

Question #1

Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

Question #2

Show that the following algorithm is a correct solution to the critical section problem for two processes (satisfies the 3 conditions). The solution for process P_i ($i = 0$ or 1) with P_j ($j = 1$ or 0) is shown below:

```

flag[2];
int turn = 0;

flag[i] = 1;
while (flag[j])
{
    if (turn == j)
    {
        flag[i] = 0;
        while(turn == j)
            ;
        flag[i] = 1;
    }
}

/* enter C.S. */
/* exit C.S. */
turn = j;
flag[i] = 0;

```

Question #3

What is the meaning of the term busy waiting? What other kinds of waiting are there in an operating system? Can busy waiting be avoided altogether? Explain your answer.

Question #4

Explain why implementing synchronization primitives by disabling interrupts is not appropriate in a single-processor system if the synchronization primitives are to be used in user-level programs.

Question #5

Explain why interrupts are not appropriate for implementing synchronization primitives in multiprocessor systems.