```
"""
Types of Files:
1. Text Files
2. csv file (Comma Separated Values) -, tsv (tab seaparted values)
   Extension: .csv

3. excel file -
4. Images
5. Audio Files
6. Video Files
"""
from os import write

"""
Syntax:
with open(<file_path>, [mode], [encoding]) as <alias_name>:
    # Body
"""


"""
path:
Location where the file is saved.

Types of Paths:
1. Relative Path (Path based on the current file)

./ => Current Directory
../ => Parent Directory

2. Absolute Path (Path from the root directory)


mode:
r = Read
    Reads the content from an existing file.
    If the file doesn't exist, raises a FileNotFoundError.

w = Write
    Creates a New File and Writes Content to the file.
    If the file already exists, overwrites the existing contents
    with new contents.

a = append
    Creates a New File and Writes the Contents to the file.
    If the file already exists, Adds the new Contents
    to the existing contents.

x = Creates a New File and Writes the Contents to the file.
    If the file already exists, raises FileExistsError.

rb = read binary (read Binary Files)
wb = write Binary (write Binary Content)
ab = append Binary (append Binary Content)
r+

In-built methods:
1. read() - reads the whole file content and returns a string
2. readline() - reads a single line in the file.
```

```
3. readlines() - reads the whole file and returns a
                 list of all the lines.

1. write() - accepts only a string and writes the string to the file
2. writelines() - accepts an iterable and writes the items in the iterable to the
file.

seek() - takes the file pointer to the specified position.
tell() - prints the current line.
"""


fp = open(".\\output.txt", "r")
content = fp.read()
fp.close()
print("File Contents are", content)


with open(".\\output.txt", "r") as fp:
    data = fp.read()
print(data)
print()

with open(".\\output.txt") as fp:
    data = fp.read()
print(data)
print()


with open("D:\\Full Stack\\Backend\\Python\\PythonJune2025\\output.txt", "r") as
fp:
    data = fp.read()
    print("Reading using Absolute Path...")
    print(data)


with open(".\\output1.txt", "w") as fp:
    fp.write("Writing to a File using Python.....")


"""
with open(".\\skdjskjdffs.txt", "r") as fp:
    data = fp.read()
print(data)
"""


with open(".\\output.txt", "w") as fp:
    fp.write("Line 85")
    fp.write("Line 86")
    fp.write("Line 87")


with open(".\\output2.txt", "a") as fp:
    fp.write("Appending Content to a File...")


with open(".\\output.txt", "a") as fp:
    fp.write("Appending Content to a File...")
```

```
"""
with open(".\\output3.txt", "x") as fp:
    fp.write("Writing Content to a File...")


with open(".\\output.txt", "x") as fp:
    fp.write("Writing Content to a File...")



with open(".\\output.txt", "r") as fp:
    fp.write("Trying to write to a file by opening in read mode...")
"""


with open(".\\output.txt", "r") as fp:
    contents1 = fp.read()

with open(".\\output1.txt", "r") as fp:
    contents2 = fp.read()

with open(".\\output2.txt", "r") as fp:
    contents3 = fp.read()

with open(".\\output3.txt", "r") as fp:
    contents4 = fp.read()

with open(".\\output4.txt", "w") as fp:
    fp.write(contents1 + '\n' + contents2 + '\n' + contents3 + '\n' + contents4)


def write_multiple_file_paths(*file_paths, destination_file_path):
    contents = ""
    for file_path in file_paths:
        with open(file_path, "r") as fp:
            contents += fp.read() + '\n'

    with open(destination_file_path, "w") as fp:
        fp.write(contents)

write_multiple_file_paths("output.txt", "output1.txt", "output2.txt",
"output3.txt", destination_file_path="output5.txt")


with open("./file_io_methods.txt", "r") as fp:
    first_line = fp.readline()
    print(first_line)
    second_line = fp.readline()
    print(second_line)
    third_line = fp.readline()
    print(third_line)


with open("./file_io_methods.txt", "r") as fp:
    lines = fp.readlines()
print(lines)
```

```
l1 = ["Line1\n", "Line2\n", "Line3\n", "Line4\n", "Line5\n"]
with open("./file_io_methods1.txt", "w") as fp:
    fp.writelines(l1)

with open("file_io_methods.txt", "a") as fp:
    fp.write("Adding New Content....")


with open("file_io_methods.txt", "a") as fp:
    content = fp.seek(5)
    print(content)
    content = fp.tell()
    print(content)
    # fp.write("Adding New Content in the middle....")
```