9



7	7	->7	2	2	->2	->4*	4	4				
->0*	0	0	->0*	0	0	2*	->2*	2				
1*	->1*	1	3*	->3*	3	3	3	->3				

swap -> arrow progress by 1

hit by arrow -> remove star

no star -> swapable

recent use -> add star

Self-test

1. B
2. C
3. C
4. B
5. A
6. C