(1) What is the major issue about “memory subsystem”?

The speed of memory access (also known as “access time”, “access latency”, or “memory response time”), has not improved as much as processors’ speed has increased. This made the performance gap between the memory and a processor grows, making the memory to be primary performance bottleneck.

Note: The meaning of “performance” should be defined/declared/explained for full credit.

(2) Why isn’t “memory compaction” an effective solution to deal with external fragmentation (at least two different reasons)?

① While a memory compaction is in progress, programs that are being relocated will be stopped. While the timing of memory compactions is usually not controlled by users, the unpredictable “freezing” of programs can cause serious consequences to (human) users.

② After a compaction, programs that are put “back to back” in the main memory can no longer grow in their program size. Since each program usually holds some data, while a program sometime generate new data (while a program is executing), the inability of increasing program size is an unrealistic limitation to many programs.

(3) What is “external (memory) fragmentation”?

The external memory fragmentation is unused (small) memory space between two programs (or two data sets) loaded to the main memory.

Note: where (or how) “unused memory space” should be explained or mentioned for full credit (i.e., just saying “unused memory space” is not good enough, since it does not distinguish “external memory fragmentation” from “internal memory fragmentation”).
(4) What is “page fault” (make sure to answer this question using the appropriate technical term(s) the CS286 lectures already covered)?

A page fault is a situation where a virtual memory page requested by a processor is not mapped to any physical memory page.

(5) What are the primary advantages in using “virtual memory (using demand paging)”?
Mention at least three different advantages in using “virtual memory”.

1. Using the on-demand paging, programs larger than the size of the physical memory can be executed.
2. Using the on-demand paging, large programs (or large data sets) can be started quickly.
3. Using the memory paging, the external memory fragmentation will be eliminated (we will not waste memory space because of the external memory fragmentation).