

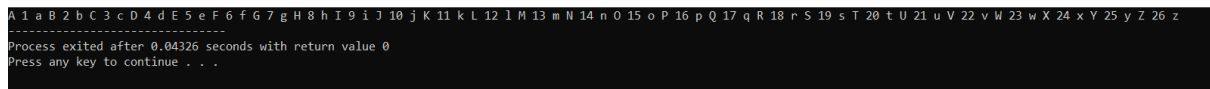
Jash Shah- AU1940286

Q-1.b

Description

First, we define a semaphore, which is a data type that is shared by multiple threads. Then, for writing different characters in the pattern, we write three thread functions: A 1 a B 2 b..Z 26 z . If you query some synchronised states, you can find out whether nums or chars are permitted. The first thread will print A and then you have to wait, followed by the second thread print 1 and then you have to wait, followed by the third thread printing a and then you have to wait, and so on until all of the alphabets have been printed.

Screenshot of Output:



```
A 1 a B 2 b C 3 c D 4 d E 5 e F 6 f G 7 g H 8 h I 9 i J 10 j K 11 k L 12 l M 13 m N 14 n O 15 o P 16 p Q 17 q R 18 r S 19 s T 20 t U 21 u V 22 v W 23 w X 24 x Y 25 y Z 26 z
-----
Process exited after 0.04326 seconds with return value 0
Press any key to continue . . .
```

Q-2.

Description of Round Robin.

Round robin is pre-emptive scheduling algorithm. Pre-emptive processes means that process can be interrupted and moved to ready state by OS. Preemption only occurs when new process arrives. The simplest method to reduce the penalty that FCFS applies on short processes is to use clock-based preemption. Because each process is given a slice of time before being preempted, it is also known as time slicing. Each request is linked with a fixed time slice known as the quantum. At regular intervals, a clock interrupt is generated. The presently operating process is placed in the ready queue when an interrupt occurs. Next job is chosen on basis of FIFO (First in First out). In the code first we have to take some input like quantum number, no of processes, arrival time, burst time etc. By adding the entire time taken and subtracting the arrival time, the turn around time is computed. By subtracting the arrival and burst times from the total and adding it to the waiting time, the waiting time is computed.

Screenshot of Output:

```

Enter the number of process in the system: 4

Enter the Arrival and Burst time of the Process 1
Arrival time is: 0
Burst time is: 1

Enter the Arrival and Burst time of the Process 2
Arrival time is: 1
Burst time is: 7

Enter the Arrival and Burst time of the Process 3
Arrival time is: 2
Burst time is: 5

Enter the Arrival and Burst time of the Process 4
Arrival time is: 3
Burst time is: 6

Enter the Time Quantum for the process: 3

Process No      Burst Time      TAT      Waiting Time
Process No 1    1      1      0
Process No 3    5      13     8
Process No 4    6      15     9
Process No 2    7      18    11
Average Turn Around Time: 7.000000
Average Waiting Time: 11.750000

```

Description of Modified Round Robin

First of all we have to ask user to enter number of process. For each process, the user enters burst time, arrival time etc. we have to set the initial time to 0. Priorities are based on arrival time. This algorithm chooses processes with a shorter arrival time than the current time. Each selected process is given a point by the algorithm. This point is used to sort the processes that have arrived. The programme calculates Time Quantum based on the processes selected. To calculate Time Quantum, the burst time's mean and median are calculated. Points and time quantum are recalculated when a new process is introduced.

Screenshot of Output:

```

Enter Number of processes in the system: 4

Enter the Arrival and Burst time of the Process 1
Arrival time is: 0
Burst time is: 4

Enter the Arrival and Burst time of the Process 2
Arrival time is: 1
Burst time is: 7

Enter the Arrival and Burst time of the Process 3
Arrival time is: 2
Burst time is: 5

Enter the Arrival and Burst time of the Process 4
Arrival time is: 3
Burst time is: 6

Quantum time calculated is : 6
Processes      Waitingtime      TurnAroundTime
process no 1    0      4
process no 2    16     22
process no 3    4      11
process no 4    11     16

```