(1) Compare CISC and RISC processors for the following factors:

<table>
<thead>
<tr>
<th>Factors</th>
<th>CISC</th>
<th>RISC</th>
</tr>
</thead>
<tbody>
<tr>
<td># of available GPRs</td>
<td>Less (📸)</td>
<td>More (📸)</td>
</tr>
<tr>
<td>Size of the internal cache</td>
<td>Smaller (📸)</td>
<td>Larger (📸)</td>
</tr>
<tr>
<td>Program size (IC)</td>
<td>Smaller (📸)</td>
<td>Larger (📸)</td>
</tr>
<tr>
<td>CPI</td>
<td>Higher (📸)</td>
<td>Smaller (📸)</td>
</tr>
<tr>
<td>Clock cycle rate</td>
<td>Lower (📸)</td>
<td>Higher (📸)</td>
</tr>
</tbody>
</table>

Note:
- # of available GPRs: which one has more registers("more" or "less")
- Size of the internal cache: which one has bigger L1 cache (bigger or smaller)
- Program size (IC): which one has bigger program size (bigger or smaller)
- CPI: which one has a larger CPI (larger or smaller)
- Clock cycle rate: which one has a higher CPI (higher or lower)

(2) What is the trade-off problem in the memory hierarchy?

The speed (memory access time) vs. unit price ($ amount per byte) – the higher in the memory hierarchy we go, the shorter the memory access latency we get and the higher the memory unit price we need to pay.
(3) How does “external memory fragmentation” happen?

As programs get started and leave memory (when they are completed), small unusable gaps appear at many different locations in memory, resulting considerable amount of unusable memory space.

(4) What are the two advantages in “virtual memory”?

The followings are the two primary advantages in using virtual memory:

(a) Programs larger than the physical memory capacity can be executed (or “more programs than the physical memory can hold can be executed”).

(b) Programs (especially large programs or programs that need large data) can be started quicker*.

* note: “programs are executed faster” is a misconception (programs will NOT be executed faster – they will be actually slower using virtual memory because of possible page faults – they just start sooner using “demand paging” in virtual memory).

(5) What is “valid flag” used in virtual memory for?

Valid flags are used to keep track of if a virtual memory page is currently mapped to a memory page in the physical memory. If the valid flag is ‘0’, the virtual memory page is not mapped to the physical memory page that appears in the VMT. If the valid flag is ‘1’, the virtual memory page is mapped to the physical memory page that appears in the VMT.