1. Definitions:
    a. **Paging** is a memory-management scheme that splits the address space into equal sized units called pages.
    b. **Compaction** is one solution to the external fragmentation problem, which shuffles the memory contents so as to place all free memory together in one large block.
    c. **Segmentation** is a memory-management scheme that splits the memory into unequal units that may have sizes more meaningful or appropriate to the program. It actually maps the programmer's view to the actual physical memory. Consequently, it provides more freedom for the system to manage memory while the programmer would have a more natural programming environment.
    d. **Swapping** makes a process to be moved temporarily out of memory to a backing store and then brought back into memory for continued execution.
    e. **Thrashing** is one of severe performance problems that happens for a process if it is spending more time paging than executing.

2. Suppose two processes need to be mapped into main memory using pages. Process P1 consists of 7 pages, and process P2 consists of 4 pages. Assume main memory consists of 16 frames, a logical page is the same size as a physical frame, and that 4 entries in a page table fills up a frame of memory. Assume also that within the process' allocated address spaces, there are two pages of shared code 'X' and 'Y' common to both address spaces of Frame #10 and #12, respectively.
   Complete the following design for a memory management system that can store these two processes and their page tables in RAM by dragging the answers to their corresponding position in the following tables.

P1's Page Table			P2's Page Table		
Logical Page	Physical Frame	Shared Code	Logical Page	Physical Frame	Shared Code
0	1		0	0	
1	2		1	11	
2	3		2	10	**X**
3	8		3	12	**Y**
4	9				
5	**12**	Y			
6	**10**	X			

Frame #	RAM
0	**P2 - Page 0**
1	**P1 - Page 0**
2	**P1 - Page 1**
3	**P1 - Page 2**
4	P1 Page Table entries 0-3
5	**Unallocated**
6	P2 Page Table entries 0-3
7	**Unallocated**
8	**P1 - Page 3**
9	**P1 - Page 4**
10	P1 - **Page 6** and P2 - **Page 2**
11	**P2 - Page 1**
12	P1 - **Page 5** and P2 - **Page 3**
13	**Unallocated**
14	**Unallocated**
15	P1 Page Table entries **4 - 6**

3. Suppose on-demand paging is employed in addition to TLB caching. The time for a TLB hit is T = 1 ns, a memory read M = 10 ns, and a disk read D = 10 ms. Let P_TLB = 90% the probability of a TLB hit, and P = 0.001 the probability of a page fault given a TLB miss.
   What is the probability of a TLB miss?
   0.1
   What is the probability of a NO page fault?
   0.9999
   What is the calculated average memory access time in Nanoseconds (up to 3 decimal places)?
   1011.9 ns

4. The Least Recently Used (LRU) page replacement policy does not suffer from Belady's Anomaly. Intuitively, given the following sequence of page references:

   1 2 3 4 2 1 5 6 7 1

   And assuming main memory is initially unloaded. Fill in the following table to show the page faulting behavior using the LRU page replacement policy.

   Case 1: Given a frame allocation of 2

1	2	3	4	2	1	5	6	7	1
1	**1**	**3**	**3**	**2**	**2**	**5**	**5**	**7**	**7**
	2	**2**	**4**	**4**	**1**	**1**	**6**	**6**	**1**

   Case 2: Given a frame allocation of 4

1	2	3	4	2	1	5	6	7	1
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	2	**2**	**2**	**2**	**2**	**2**	**2**	**7**	**7**
		3	**3**	**3**	**3**	**5**	**5**	**5**	**5**
			4	**4**	**4**	**4**	**6**	**6**	**6**

   Call

   - A = Number of page faults in Case 1
   - B = Number of page faults in Case 2

   A is **greater than** B

5. Given a frame allocation of 3, and the following sequence of page references

   3 2 4 3 4 2 2 3 4 5 6 7 7 6 5 4 5 6 7 2 1

   and assuming main memory is initially unloaded, show the page faulting behavior using the following page replacement policies. How many page faults are generated by each page replacement algorithm?

   a. FIFO

      12

   b. OPT

      10

   c. LRU

      10

   d. Which generates the fewest page faults?

      OPT

6. A memory manager for a variable-sized region strategy has a free list of blocks of size 600, 400, 1000, 2200, 1600, and 1050 bytes. What block will be selected to honor a request for:
    a. 1603 bytes using a best-fit policy?
    2200
    b. 949 bytes using a best-fit policy?
    1000
    c. 1603 bytes using a worst-fit policy?
    2200
    d. 349 bytes using a worst-fit policy?
    2200
    e. 1603 bytes using a first-fit policy? (assume the free list is ordered as listed above)
    2200
    f. 1049 bytes using a first-fit policy?
    2200