Think-Pair-Share
What is the state of Memory and CPU registers after executing two instructions?
Q. 01

What is the value in DR after executing two instructions?
Is this value also in the Memory?
Q. 01

### CPU Registers

<table>
<thead>
<tr>
<th>Memory</th>
<th>CPU Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0531 0x0860</td>
<td>PC 0x0533</td>
</tr>
<tr>
<td>0x0532 0x3862</td>
<td>DR 0x0008</td>
</tr>
<tr>
<td>0x0533 0x1861</td>
<td>IR 0x3862</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>0x0860 0x0008</td>
<td></td>
</tr>
<tr>
<td>0x0861 0x0002</td>
<td></td>
</tr>
<tr>
<td>0x0862 0x0005</td>
<td></td>
</tr>
</tbody>
</table>

### Opcode

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Load from memory to DR</td>
</tr>
<tr>
<td>0001</td>
<td>Store DR to memory</td>
</tr>
<tr>
<td>0010</td>
<td>Divide DR by memory</td>
</tr>
<tr>
<td>0011</td>
<td>Subtract memory from DR</td>
</tr>
<tr>
<td>0100</td>
<td>Multiply memory with DR</td>
</tr>
<tr>
<td>0101</td>
<td>Add to DR from memory</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

### Registers

<table>
<thead>
<tr>
<th>PC</th>
<th>Program Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR</td>
<td>Data Register</td>
</tr>
<tr>
<td>IR</td>
<td>Instruction Register</td>
</tr>
</tbody>
</table>
Q. 01

Memory

<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0531</td>
<td>0x0860</td>
</tr>
<tr>
<td>0x0532</td>
<td>0x3862</td>
</tr>
<tr>
<td>0x0533</td>
<td>0x1861</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0x0860</td>
<td>0x0008</td>
</tr>
<tr>
<td>0x0861</td>
<td>0x0002</td>
</tr>
<tr>
<td>0x0862</td>
<td>0x0005</td>
</tr>
</tbody>
</table>

CPU Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>0x0533</td>
</tr>
<tr>
<td>DR</td>
<td>0x0003</td>
</tr>
<tr>
<td>IR</td>
<td>0x3862</td>
</tr>
</tbody>
</table>

Opcode

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Load from memory to DR</td>
</tr>
<tr>
<td>0001</td>
<td>Store DR to memory</td>
</tr>
<tr>
<td>0010</td>
<td>Divide DR by memory</td>
</tr>
<tr>
<td>0011</td>
<td>Subtract memory from DR</td>
</tr>
<tr>
<td>0100</td>
<td>Multiply memory with DR</td>
</tr>
<tr>
<td>0101</td>
<td>Add to DR from memory</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Execution Stage

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x3862</td>
<td>0011 1000 0110 0010</td>
</tr>
</tbody>
</table>

Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>DR</td>
<td>Data Register</td>
</tr>
<tr>
<td>IR</td>
<td>Instruction Register</td>
</tr>
</tbody>
</table>
Q. 01

Memory

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0531</td>
<td>0x0860</td>
</tr>
<tr>
<td>0x0532</td>
<td>0x3862</td>
</tr>
<tr>
<td>0x0533</td>
<td>0x1861</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>0x0860</td>
<td>0x0008</td>
</tr>
<tr>
<td>0x0861</td>
<td>0x0002</td>
</tr>
<tr>
<td>0x0862</td>
<td>0x0005</td>
</tr>
</tbody>
</table>

CPU Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>0x0534</td>
</tr>
<tr>
<td>DR</td>
<td>0x0003</td>
</tr>
<tr>
<td>IR</td>
<td>0x1861</td>
</tr>
</tbody>
</table>

Opcode

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Load from memory to DR</td>
</tr>
<tr>
<td>0001</td>
<td>Store DR to memory</td>
</tr>
<tr>
<td>0010</td>
<td>Divide DR by memory</td>
</tr>
<tr>
<td>0011</td>
<td>Subtract memory from DR</td>
</tr>
<tr>
<td>0100</td>
<td>Multiply memory with DR</td>
</tr>
<tr>
<td>0101</td>
<td>Add to DR from memory</td>
</tr>
</tbody>
</table>

Fetch Stage

0x1861

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>1000</td>
</tr>
<tr>
<td>0110</td>
<td>0001</td>
</tr>
</tbody>
</table>

Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>DR</td>
<td>Data Register</td>
</tr>
<tr>
<td>IR</td>
<td>Instruction Register</td>
</tr>
</tbody>
</table>
### Q. 01

#### Execution Stage

<table>
<thead>
<tr>
<th>Memory</th>
<th>CPU Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0531</td>
<td>PC: 0x0534</td>
</tr>
<tr>
<td>0x0532</td>
<td>DR: 0x0003</td>
</tr>
<tr>
<td>0x0533</td>
<td>IR: 0x1861</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>0x0860</td>
<td></td>
</tr>
<tr>
<td>0x0861</td>
<td>0x0003</td>
</tr>
<tr>
<td>0x0862</td>
<td>0x0005</td>
</tr>
</tbody>
</table>

#### Opcode

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Load from memory to DR</td>
</tr>
<tr>
<td>0001</td>
<td>Store DR to memory</td>
</tr>
<tr>
<td>0010</td>
<td>Divide DR by memory</td>
</tr>
<tr>
<td>0011</td>
<td>Subtract memory from DR</td>
</tr>
<tr>
<td>0100</td>
<td>Multiply memory with DR</td>
</tr>
<tr>
<td>0101</td>
<td>Add to DR from memory</td>
</tr>
</tbody>
</table>

#### Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>DR</td>
<td>Data Register</td>
</tr>
<tr>
<td>IR</td>
<td>Instruction Register</td>
</tr>
</tbody>
</table>
Q. 01

What many times the Memory is accessed for each instruction?
Q. 02

If a system has two levels of caches and main memory with the following parameters, what is the EAT?

Consider the (1) data and (2) memory hierarchy perspectives.

<table>
<thead>
<tr>
<th></th>
<th>Hit ratio</th>
<th>Access time</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 caches</td>
<td>$H_1$</td>
<td>$T_1$</td>
</tr>
<tr>
<td>L2 caches</td>
<td>$H_2$</td>
<td>$T_2$</td>
</tr>
<tr>
<td>Main memory</td>
<td>$-</td>
<td>$T_3$</td>
</tr>
</tbody>
</table>
Q. 02

<table>
<thead>
<tr>
<th></th>
<th>Hit ratio</th>
<th>Access time</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 caches</td>
<td>$H_1$</td>
<td>$T_1$</td>
</tr>
<tr>
<td>L2 caches</td>
<td>$H_2$</td>
<td>$T_2$</td>
</tr>
<tr>
<td>Main memory</td>
<td>$-$</td>
<td>$T_3$</td>
</tr>
</tbody>
</table>

From the **data** perspective:

Q.03.01: What is the distribution of data?

$$H_1 \quad (1 - H_1)H_2 \quad (1 - H_1)(1 - H_2)$$

L1 Cache        L2 Cache        Main memory

Q.03.02: What is the time to deliver each portion of data?

$$H_1 \times T_1 \quad (1 - H_1)H_2 \times (T_1 + T_2) \quad (1 - H_1)(1 - H_2) \times (T_1 + T_2 + T_3)$$

L1 Cache        L2 Cache        Main memory
<table>
<thead>
<tr>
<th></th>
<th>Hit ratio</th>
<th>Access time</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 caches</td>
<td>$H_1$</td>
<td>$T_1$</td>
</tr>
<tr>
<td>L2 caches</td>
<td>$H_2$</td>
<td>$T_2$</td>
</tr>
<tr>
<td>Main memory</td>
<td>—</td>
<td>$T_3$</td>
</tr>
</tbody>
</table>

From the **memory hierarchy** perspective:

Q.03.01: What is average time to deliver all data from main memory to L2 caches?

$$ (1 - H_1)(1 - H_2) * T_3 $$

Main to L2 caches

Now, no data is in the main memory
Q. 02

<table>
<thead>
<tr>
<th></th>
<th>Hit ratio</th>
<th>Access time</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 caches</td>
<td>$H_1$</td>
<td>$T_1$</td>
</tr>
<tr>
<td>L2 caches</td>
<td>$H_2$</td>
<td>$T_2$</td>
</tr>
<tr>
<td>Main memory</td>
<td>—</td>
<td>$T_3$</td>
</tr>
</tbody>
</table>

From the **memory hierarchy** perspective:

Q.03.02: What is time to deliver all data from L2 caches to L1 caches?

\[
(1 - H_1) \cdot T_2 \quad (1 - H_1)(1 - H_2) \cdot T_3
\]

- L2 caches to L1 caches
- Main to L2 caches

Now, no data is in the L2 caches or main memory
<table>
<thead>
<tr>
<th></th>
<th>Hit ratio</th>
<th>Access time</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 caches</td>
<td>$H_1$</td>
<td>$T_1$</td>
</tr>
<tr>
<td>L2 caches</td>
<td>$H_2$</td>
<td>$T_2$</td>
</tr>
<tr>
<td>Main memory</td>
<td>$-$</td>
<td>$T_3$</td>
</tr>
</tbody>
</table>

From the **memory hierarchy** perspective:

Q.03.03: What is time to deliver all data from L1 caches to CPU?

\[
T_1 \quad (1 - H_1) \ast T_2 \quad (1 - H_1)(1 - H_2) \ast T_3
\]

L1 cache to CPU \quad L2 caches to L1 caches \quad Main to L2 caches

Now, all data is delivered to CPU (registers)
### Q. 02

<table>
<thead>
<tr>
<th></th>
<th>Hit ratio</th>
<th>Access time</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 caches</td>
<td>$H_1 = 0.9$</td>
<td>$T_1 = 20$</td>
</tr>
<tr>
<td>L2 caches</td>
<td>$H_2 = 0.6$</td>
<td>$T_2 = 60$</td>
</tr>
<tr>
<td>Main memory</td>
<td>$-$</td>
<td>$T_3 = 12 \times 10^6$</td>
</tr>
</tbody>
</table>

Try your homework problems!
Q. 03

Suppose we have a multiprogrammed computer where each job has identical characteristics: Each job runs $N$ computational periods of length $T$, where the first half of $T$ is spent on computation and the second half on I/O. Assume we have four jobs running on the computer. Jobs are dispatched in a simple round-robin manner, and I/O activity can overlap with CPU operations.

Compute:
1. **Average turnaround time** (average total time to complete a job)
2. **Throughput** (average number of jobs completed per time period $T$)
3. **CPU utilization** (percentage of time that the processor is not idle)
Q. 03

Suppose we have a multiprogrammed computer where each job has identical characteristics: Each job runs $N$ computational periods of length $T$, where the first half of $T$ is spent on computation and the second half on I/O. Assume we have four jobs running on the computer. Jobs are dispatched in a simple round-robin manner, and I/O activity can overlap with CPU operations.

Compute:

1. **Average turnaround time** (average time to complete a job)

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>2T</th>
<th>3T</th>
<th>4T</th>
<th>5T</th>
<th>6T</th>
<th>7T</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I/O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Execution graph
Q. 03

Suppose we have a multiprogrammed computer where each job has identical characteristics: Each job runs $N$ computational periods of length $T$, where the first half of $T$ is spent on computation and the second half on I/O. Assume we have four jobs running on the computer. Jobs are dispatched in a simple round-robin manner, and I/O activity can overlap with CPU operations.

Compute:

1. **Average turnaround time** (average time to complete a job)
Q. 03

Q. 03.1.1: How much time to finish one unit (period) of a job?

For example, how much time to finish a unit of Job 1?

2T

Compute:

1. **Average turnaround time** (average time to complete a job)
Q. 03

Q. 03.1.1: How much time to finish one unit (period) of a job? For example, how much time to finish a unit of Job 1?

2T

How about the time to finish a unit of Job 2? 2T

How about the time to finish a unit of a job? 2T

Compute:

1. **Average turnaround time** (average time to complete a job)

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>2T</th>
<th>3T</th>
<th>4T</th>
<th>5T</th>
<th>6T</th>
<th>7T</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>I/O</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Round 1 | Round 2 | Round 3 | ...
Q. 03

Q. 03.1.2: How much time to finish a job with $N$ units (periods)?

$$2T \times N$$

Time to finish one unit * The number of units of a job

So this number is the **average turnaround time**

---

**Compute:**

1. **Average turnaround time** (average time to complete a job)

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>2T</th>
<th>3T</th>
<th>4T</th>
<th>5T</th>
<th>6T</th>
<th>7T</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>I/O</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

...
Suppose we have a multiprogrammed computer where each job has identical characteristics: Each job runs $N$ computational periods of length $T$, where the first half of $T$ is spent on computation and the second half on I/O. Assume we have four jobs running on the computer. Jobs are dispatched in a simple round-robin manner, and I/O activity can overlap with CPU operations.

Compute:

2. Throughput (average number of jobs completed per time period $T$)

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>2T</th>
<th>3T</th>
<th>4T</th>
<th>5T</th>
<th>6T</th>
<th>7T</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>I/O</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Round 1 | Round 2 | Round 3 | ...

Suppose we have a multiprogrammed computer where each job has identical characteristics: Each job runs $N$ computational periods of length $T$, where the first half of $T$ is spent on computation and the second half on I/O. Assume we have four jobs running on the computer. Jobs are dispatched in a simple round-robin manner, and I/O activity can overlap with CPU operations.

2. Throughput (average number of jobs completed per time period $T$)

Q. 03.2.1: How much time to finish the four jobs?

4 jobs $\rightarrow$ $2T \times N$ (Without considering the overhead)
Q. 03

Suppose we have a multiprogrammed computer where each job has identical characteristics: Each job runs $N$ computational periods of length $T$, where the first half of $T$ is spent on computation and the second half on I/O. Assume we have four jobs running on the computer. Jobs are dispatched in a simple round-robin manner, and I/O activity can overlap with CPU operations.

2. Throughput (average number of jobs completed per time period $T$)

Q. 03.2.2: How many jobs we can finish within time period $T$?

4 jobs $\rightarrow$ $2T \times N$ (Without considering the overhead)

This is the throughput!

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>2T</th>
<th>3T</th>
<th>4T</th>
<th>5T</th>
<th>6T</th>
<th>7T</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>I/O</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Round 1 | Round 2 | Round 3 | ...
Suppose we have a multiprogrammed computer where each job has identical characteristics: Each job runs $N$ computational periods of length $T$, where the first half of $T$ is spent on computation and the second half on I/O. Assume we have four jobs running on the computer. Jobs are dispatched in a simple round-robin manner, and I/O activity can overlap with CPU operations.

2. Throughput (average number of jobs completed per time period $T$)

Q. 03.2.2: How many jobs we can finish within time period $T$?

This is the throughput!

4 jobs \[ \frac{2T \times N}{T} \]

(Without considering the overhead)

? (throughput) = \[ \frac{4T}{2TN} = \frac{2}{N} \]
Q. 03

Suppose we have a multiprogrammed computer where each job has identical characteristics: Each job runs \( N \) computational periods of length \( T \), where the first half of \( T \) is spent on computation and the second half on I/O. Assume we have four jobs running on the computer. Jobs are dispatched in a simple round-robin manner, and I/O activity can overlap with CPU operations.

Compute:
1. **Average turnaround time** (average time to complete a job) \( \frac{2T N}{4 \times \frac{T}{2}} \) 
2. **Throughput** (average number of jobs completed per time period \( T \)) \( \frac{4 \times \frac{T}{2}}{2TN} \) 
3. **CPU utilization** (percentage of time that the processor is not idle) \( 100\% \)

A last question:
Does the *average turnaround time* equals to \( (1 / \text{throughput}) \)?

Takeaway: consider the problem from hardware perspective using the execution graph
Q. 03
Suppose we have a multiprogrammed computer where each job has identical characteristics: Each job runs $N$ computational periods of length $T$, where the first half of $T$ is spent on computation and the second half on I/O. Assume we have four jobs running on the computer. Jobs are dispatched in a simple round-robin manner, and I/O activity can overlap with CPU operations.

Compute:
1. Average turnaround time (average time to complete a job) $2TN$
2. Throughput (average number of jobs completed per time period $T$) $4*T/2TN$
3. CPU utilization (percentage of time that the processor is not idle) $100$

A last question:
Does the average turnaround time equals to $(1 / \text{throughput})$?

Takeaway: consider the problem from hardware perspective using the execution graph
Q. 04

Consider a **FCFS** (non-preemptive) scheduler

\[ P_1: t_1 = 60 \text{ (arrival at } 0 - \mu) \]
\[ P_2: t_2 = 10 \text{ (arrival at } 0) \]
\[ P_3: t_3 = 30 \text{ (arrival } 30) \]
\[ P_4: t_4 = 20 \text{ (arrival at } 50) \]
\[ \mu = \text{an extremely small unit of time} \]

1. Draw Gantt Chart

2. Compute the following quantities:
   - throughput
   - CPU utilization
   - average service time (time spent executing on the CPU)
   - average wait time (time spent ready waiting for CPU)
   - average turnaround time (total average time spent waiting + executing)
Consider a FCFS (non-preemptive) scheduler

- \( P_1 : t_1 = 60 \) (arrival at 0 - \( \mu \))
- \( P_2 : t_2 = 10 \) (arrival at 0)
- \( P_3 : t_3 = 30 \) (arrival 30)
- \( P_4 : t_4 = 20 \) (arrival at 50)

\( \mu \) = an extremely small unit of time

1. Draw Gantt Chart

The Gantt chart shows the arrival times and processing times for processes 1 through 4. The horizontal axis represents time in units of 10, starting from 0 and ending at 120. The vertical axis represents the processes. The chart indicates:

- \( P_1 \) and \( P_2 \) arrive at time 0.
- \( P_3 \) arrives at time 30.
- \( P_4 \) arrives at time 50.

Each process is depicted with a shaded bar indicating its processing time.
Q. 04

2. Compute the following quantities:
   – throughput  \[\frac{120}{4} = \frac{1}{?}\Rightarrow ? = \frac{1}{30}\]
   – CPU utilization 100%
   – average service time (time spent executing on the CPU)
   – average wait time (time spent ready waiting for CPU)
   – average turnaround time (total average time spent waiting + executing)
Q. 04

2. Compute the following quantities:
   – throughput \( \frac{120}{4} = 1 / \, ? \Rightarrow \, ? = \frac{1}{30} \)
   – CPU utilization \( 100\% \)
   – average service time (time spent executing on the CPU)

   \[
   P1: 60 \quad P2: 10 \quad P3: 30 \quad P4: 20
   \]

   Average: \( \frac{60 + 10 + 30 + 20}{4} = 30 \)
2. Compute the following quantities:
   – throughput \[ \frac{120}{4} = \frac{1}{?} \rightarrow ? = \frac{1}{30} \]
   – CPU utilization \[ 100\% \]
   – average service time (time spent executing on the CPU)
   – average wait time (time spent ready waiting for CPU)

   \[ \text{P1: 0} \quad \text{P2: 60} \quad \text{P3: 40} \quad \text{P4: 50} \]

   \[ \text{Average: } \frac{0 + 60 + 40 + 50}{4} = \frac{150}{4} = 37.5 \]
Q. 04

2. Compute the following quantities:
   - throughput \( \frac{120}{4} = 1 / ? \Rightarrow ? = 1 / 30 \)
   - CPU utilization \( 100\% \)
   - average service time (time spent executing on the CPU)
   - average wait time (time spent ready waiting for CPU)
   - average turnaround time (total average time spent waiting + executing)

Turnaround = 30 (service time) + 37.5 (wait time) = 67.5
Q. 05

Consider a **Round Robin** \((q = 10)\) scheduler

\[
P_1 : t_1 = 60 \text{ (arrival at } 0 - \mu) \\
P_2 : t_2 = 10 \text{ (arrival at } 0) \\
P_3 : t_3 = 30 \text{ (arrival at } 30) \\
P_4 : t_4 = 20 \text{ (arrival at } 50) \\
\mu = \text{ an extremely small unit of time}
\]

1. **Draw Gantt Chart**
2. **Compute the following quantities:**
   - throughput
   - CPU utilization
   - average service time (time spent executing on the CPU)
   - average wait time (time spent ready waiting for CPU)
   - average turnaround time (total average time spent waiting + executing)
Consider a **Round Robin** \((q = 10)\) scheduler

- \(P_1: t_1 = 60\) (arrival at \(0 - \mu\))
- \(P_2: t_2 = 10\) (arrival at \(0\))
- \(P_3: t_3 = 30\) (arrival 30)
- \(P_4: t_4 = 20\) (arrival at 50)

\(\mu\) = an extremely small unit of time

1. **Draw Gantt Chart**

   **At time 0:**

   ![Gantt Chart Diagram]

   
   - \(P_1\) and \(P_2\) arrive
   - \(P_3\) arrives
   - \(P_4\) arrives
Consider a **Round Robin** \((q = 10)\) scheduler

\[ P_1: t_1 = 60 \text{ (arrival at } 0 - \mu) \]
\[ P_2: t_2 = 10 \text{ (arrival at } 0) \]
\[ P_3: t_3 = 30 \text{ (arrival at } 30) \]
\[ P_4: t_4 = 20 \text{ (arrival at } 50) \]
\[ \mu = \text{an extremely small unit of time} \]

1. **Draw Gantt Chart**

![Gantt Chart](image_url)
Q. 05

Consider a **Round Robin** \((q = 10)\) scheduler

\(P_1: t_1 = 60\) (arrival at \(0 - \mu\))

\(P_2: t_2 = 10\) (arrival at \(0\))

\(P_3: t_3 = 30\) (arrival 30)

\(P_4: t_4 = 20\) (arrival at 50)

\(\mu = \text{an extremely small unit of time}\)

1. **Draw Gantt Chart**

![Gantt Chart]

P1 and P2 arrive  P3 arrives  P4 arrives
Consider a **Round Robin** \((q = 10)\) scheduler

\[
P_1: \ t_1 = 60 \text{ (arrival at } 0 - \mu) \\
P_2: \ t_2 = 10 \text{ (arrival at } 0) \\
P_3: \ t_3 = 30 \text{ (arrival at } 30) \\
P_4: \ t_4 = 20 \text{ (arrival at } 50) \\
\mu = \text{an extremely small unit of time}
\]

1. **Draw Gantt Chart**

   ![Gantt Chart Diagram]

   **At time 30:**

   - P1
   - P3

   **P1 and P2 arrive**  **P3 arrives**  **P4 arrives**
Consider a Round Robin \((q = 10)\) scheduler

\[
P_1: t_1 = 60 \text{ (arrival at } 0 - \mu) \\
P_2: t_2 = 10 \text{ (arrival at } 0) \\
P_3: t_3 = 30 \text{ (arrival } 30) \\
P_4: t_4 = 20 \text{ (arrival at } 50) \\
\mu = \text{an extremely small unit of time}
\]

1. **Draw Gantt Chart**

```
0 10 20 30 40 50 60 70 80 90 100 110 120
P1
P2
P3
P4
```

P1 and P2 arrive  P3 arrives  P4 arrives
Consider a **Round Robin** \((q = 10)\) scheduler

\[
P_1: t_1 = 60 \text{ (arrival at } 0 - \mu) \\
P_2: t_2 = 10 \text{ (arrival at } 0) \\
P_3: t_3 = 30 \text{ (arrival at } 30) \\
P_4: t_4 = 20 \text{ (arrival at } 50) \\
\mu = \text{an extremely small unit of time}
\]

1. **Draw Gantt Chart**

At time 50:

\[
\begin{array}{c|cc}
\hline
& P1 & P4 & P3 \\
\hline
0 & & & \\
10 & & & \\
20 & & & \\
30 & & & \\
40 & & & \\
50 & & & \\
60 & & & \\
70 & & & \\
80 & & & \\
90 & & & \\
100 & & & \\
110 & & & \\
120 & & & \\
\hline
\end{array}
\]

\[
\begin{array}{cc}
P1 & P2 & P3 & P4 \\
0 & & & \\
10 & & & \\
20 & & & \\
30 & & & \\
40 & & & \\
50 & & & \\
60 & & & \\
70 & & & \\
80 & & & \\
90 & & & \\
100 & & & \\
110 & & & \\
120 & & & \\
\hline
\end{array}
\]

P1 and P2 arrive  P3 arrives  P4 arrives
Consider a **Round Robin** \((q = 10)\) scheduler

\[
P_1: t_1 = 60 \text{ (arrival at } 0 - \mu) \\
P_2: t_2 = 10 \text{ (arrival at } 0) \\
P_3: t_3 = 30 \text{ (arrival } 30) \\
P_4: t_4 = 20 \text{ (arrival at } 50) \\
\mu = \text{ an extremely small unit of time}
\]

1. **Draw Gantt Chart**

   ![Gantt Chart](image-url)

   - P1 and P2 arrive
   - P3 arrives
   - P4 arrives
Consider a **Round Robin** \((q = 10)\) scheduler

\[P_1: t_1 = 60 \text{ (arrival at } 0 - \mu)\]

\[P_2: t_2 = 10 \text{ (arrival at } 0)\]

\[P_3: t_3 = 30 \text{ (arrival at } 30)\]

\[P_4: t_4 = 20 \text{ (arrival at } 50)\]

\(\mu = \text{an extremely small unit of time}\)

1. **Draw Gantt Chart**

   **At time 50:**

   P1 | P4 | P3

   ![Gantt Chart Diagram]

   **P1 and P2 arrive**  **P3 arrives**  **P4 arrives**
Q. 05

2. Compute the following quantities:
   – throughput \( \frac{120}{4} = \frac{1}{?} \Rightarrow ? = \frac{1}{30} \)
   – CPU utilization 100%
   – average service time (time spent executing on the CPU)
   – average wait time (time spent ready waiting for CPU)
   – average turnaround time (total average time spent waiting + executing)
2. Compute the following quantities:
   - throughput \( \frac{120}{4} = 1 / ? \Rightarrow ? = 1 / 30 \)
   - CPU utilization 100%
   - average service time (time spent executing on the CPU)

   \[
   \text{P1: 60} \quad \text{P2: 10} \quad \text{P3: 30} \quad \text{P4: 20}
   \]

   Average: \( \frac{60 + 10 + 30 + 20}{4} = 30 \)
Q. 05

2. Compute the following quantities:
   - throughput \( \frac{120}{4} = 1 / \, \text{?} \rightarrow \, ? = 1 / 30 \)
   - CPU utilization 100%
   - average service time (time spent executing on the CPU)
   - average wait time (time spent ready waiting for CPU)

   \[ \text{P1: 60} \quad \text{P2: 10} \quad \text{P3: 30} \quad \text{P4: 30} \]

   Average: \( \frac{(60 + 10 + 30 + 30)}{4} = \frac{130}{4} = 32.5 \)

   Recall: For FCFS, the average wait time is 37.5
2. Compute the following quantities:
   - throughput  
     \[ \frac{120}{4} = \frac{1}{?} \Rightarrow ? = \frac{1}{30} \]
   - CPU utilization 100%
   - average service time (time spent executing on the CPU)
   - average wait time (time spent ready waiting for CPU)
   - average turnaround time (total average time spent waiting + executing)

   Turnaround = 30 (service time) + 32.5 (wait time) = 62.5