4. (a) A computer has 16 pages of virtual address space, but only 4 page frames. Initially, memory is empty. A program references the virtual pages in the following order: **0, 1, 2, 1, 2, 4, 3, 6, 0, 5, 7, 8, 7, 8, 2, 1, 3, 7, 3, and 4**.

For each memory reference, write the virtual page stored in each page frame under the **Least Recently Used (LRU)** and **First-In-First-Out (FIFO)** page replacement policies (clearly show the page frames in memory for each reference).

[10 Marks]

LRU

0	1	2	1	2
0	0	0	0	0
	1	1	1	1
		2	2	2

4	3	6	0	5
0	3	3	3	3
1	1	6	6	6
2	2	2	0	0
4	4	4	4	5

7	8	7	8	2
7	7	7	7	7
6	8	8	8	8
0	0	0	0	2
5	5	5	5	5

1	3	7	3	4
7	3	3	3	3
8	8	7	7	7
2	2	2	2	4
1	1	1	1	1

FIFO

0	1	2	1	2
0	0	0	0	0
	1	1	1	1
		2	2	2

4	3	6	0	5
0	3	3	3	3
1	1	6	6	6
2	2	2	0	0
4	4	4	4	5

7	8	7	8	2
7	7	7	7	7
6	8	8	8	8
0	0	0	0	2
5	5	5	5	5

1	3	7	3	4
7	3	3	3	3
8	8	7	7	7
2	2	2	2	4
1	1	1	1	1

4. (a) A computer has 16 pages of virtual address space, but only 4 page frames. Initially, memory is empty. A program references the virtual pages in the following order: **0, 1, 2, 2, 1, 3, 4, 5, 0, 6, 7, 8, 8, 7, 1, 2, 7, 3, 4, 3**.

For each memory reference, write the virtual page stored in each page frame under the **Most Recently Used (MRU)** and **First-In-First-Out (FIFO)** page replacement policies (clearly show the page frames in memory for each reference).

[10 Marks]

$\underbrace{\text{MRU - Most Recently Used}}$

0	1	2	2	1	3	4	5	0	6
0	0	0	0	0	0	0	0	0	6
	1	1	1	1	1	1	1	1	1
		2	2	2	2	2	2	2	2
					3	4	5	5	5

7	8	8	7	1	2	7	3	4	3
7	8	8	7	7	7	7	3	4	3
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
5	5	5	5	5	5	5	5	5	5

FIFO - First-In-First-Out

0	1	2	2	1	3	4	5	0	6
0	0	0	0	0	0	4	4	4	4
	1	1	1	1	1	1	5	5	5
		2	2	2	2	2	2	0	0
					3	3	3	3	6

7	8	8	7	1	2	7	3	4	3
7	7	7	7	7	7	7	3	3	3
5	8	8	8	8	8	8	8	8	8
0	0	0	0	1	1	1	1	4	4
6	6	6	6	6	2	2	2	2	2

(b) Suppose you have **256MB** of virtual memory, **32MB** of physical memory, and the page size of **64KB** (2^{16}) bytes.

(i). How many pages are there in virtual memory and in physical memory?

[5 Marks]

No. of Pages in VM = 256 MB / 64KB

= 2^{12} pages

No. of frams in PM = 32MB / 64KB

= 2^{9} page frames.

(ii). Assume that the relevant portion of your Page Table is as follows:

Page	Frame #	Valid Bit
0	2	1
1	3	0
4	8	0
8	0	1
11	6	1
....	....	....
15	7	1
....	....	....
17	4	1
....	....	....
21	13	1
....	....	....

Calculate the referenced address in main memory for the following virtual addresses: **0x0011D910** and **0x00157808**.

[10 Marks]

$0x\boxed{0011}\boxed{D910}$

page No offset

Page No 17 is in frame No 4.

∴ Physical Address

0x 0004D910

$0x\boxed{0015}\boxed{7808}$

Page No. Offset

Page No 21 is in frame No 13.

∴ Physical Address -

0x 000D7808

****

(b) Suppose you have **4GB** of virtual memory, **32MB** of physical memory, and the page size of **16KB** (2^{14}) bytes.

(i). How many pages are there in virtual memory and in physical memory?

[5 Marks]

No. of Pages in Virtual Memory = 4GB / 16KB

= 2^{18} pages

No. of Page frames in Physical Memory = 30 MB / 16KB

= 2^{11} page frames.

(ii). Assume that the relevant portion of your Page Table is as follows:

Page	Frame #	Valid Bit
0	2	1
1	3	0
4	8	0
8	0	1
11	5	1
....	....	....
15	7	1
....	....	....
17	4	1
....	....	....
20	10	1
....	....	....

Calculate the referenced address in main memory for the following virtual addresses: **0x0002D910** and **0x00053808**.

[10 Marks]

0x 0002D910

$\underbrace{0000\ 0000\ 0000\ 0010\ 11}_{\text{page NO.}}\underbrace{01\ 1001\ 0001\ 0000}_{\text{offset}}$

$\therefore$ page NO = 11 $\Rightarrow$ page frame = 5

$\therefore$ Referred Address = 5 × 16KB + 6,416
= 5 × 16,384 + 6,416

= 88,336

0x 00053808

$\underbrace{0000\ 0000\ 0000\ 0101\ 00}_{\text{page NO.}}\underbrace{11\ 1000\ 0000\ 1000}_{\text{offset}}$

$\therefore$ page NO = 20 $\Rightarrow$ page frame = 10

$\therefore$ Referred Address = 10 × 16,384 + 14,344

= 178,184

****