I) Imagine that you are implementing a file system. The main abstractions in your design would be files and directories. Directories can contain zero or more files or directories. You want to treat directories and files in a uniform way, e.g., both will have name and will provide operations to stream content in and out, and to list children.

a) What design pattern could be used to achieve this design? [1 mark]

Composite, Bridge.

b) Please explain your design by giving a class diagram. [5 marks]

Hint: Consider introducing an additional abstraction.

II) Imagine that you would like to implement utility programs such as ls and chmod that need to iterate over file/directory structures from the previous question and perform operations on the visited nodes. In your design, you would like to avoid the need to extend the interface of the classes representing the file/directory structure whenever you add a new utility program.

a) What design pattern could be used to achieve this design? [1 mark]

Visitor.

b) Please explain your design by giving a class diagram. [5 marks]