## 1️⃣ Create a Simple Package

Create a package named `maths` with two modules:

- `add.py`
- `sub.py`

Each module should contain one function.
Write a `main.py` to use the package.

---

## 2️⃣ Use `__init__.py`

Create a package `shapes` with:

- `circle.py` → function `area(radius)`
- `square.py` → function `area(side)`

Expose both functions using `__init__.py` so they can be imported directly.

---

## 3️⃣ Import Specific Function

Create a package `greetings` with:

- `hello.py` → function `say_hello()`
- `bye.py` → function `say_bye()`

Import **only** `say_hello()` in `main.py`.

---

## 4️⃣ Package with Sub-Package

Create this structure:

```
school/
├── staff/
│   ├── teacher.py
│   └── __init__.py
└── __init__.py
```

Write a function `info()` in `teacher.py` and call it from `main.py`.

---

## 5 Use Relative Import (`.`)

Inside a package `animals`, create:

- `dog.py`
- `cat.py`

From `cat.py`, import a function from `dog.py` using relative import.

---

## 6 Use Parent Import (`..`)

Create:

```
company/
├── hr/
│   └── salary.py
└── finance/
    └── tax.py
```

From `salary.py`, import a function from `tax.py`.

---

## 7 Make Package Executable

Create a package `calculator` with:

- `add.py`
- `__main__.py`

Run the package using:

```
python -m calculator
```

---

## 8 Import Package as Alias

Create a package `tools` with a module `convert.py`.
Import the package using an alias in `main.py`.

---

## 9️⃣ What Will Be the Output?

Given:

```
# calculator/__init__.py
print("Calculator package loaded")
```

What happens when:

```
import calculator
```

Explain briefly.

---

## 🔟 Identify the Error

Given this code inside `main.py`:

```
from .add import add
```

Is this correct?
If not, why?