Q1. Which function is used to open a file? What are the different modes of opening a file?

Function used to open a file

The built-in **`open()`** function is used to open a file in Python.

```
file = open("sample.txt", "r")
```

File Opening Modes

Mode	Description
`r`	Read mode. Opens an existing file for reading. Error if file does not exist.
`w`	Write mode. Creates a new file or overwrites existing file.
`a`	Append mode. Adds data at the end of the file.
`x`	Exclusive creation mode. Fails if file already exists.
`r+`	Read and write mode. File must exist.
`w+`	Write and read mode. Overwrites existing file.
`a+`	Append and read mode.
`b`	Binary mode (used with other modes, e.g., `rb`, `wb`).
`t`	Text mode (default mode).

Q2. Why is `close()` function used? Why is it important?

The **`close()`** function is used to close an opened file.

Importance of closing a file

- Releases system resources

- Prevents data loss
- Ensures all data is properly written to the file
- Avoids file corruption

```
file.close()
```

Q3. Python program to create, write, close, open, and read a file

```
# Create and write to the file
file = open("data.txt", "w")
file.write("I want to become a Data Scientist")
file.close()

# Open and read the file
file = open("data.txt", "r")
content = file.read()
print(content)
file.close()
```

Q4. Explain read(), readline(), and readlines() with examples

1. read()

Reads the entire content of the file as a single string.

```
file = open("data.txt", "r")
print(file.read())
file.close()
```

---

2. readline()

Reads **one line at a time**.

```
file = open("data.txt", "r")
print(file.readline())
file.close()
```

---

3. readlines()

Reads all lines and returns them as a **list of strings**.

```
file = open("data.txt", "r")
print(file.readlines())
file.close()
```

Q5. Why is the `with` statement used with `open()`? What are its advantages?

The **`with` statement** is used for **automatic file handling**.

Advantages

- Automatically closes the file
- No need to call `close()`
- Cleaner and safer code
- Prevents memory leaks

```
with open("data.txt", "r") as file:
    print(file.read())
```

Q6. Explain write() and writelines() with examples

write()

Writes a **single string** to a file.

```
file = open("sample.txt", "w")
file.write("Hello Python")
file.close()
```

---

writelines()

Writes **multiple strings** (list of strings) to a file.

```
lines = ["Hello\n", "Welcome to Python\n", "File Handling\n"]

file = open("sample.txt", "w")
file.writelines(lines)
file.close()
```