Question 1: Which of these registers are used in paging? **(1 marks)**
PTBR, **CR3**, CS, SS

Question 2: Arrange these memories in increasing order of their access latency: a) L3, b) register, c) DRAM, d) L1
Answer: b-d-a-c **(1 marks)**

Registers->L1 cache->L2 cache->L3 cache->DRAM

Question 3: assuming page size of 4KB, what should be the optimal TLB size for a program doing a malloc of 16KB? a) 10, b) 2, c) 6, d) **4** (1 marks)
Maximum number of physical pages required = 16/4 = 4 = maximum number of valid entries in page table. Hence, maximum VPN->PPN translations = 4 = optimal number of TLB size.

Question 4: What kind of locality is exploited in a vector sum program (A+B=C) during the array accesses:
a) Temporal, b) **spatial**, c) none of the above, d) both. **(1 marks)**

for(int i=0; i<N; i++) A[i]+B[i]=C[i]; The array accesses are happening in succession, hence spatial locality. Same index is not being used again, hence no temporal locality.

Question 5: In 8-bit addressing mode, what will be the maximum size of page table? Assume page size and pointer size is 4 bytes. **(2 marks)**
a) 32 bytes, b) **256 bytes**, c) 64 bytes, d) 128 bytes
Total physical pages = 2^8 / 4 = 2^6. Maximum number of entries in page table = maximum number of pages. Hence, max size of page table = 2^6 x pointer size = 2^8 = 256 bytes.
See lecture slides for more examples.

Question 6: By a factor of what we can reduce the maximum size of a page table if we increase the page size by a factor of N? (1 marks)
a) N/4, b) N/2, c) **N**, d) none of the above
See lecture slides for examples.

Question 7: What kind of data structure is used for page table implementation in inverted page table? (1 marks)
a) Array, b) linked list, c) **hash map**, d) none of the above
See lecture slides.

Question 8: 90-10 rule says programs spend 10% of their time in 90% of their code. Is it correct: (1 marks)
a) True, b) False, c) none of the above.
Program spends 90% of their time in 10% of their code == Program spends 10% of their time in 90% of their code. Hence either of the options "**a**" or "**b**" are correct (There should have been another option for both correct, but we missed it)

Question 9: Page table entry make demand paging implementable using which bit inside the page table entry? (1 marks)
a) **"P"**, b) "PCD", c) "PWT", d) none of the above

Present bit (P) is used. See lecture 20.

Question 10: How many page faults will be there in FIFO page replacement policy for the following stream of page access for a page table of size 2: Page-A, Page-B, Page-C, Page-D, Page-D, Page-B, Page-A, Page-B? (2 marks)
a) 4, b) 3, c) 5, d) **6**, e) none of the above
A (+1) – B (+1) – C (+1) – D (+1) – D – B (+1) – A (+1) – B
Total page faults = 6

Question 11: Clock algorithm uses the following data structure for iterating across the active pages currently being used by the processes: a) array, b) hash map, c) linked list, d) **circular linked list** (1 marks)
See lecture slides.

Question 12: Clock algorithm is used for approximating which of the following page replacement policies? A) MIN, b) FIFO, c) RANDOM, d) **LRU** (1 marks)
See lecture slides.

Question 13: Which of followings are shared by the threads of a single process: a) stack, b) registers, c) **heap**, d) program counter, e) **global variables in program**? (1 marks)

See lecture slides. Threads cannot share stack, registers and program counter as each of the threads will have independent execution path.

Question 14: Arrange the followings in the order of their **creation** cost: a) 100 kernel threads, b) 100 user level threads, and c) 100 processes? (1 marks)
Answer: b-a-c
See lecture slides.

Question 15: Arrange the followings in the order of their **context switch** cost: a) processes, b) user level threads, and c) kernel threads? (1 marks)
Answer: b-c-a
See lecture slides.

Question 16: Which of the followings is/are correct: (1 marks)

a) Multithreaded array sum uses more OS resources as compared to its multi-process version.
b) Deadlocks can happen when multiple threads attempt to lock one mutex variable simultaneously.
c) Starvation is guaranteed by using pthread mutex lock v/s using semaphores.

Answer: None of the above

Question 17: The OS kernel is unaware of the context switch of: a) processes, b) pthreads, c) **user level threads**, d) none of the above. (1 marks)
See lecture slides.

Question 18: Which of the followings are correct? (1 marks)

a) Process can create several thousands of threads.
b) Process can create limited number of threads.
c) After creating threads, a process can call **wait** system call to wait for the termination of those threads.

Answer: **b**
Threads inside the process share the address space of the parent process for creating their own stack. Creating unlimited number of threads (thousands or infinite) will lead to out of memory error.

Question 19: Which of the following APIs can be used to avoid deadlocks? (1 marks)

a) pthread_mutex_lock
b) pthread_mutex_trylock
c) pthread_mutex_timedlock

Answer: c
See lecture slides. trylock and lock will behave the same way in the deadlock example explained in lecture slides.

Question 20: Which one of these statements are correct about x86 processors?

a) Paging can be used alone.
b) Segmentation can be used alone.
c) Paging and segmentation can be used together.

Answer: b and c.
See lecture slides. Segmentation is mandatory but paging is optional (it may/may_not be used along with segmentation).