1. A system with paging memory management, has 8 GB of logical addressing space and 1 GB of physical memory. At any given time, the system's page table presents the following content:

Página	Nº de marco	Bit de validez
0	67	valido
3	10	valido
…	…	…
23	4	valido
…	…	…
42	22	valido
…	…	…
600	1	valido
…	…	…

Fill in the blank cells in the table below. Each row represents a different assumption of field distribution of the physical and logical addresses. Ignore the gray cells. Justify your result for each case by indicating the appropriate operations you have performed.

1 Fill in the blank cells with the corresponding values

	Direc. física	Direc. lógica	Tamaño de página	Nº de marcos	Nº de páginas
Caso 1			256 KB = 2^{18}	4096=2^{12}	
Caso 2				128 * 1024 = 2^{17}	1048576=2^{20}
Caso 3	20830	6494	2048=2^{11}		
Caso 4	314816	1560000		16384 =2^{14}	

Justifique para cada caso los valores de la tabla anterior

Caso 1	Physical address = 1GBytes = 2^{30}=> 30 bits N. frames = 4096 = 2^{12} Frame size = Page size = 2^{30} / 2^{12} = 2^{18} => 256 KB
Caso 2	Logical address = 8 GB = 2^{33} => 33 bits Number of pages = 1048576 = 2^{20} Number of bits for page offset = 33 bits – 20 bits = 13 bits Page size = frame size = 2^{13} = 8KB Number of frames = 2^{30} / 2^{13} = 2^{17} => 128 * 1024 frames
Caso 3	Bits for page offset = 2^{11} = 2KB => 11bits Number of frame: floor(20830 / 2048) = 10; offset = 20830 mod 2048 = 350 In frame 10 there is allocated page 3 (see table) Logical address = 3 * 2048 + 350 = 6494
Caso 4	The frame size is 2^{30} / 2^{14} = 2^{16} = 64KB = page size Number of page = floor(1560000 / (64 * 1024)) = 23 ; Page 23 => frame 4 Offset = 1560000 mod (64 * 1024) = 52672 Physical address = 4 *(64*1024) + 52672 = 314816

1. One system uses 4Kbyte pages, pagination on demand, with the maximum logical size per process being 256 pages. At a given time this system has only 6 frames (0x12, 0x13, 0x14, 0x2A, 0x2B and 0x2C) to run user processes Y and Z. Suppose that the information related to the processes Y and Z at the instant t=50 is the one shown in the following table.

PROCESO:PÁGINA	Marco (hexadecimal)	Instante de carga	Instante de última referencia	Bit Validez
Y:0x0	0x12	10	50	1
Y:0x1	0x13	15	35	1
Y:0x2				0
Y:0x3	0x14	20	20	1
Z:0x40	0x2A	12	39	1
Z:0x41	0x2B	17	17	1
Z:0x42				0
Z:0x43				0

From the instant t=50, the following sequence of pages Y:0x2, Y:0x1, Z:0x42, Z:0x43, Y:0x3, Z:0x40, Y:0x0, Y:0x3 is referenced as indicated in the tables to be completed.

a) From the instant t=50, specify for each instant t, the evolution of the content of the frames assigned to Y and Z if a FIFO replacement algorithm of GLOBAL scope is applied.

b) From the instant t=50, suppose that the system applies an fair frame distribution policy for the processes Y and Z and indicate for each instant t, the evolution of the content of the frames if a LRU replacement algorithm of LOCAL scope is applied.

c) Indicate in a justified way if in a system with paging on demand external or internal fragmentation can appear and how much memory could be unused for that reason.

1,25 Puntos (0,5+ 0,5+0,25)

1

a) FIFO replacement algorithm GLOBAL scope.

	t=50	t=51	t=52	t=53	t=54	t=55	t=57	t=58	t=59
Marco	----	**Y:0x02**	**Y:0x01**	**Z:0x42**	**Z:0x43**	**Y:0x03**	**Z:0x40**	**Y:0x00**	**Y:0x03**
0x12	Y:0x00	Y:0x00	Y:0x00	**Z:0x42**	Z:0x42	Z:0x42	Z:0x42	Z:0x42	Z:0x42
0x13	Y:0x01	Y:0x01	**Y:0x01**	Y:0x01	Y:0x01	Y:0x01	**Z:0x40**	Z:0x40	Z:0x40
0x14	Y:0x03	Y:0x03	Y:0x03	Y:0x03	Y:0x03	**Y:0x03**	Y:0x03	Y:0x03	**Y:0x03**
0x2A	Z:0x40	Z:0x40	Z:0x40	Z:0x40	**Z:0x43**	Z:0x43	Z:0x43	Z:0x43	Z:0x43
0x2B	Z:0x41	Z:0x41	Z:0x41	Z:0x41	Z:0x41	Z:0x41	Z:0x41	**Y:0x00**	Y:0x00
0x2C	free	**Y:0x02**	Y:0x02	Y:0x02	Y:0x02	Y:0x02	Y:0x02	Y:0x02	Y:0x02

Page fault number = 5

b) LRU replacement algorithm LOCAL scope.

	t=50	t=51	t=52	t=53	t=54	t=55	t=57	t=58	t=59
Marco	------	**Y:0x02**	**Y:0x01**	**Z:0x42**	**Z:0x43**	**Y:0x03**	**Z:0x40**	**Y:0x00**	**Y:0x03**
0x12	Y:0x00	Y:0x00	Y:0x00	Y:0x00	Y:0x00	**Y:0x03**	Y:0x03	Y:0x03	**Y:0x03**
0x13	Y:0x01	Y:0x01	**Y:0x01**	Y:0x01	Y:0x01	Y:0x01	Y:0x01	Y:0x01	Y:0x01
0x14	Y:0x03	**Y:0x02**	Y:0x02	Y:0x02	Y:0x02	Y:0x02	Y:0x02	**Y:0x00**	Y:0x02
0x2A	Z:0x40	Z:0x40	Z:0x40	Z:0x40	Z:0x40	Z:0x40	**Z:0x40**	Z:0x40	Z:0x40
0x2B	Z:0x41	Z:0x41	Z:0x41	Z:0x41	**Z:0x43**	Z:0x43	Z:0x43	Z:0x43	Z:0x43
0x2C	free	free	free	**Z:0x42**	Z:0x42	Z:0x42	Z:0x42	Z:0x42	Z:0x42

Page fault number = 5

c) External or internal fragmentation and amount of memory may be unusable

When the partitions are fixed as occurs in paging, the fragmentation generated is internal.

There is not external fragmentation.

In general, the average internal fragmentation is the average case that corresponds to the page size divided by 2. In this case, the internal fragmentation in averages case is 2KB or 50% of the page size.

3. A computer system has a physical memory of 16MBytes and implements on-demand paging with 16-bit logical addresses and 256Byte page size. Logical memory management is based on multi-level paging with two levels and a first-level page table of 16 entries.

a) Determine the format of the logical and physical addresses.

b) Given the following set of references made by processes A and B:
(A,0x01EF),(A,0x01DF), (B,0x0213),(B,0x0302), (B,0x0489), (A,0x01FF), (B,0x0500), (A,0x03AB), (B,0x0304), (A,0x0207),(B,0x01AA)
Represent the evolution of memory assuming that processes A and B can only use frames 0, 1, 2, 3 and 4, which are initially free. The replacement algorithm is global LRU.

c) Determine the physical address corresponding to the logical address (A,0x0145)

d) Assuming that 8MBytes is reserved in main memory for the operating system and a minimum of 128 frames per process, indicate the maximum degree of multiprogramming of this system

a) Logical address: 16 bits ; offset: 256 B => 2^8 => 8 bits ; 8 bits for offset and 8 bits for paging
first level page has 16 entries => 2^4 => 4 bits
second level page : 16 – 4 (1st level) – 8 (offset) = 4 bits
format: **4 (1st) + 4 (2nd) + 8 (offset) bits**

Physical address: 16MB = 2^{24} => 24 bits ; frame size (offset) = 256B = 2^8 => 8 bits
Number of frames: 2^{24} / 2^8 = 2^{16} frames => 16 bits
Format: 16 (frame number) + 8 (offset) bits

b) The list of references can be split into page, offset: **(A,0x01, 0xEF),(A,0x01, 0xDF), (B,0x02, 0x13),(B,0x03, 0x02), (B,0x04, 0x89), (A,0x01, 0xFF), (B,0x05, 0x00), (A,0x03, 0xAB), (B,0x03, 0x04), (A,0x02, 0x07), (B,0x01, 0xAA)**

Frame	(A,1)	(B,2)	(B,3)	(B,4)	(A,1)	(B,5)	(A,3)	(B,3)	(A,2)	(B,1)
0	**(A,1)**	(A,1)	(A,1)	(A,1)	(A,1)	(A,1)	(A,1)	(A,1)	(A,1)	**(B,1)**
1		**(B,2)**	(B,2)	(B,2)	(B,2)	(B,2)	**(A,3)**	(A,3)	(A,3)	(A,3)
2			**(B,3)**	(B,3)	(B,3)	(B,3)	(B,3)	(B,3)	(B,3)	(B,3)
3				**(B,4)**	(B,4)	(B 4)	(B,4)	(B,4)	**(A,2)**	(A,2)
4						**(B,5)**	(B,5)	(B,5)	(B,5)	(B,5)

Number of faults: 8. **3 faults have required page replacement**, **5 faults have been solved without page replacement**

c) Logical address: (A, 0x0145) => page 0x01 offset 0x45
Page 0x01 of process A is not in memory when the sequence has been finished. A page fault will occur and the page should be allocated in frame: 4. So, the physical address will be: 4 * 256 + decimal(0x45) = 1024 + 69 = 1093 = 0x000445

d) The maximum multiprogramming degree can be obtained by allocating the maximum number of processes according to the constraints.
Memory for processes: 8 MB = 2^{23}
Number of frames: 2^{23} / 2^8 = 2^{15} frames
Minimum number of frames per process: 128 = 2^7
Maximum number of processes: 2^{15} / 2^7 = 2^8 => 256 processes
