Q1 .WAP to implement First fit allocation strategy. Take size of 5 memory holes initially. Then take size of 5 processes. As output specify which process fits in while memory hole.

Q2. WAP to implement Best fit allocation strategy. Take size of 5 memory holes initially. Then take size of 5 processes. As output specify which process fits in while memory hole.

Q3. WAP to implement Worst fit allocation strategy. Take size of 5 memory holes initially. Then take size of 5 processes. As output specify which process fits in while memory hole.

Q4. WAP to implement FCFS scheduling algo. Implement for 5 processes. Print the turn around time and waiting time for each process. Print average turn around time and average waiting time too.

Q5. WAP to implement round robin scheduling algo. Implement for 5 processes, input execution time of each process. Take time quantum for round robin as 2 sec. Print the turn around time and waiting time for each process. Print average turn around time and average waiting time too.

Q6. WAP to implement Shortest Job First(non-preemptive) scheduling algo. Implement for 5 processes, input execution time of each process. Print the turn around time and waiting time for each process. Print average turn around time and average waiting time too.

Q7. WAP to implement Shortest Remaining Time First scheduling algo.
Implement for 5 processes, input execution time and arrival time of each process.
Print the turn around time and waiting time for each process. Print average turn around time and average waiting time too.

Q8. WAP to implement Non-Preemptive Priority based scheduling algo.
Implement for 5 processes, input execution time and priority of each process.
Print the turn around time and waiting time for each process. Print average turn around time and average waiting time too.

Q9. WAP where parent forks the child and both execute the same program & code. Print the process ID of parent n child process.

Q10. WAP where parent forks the child & both execute same program but different code section. Print the process ID of parent process and child process. Print the statement “Hello, I m child process” in child process.

Q11. WAP where parent forks the child & both execute different programs, child executes the code for the command “date”

Q12. WAP where parent forks the child & both execute same program but different code section and parent waits for the child to execute first and then execute itself. Child executes code for “ls” command

Q13. Using n fork calls, create 2^n processes and print the process id of each of them. Take n as 3.

Q14. WAP to create a thread and assign a function for calculating sum of first n natural numbers to the thread. N to be given as input on command line.

Q15. WAP to create 2 threads and assign a function for calculating sum of first n natural numbers to one and for calculating cube of the number to another. User should be asked to enter two numbers.

Q16. WAP to create a thread and assign a function for calculating sum of squares of first n natural numbers to the thread. Both numbers to be given on command line.

Q17. WAP to create 2 threads and assign a function for printing the multiplication table of a number and for calculating sum of cubes of first n natural numbers. Both numbers to be given on command line.

Q18. WAP to create a thread and assign a function for printing the multiplication table of a number. Number to be entered by user.