Placement Examination – I

*This question paper consists of two sections A and B. You should answer **all** questions from **Section A** and any **five** questions from **Section B***

SECTION – A (25 x 2 = 50)

1. What is a *Thread*? What are the differences between a *Thread* and a *Process*?
2. State the differences between a *counting semaphore* and a *mutex.*
3. What is a *Scheduler?* State and define the different types of Schedulers. How it is different from a *Dispatcher*?

4. Let the initial value of a Counting Semaphore be 20. What will be its value after 9P and 6V operations? Here, P means *wait* and V means *signal*.

5. Differentiate between *Multilevel Queue Scheduling* and *Multilevel Feedback Queue Scheduling*.

6. Differentiate between *External Fragmentation* and *Internal Fragmentation* of memory.

7. Assume that out of 10 pages, 4 pages can be found at a time in the main memory. The average page-fault service time is 25 milliseconds and the memory access time is 100 nanoseconds. Find out the *effective access time*.

8. Consider the following page references using *three* frames that are initially empty. Find the number of *page faults* for the following page references : 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 when *FIFO* and *LRU* page-replacement algorithms are used.

9. State and explain the requirements of the solution of the *critical section problem*.

10. Consider 3 processes P1, P2 and P3 as shown in the below table:

Process	Arrival Time	Burst Time
P1	0	5
P2	1	7
P3	3	4

Calculate the *Average Waiting Time* if *FCFS* and *RR* (with time quantum of 2 units) is used.

11. A system uses FIFO page replacement policy. It has 4 page frames with no pages loaded to begin with. The system first accesses 100 distinct pages in some order and then accesses the same 100 pages but now in the reverse order. How many page faults will occur?

a. 196
b. 192
c. 197
d. 195

N.B. Please explain your answer.

12. Consider a disk system with 100 cylinders. The requests to access the cylinders occur in the following sequence: 4, 34, 10, 7, 19, 73, 2, 15, 6, 20. Assuming that the head is currently at cylinder 50, what is the time taken to satisfy all requests if it takes 1 millisecond to move from one cylinder to the adjacent one and *shortest seek time first* policy is used ?

13. What are the different *states* of a process? Explain them with the help of a block diagram.

14. State the differences between *Physical Memory* and *Virtual Memory*. How we can map between them ?

15. State the 4 neccessary and sufficient conditions for Deadlock to happen.
16. What is *Starvation* in Operating System? How it can be solved?
17. What are the various *Inter Process Communication (IPC)* mechanisms?
18. What is *Context Switching*? When does it happen?
19. What is the basic difference between *Preemptive* and *Non-preemptive* scheduling?
20. Throw some light on *Race Condition* of processes. How it can be solved?
21. Why is *Round-Robin (RR)* algorithm considered better than *FIFO* algorithm?
22. What is *Belady's* Anomaly? When does it occur?
23. What is *Translation Look-Aside Buffer (TLB)*?
24. Differentiate between *Demand Paging* and *Pre-Paging*.
25. Explain the *Working Set* mechanism in Operating Systems.

SECTION – B (5 x 10 = 50)

1. Write a JAVA program to input a string from the user and form another string consisting of middle elements of each word of the input string. If the word length is odd then take the middle character and if it is even then take the 2 middle characters. For e.g. : Input String = "My name is Dipanjan Saha", Output String = "My am is an ah".

2. Write a JAVA program to find out the unique characters and their frequency of occurrence in a string given by the user.

3. Write a JAVA program to remove all the occurrences of a given character in a given string and hence return the modified string.

4. Write a JAVA program to find out the first non-repeated character of a given string.

5. Write a JAVA program to find out all the unique triplets in a given range which sums up to a given value by the user.

6. Write a JAVA program to display all the *Leaders* in the array. An element is *leader* if it is greater than all the elements present in its right-hand side in the array. *Note:* The last element of the array is always a Leader.

7. Write a JAVA program to find out the difference between the smallest and the largest number which is embedded in a given string. For e.g. Input String: "*My2020 na2049me is Dip2018an5463ja98*". Here, the largest number is 5463 and the smallest number is 98. So output will be their difference.

----------XXXXXX----------