Q1. Is it permissible to use several import statements to import the same module? What would the goal be? Can you think of a situation where it would be beneficial?

**Ans:** A module is loaded only once, regardless of the number of times it is imported. This prevents the module execution from happening over and over again if multiple imports occur.

Q2. What are some of a module's characteristics? (Name at least one.)

**Ans:** A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.

Q3. Circular importing, such as when two modules import each other, can lead to dependencies and bugs that aren't visible. How can you go about creating a program that avoids mutual importing?

**Ans:** A simple solution is that sometimes both modules can just be merged into a single, larger module.

Another solution could have been to defer the import of module2 to import it only when it is needed.

Or put all imports in a central module, e.g. in \_\_init\_\_.py.

Q4. Why is \_ \_all\_ \_ in Python?

**Ans:** It declares the semantically "public" names from a module. If there is a name in \_\_all\_\_, users are expected to use it, and they can have the expectation that it will not change.

import \*

\_\_all\_\_ in a module, e.g. module.py:

\_\_all\_\_ = ['foo', 'Bar']

means that when you import \* from the module, only those names in the \_\_all\_\_ are imported

Q5. In what situation is it useful to refer to the \_ \_name\_ \_ attribute or the string '\_ \_main\_ \_'?

**Ans:** Before executing code, Python interpreter reads source file and define few special variables/global variables.

If the python interpreter is running that module (the source file) as the main program, it sets the special \_\_name\_\_ variable to have a value “\_\_main\_\_”.

Q6. What are some of the benefits of attaching a program counter to the RPN interpreter application, which interprets an RPN script line by line?

**Ans:** benefit of attaching a program counter, that It will keep count of every element in itself which will help much when RPN script will execute line by line.

Q7. What are the minimum expressions or statements (or both) that you'd need to render a basic programming language like RPN primitive but complete— that is, capable of carrying out any computerised task theoretically possible?

**Ans:** There is no specific rule for minimum expression or statement, it basically depends upon how much task that we need o complete by program and what are the functionalities/features we want to achieve via program.