In [7]:
```python
import os

# 1. Create a text file and check if exists
def create_text_file(filename):
    try:
        if os.path.exists(filename):
            print(f"File '{filename}' already exists.")
        else:
            with open(filename, "w") as f:
                f.write("File created successfully.\n")
            print(f"File '{filename}' created.")
    except Exception as e:
        print("Error:", e)
create_text_file("demo.txt")
```

```
File 'demo.txt' created.
```

In [8]:
```python
# 2. Count vowels and consonants
def count_vowels_consonants(filename):
    vowels, consonants = 0, 0
    try:
        with open(filename, "r") as f:
            text = f.read().lower()
        for ch in text:
            if ch.isalpha():
                if ch in "aeiou":
                    vowels += 1
                else:
                    consonants += 1
        print("Vowels:", vowels, "Consonants:", consonants)
    except FileNotFoundError:
        print("File not found!")
count_vowels_consonants("demo.txt")
```

```
Vowels: 8 Consonants: 15
```

In [9]:
```python
# 3. Convert to uppercase
def convert_to_upper(filename):
    try:
        with open(filename, "r") as f:
```

```
            text = f.read()
        with open(filename, "w") as f:
            f.write(text.upper())
        print("Converted all letters to uppercase.")
    except Exception as e:
        print("Error:", e)
convert_to_upper("demo.txt")
```

```
Converted all letters to uppercase.
```

In [10]:
```
# 4. Count words starting with a letter
def count_words_starting_with(filename, letter):
    try:
        with open(filename, "r") as f:
            words = f.read().split()
        count = sum(1 for w in words if w.lower().startswith(letter.lower()))
        print(f"Words starting with '{letter}':", count)
    except Exception as e:
        print("Error:", e)
count_words_starting_with("demo.txt", "T")
```

```
Words starting with 'T': 0
```

In [11]:
```
# 5. Insert demo data (in reverse)
def insert_reverse_text(filename):
    demo_data = "Lorem Ipsum is simply dummy text of the printing and typesetting industry."
    reversed_text = demo_data[::-1]
    with open(filename, "w") as f:
        f.write(reversed_text)
    print("Inserted reversed demo data.")
insert_reverse_text("sample.txt")
```

```
Inserted reversed demo data.
```

In [12]:
```
# 6. Insert personal details
def insert_personal_details():
    name, age, mobile, addr, salary = "Rohan", 28, "9876543210", "New Delhi", 50000
    with open("details.txt", "w") as f:
        f.write(f"Name: {name}\nAge: {age}\nMobile: {mobile}\nAddress: {addr}\nSalary: {salary}")
    print("Details saved.")
insert_personal_details()
```

```
Details saved.
```

In [13]:
```python
# 7. Display words containing vowels
def read_vowel_words(filename):
    with open(filename) as f:
        words = f.read().split()
        vowel_words = [w for w in words if any(v in w.lower() for v in "aeiou")]
        print("Words with vowels:", vowel_words)
read_vowel_words("sample.txt")
```

```
Words with vowels: ['.yrtsudni', 'gnittesepyt', 'dna', 'gnitnirp', 'eht', 'fo', 'txet', 'ymmud', 'ylpmis', 'si', 'mu
spI', 'meroL']
```

In [14]:
```python
# 8. Read file in reverse order
def read_reverse(filename):
    with open(filename) as f:
        text = f.read()[::-1]
        print("Reversed file content:\n", text)
read_reverse("details.txt")
```

```
Reversed file content:
 00005 :yralaS
ihleD weN :sserddA
0123456789 :eliboM
82 :egA
nahoR :emaN
```

In [15]:
```python
# 9. Delete file
def delete_file(filename):
    if os.path.exists(filename):
        os.remove(filename)
        print(f"Deleted '{filename}'.")
    else:
        print("File not found.")
delete_file("remove_me.txt")
```

```
File not found.
```

In [16]:
```python
# 10. Create 10 unique coupon codes
def coupon_codes(filename):
    codes = [f"CUPON{i:03d}" for i in range(1, 11)]
    with open(filename, "w") as f:
```

```
        f.write("\n".join(codes))
    print("10 coupon codes created.")
coupon_codes("coupons.txt")
```

```
10 coupon codes created.
```

In [17]:
```
# 11. Count words starting with T/t
def count_t_words(filename):
    with open(filename) as f:
        words = f.read().split()
        count = sum(1 for w in words if w.lower().startswith("t"))
    print("Words starting with T/t:", count)
count_t_words("demo.txt")
```

```
Words starting with T/t: 0
```

In [20]:
```
# 12. Count "the" occurrences
def count_the(filename):
    with open(filename) as f:
        text = f.read().lower().split()
        print("'the' appears:", text.count("the"))
count_the("details.txt")
```

```
'the' appears: 0
```

In [22]:
```
# 13. Display <4 char words
def display_short_words(filename):
    with open(filename) as f:
        small = [w for w in f.read().split() if len(w) < 4]
        print("Short words:", small)
display_short_words("sample.txt")
```

```
Short words: ['dna', 'eht', 'fo', 'si']
```

In [24]:
```
# 14. Replace 'J' with 'I'
def j_to_i(filename):
    with open(filename) as f:
        data = f.read().replace("J", "I")
    print("Corrected Data:\n", data)
j_to_i("details.txt")
```

```
Corrected Data:
 Name: Rohan
Age: 28
Mobile: 9876543210
Address: New Delhi
Salary: 50000
```

In [26]:
```python
# 15. Credit card generator
from datetime import datetime
def generate_card(holder):
    bank = "PUNJAB NATIONAL BANK"
    number = "1234 5678 9012 3456"
    valid_from = datetime.now().strftime("%m/%y")
    expiry = f"{int(valid_from.split('/')[0]):02d}/{int(valid_from.split('/')[1])+5}"
    cvv = 321
    with open("card.txt", "w") as f:
        f.write(f"Bank: {bank}\nAccount Holder: {holder}\nCard Number: {number}\nValid From: {valid_from}\nExpiry D
    print("Card generated.")
generate_card("Sandeep")
```

```
Card generated.
```

In [27]:
```python
# 16. Random names with numb
def names_with_number():
    names = ["Dev", "Ajay", "Aman", "Shubham"]
    data = [f"{n.upper()}-{1000+i*111}" for i, n in enumerate(names)]
    print("Generated List:", data)
names_with_number()
```

```
Generated List: ['DEV-1000', 'AJAY-1111', 'AMAN-1222', 'SHUBHAM-1333']
```

In [28]:
```python
# 17. Employee files
employees = [
    {"name": "Amit", "id": "E201", "department": "HR", "email": "amit@abc.com"},
    {"name": "Priya", "id": "E202", "department": "IT", "email": "priya@abc.com"}
]
def create_employee_files():
    for e in employees:
        fname = e["name"].replace(" ", "_") + ".txt"
        with open(fname, "w") as f:
            f.write(f"Welcome {e['name']}!\nID: {e['id']}\nDept: {e['department']}\nEmail: {e['email']}")
```

```
    print("Employee files created.")
create_employee_files()
```

```
Employee files created.
```

In [29]:
```
# 18. Candidate scheduler (static version)
def candidate_files():
    candidates = ["Ravi", "Neha"]
    for i, c in enumerate(candidates, 1):
        cid = f"CAND{i:04d}"
        meet_time = f"{9+i}:00 AM"
        meet_id = f"10000{i}"
        with open(f"{c}.txt", "w") as f:
            f.write(f"Candidate: {c}\nID: {cid}\nMeeting Time: {meet_time}\nMeeting ID: {meet_id}\nPlatform: Zoom")
    print("Candidate files created.")
candidate_files()
```

```
Candidate files created.
```

In [30]:
```
# 19. Replace char in file
def replace_char(infile, outfile, old, new):
    with open(infile) as f:
        data = f.read().replace(old, new)
    with open(outfile, "w") as f:
        f.write(data)
    print(f"Replaced '{old}' with '{new}'.")
replace_char("demo.txt", "demo_new.txt", "A", "@")
```

```
Replaced 'A' with '@'.
```

In [31]:
```
# 20. Student classification
students = {
    "Ravi": {"marks": 75}, "Priya": {"marks": 29},
    "Aman": {"marks": 45}, "Sita": {"marks": 18}
}
def classify():
    passf = open("pass_students.txt", "w")
    failf = open("fail_students.txt", "w")
    for n, d in students.items():
        if d["marks"] >= 33:
            passf.write(f"{n}: {d['marks']}\n")
        else:
```

```
            failf.write(f"{n}: {d['marks']}\n")
    passf.close(); failf.close()
    print("Students classified.")
classify()
```

```
Students classified.
```

In [32]:
```
# 21. Student files
def student_files():
    names = ["Aman", "Rohan"]
    for i, n in enumerate(names, 1):
        with open(f"{n}.txt", "w") as f:
            f.write(f"Name: {n}\nRoll: {100+i}\nAge: {18+i}\nAddress: Delhi")
    print("Created individual student files.")
student_files()
```

```
Created individual student files.
```

In [33]:
```
# 22. Copy file
def copy_file(src, dest):
    with open(src) as f1, open(dest, "w") as f2:
        f2.write(f1.read())
    print("File copied successfully.")
copy_file("demo.txt", "copy_demo.txt")
```

```
File copied successfully.
```

In [34]:
```
# 23. Word frequency
def word_freq(filename):
    freq = {}
    with open(filename) as f:
        for word in f.read().split():
            freq[word] = freq.get(word, 0) + 1
    with open("word_freq.txt", "w") as out:
        for k, v in freq.items():
            out.write(f"{k}: {v}\n")
    print("Word frequencies saved.")
word_freq("demo.txt")
```

```
Word frequencies saved.
```

In [35]:
```python
# 24. Split large file
def split_large_file(filename):
    with open(filename) as f:
        lines = f.readlines()
    for i in range(0, len(lines), 10):
        with open(f"part_{i//10+1}.txt", "w") as part:
            part.writelines(lines[i:i+10])
    print("File split completed.")
split_large_file("demo.txt")
```

```
File split completed.
```

In [36]:
```python
# 25. Search for word
def search_word(filename, word):
    with open(filename) as f:
        for i, line in enumerate(f, 1):
            if word.lower() in line.lower():
                print(f"Found '{word}' in line {i}")
search_word("demo.txt", "file")
```

```
Found 'file' in line 1
```

In [ ]:
```python
# 27

filename = "students.txt"

n = int(input("Enter number of students: "))
with open(filename, "w") as f:
    for _ in range(n):
        name = input("Enter student name: ")
        marks = input("Enter marks: ")
        f.write(f"{name} - {marks}\n")

print("\nStudent Data:")
with open(filename, "r") as f:
    print(f.read())
```

In [ ]:
```python
# 28

filename = input("Enter filename: ")
```

```
count = 0

with open(filename, "r") as f:
    for word in f.read().split():
        if word.isdigit():
            count += 1

print("Total numeric values:", count)
```

In [ ]:

```
# 29

import pickle

numbers = [int(input(f"Enter number {i+1}: ")) for i in range(5)]

# Write binary file
with open("numbers.dat", "wb") as f:
    pickle.dump(numbers, f)

# Read binary file
with open("numbers.dat", "rb") as f:
    data = pickle.load(f)

print("Numbers read from binary file:", data)
```

In [ ]:
```python
# 30

import pickle

numbers = [int(input(f"Enter number {i+1}: ")) for i in range(5)]

# Write binary file
with open("numbers.dat", "wb") as f:
    pickle.dump(numbers, f)

# Read binary file
with open("numbers.dat", "rb") as f:
    data = pickle.load(f)

print("Numbers read from binary file:", data)
```

In [ ]:
```python
# 31

replacements = {"Python": "Java", "good": "excellent"}
input_file = input("Enter input filename: ")
output_file = "modified.txt"

with open(input_file, "r") as f:
    text = f.read()

for old, new in replacements.items():
    text = text.replace(old, new)

with open(output_file, "w") as f:
    f.write(text)

print("Words replaced and saved in modified.txt")
```

In [ ]:
```python
# 32

replacements = {"Python": "Java", "good": "excellent"}
input_file = input("Enter input filename: ")
output_file = "modified.txt"
```

```
with open(input_file, "r") as f:
    text = f.read()

for old, new in replacements.items():
    text = text.replace(old, new)

with open(output_file, "w") as f:
    f.write(text)

print("Words replaced and saved in modified.txt")
```

In [ ]:
```
# 33

filename = input("Enter filename: ")
with open(filename, "r") as f:
    lines = f.readlines()

longest = max(lines, key=len)
shortest = min(lines, key=len)

print("Longest Line:", longest.strip())
print("Shortest Line:", shortest.strip())
```

In [ ]:
```
# 34

from datetime import datetime

with open("log.txt", "a") as f:
    f.write(datetime.now().strftime("%Y-%m-%d %H:%M:%S") + "\n")

print("Timestamp added to log.txt")
```

In [ ]:
```
# 35

filename = input("Enter filename: ")

with open(filename, "r") as f:
    lines = f.readlines()

unique = list(set(lines))
```

```
with open(filename, "w") as f:
    f.writelines(unique)

print("Duplicate lines removed.")
```

In [ ]:
```
# 36

employees = {
    "Alice": 50000,
    "Bob": 70000,
    "Charlie": 60000,
    "David": 55000
}

sorted_employees = sorted(employees.items(), key=lambda x: x[1], reverse=True)

with open("employee_report.txt", "w") as f:
    for name, salary in sorted_employees:
        f.write(f"{name} - {salary}\n")

print("Report saved in employee_report.txt")
```

In [ ]:
```
# 37

files = input("Enter filenames separated by space: ").split()
with open("merged.txt", "w") as outfile:
    for fname in files:
        with open(fname, "r") as infile:
            outfile.write(infile.read() + "\n")

print("All files merged into merged.txt")
```

In [ ]:
```
# 38

import re

filename = input("Enter filename: ")

with open(filename, "r") as f:
```

```
    text = f.read()

emails = re.findall(r"[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-z]{2,}", text)

with open("emails.txt", "w") as f:
    f.write("\n".join(emails))

print("Emails extracted and saved in emails.txt")
```

In [ ]:

```
# 39

filename = input("Enter filename: ")

upper = lower = digits = special = 0

with open(filename, "r") as f:
    for ch in f.read():
        if ch.isupper():
            upper += 1
        elif ch.islower():
            lower += 1
        elif ch.isdigit():
            digits += 1
        elif not ch.isspace():
            special += 1

print(f"Uppercase: {upper}")
print(f"Lowercase: {lower}")
print(f"Digits: {digits}")
print(f"Special: {special}")
```

In [ ]:
```python
# 40

import random

with open("random_numbers.txt", "w") as f:
    for _ in range(50):
        f.write(str(random.randint(10, 100)) + "\n")

print("50 random numbers saved in random_numbers.txt")
```

In [ ]:
```python
# 41

file1 = input("Enter first file: ")
file2 = input("Enter second file: ")

with open(file1, "r") as f1, open(file2, "r") as f2:
    words1 = set(f1.read().split())
    words2 = set(f2.read().split())

common = words1 & words2
print("Common words:", common)
```

In [ ]:
```python
# 42

import json

with open("data.json", "r") as f:
    data = json.load(f)

with open("data.txt", "w") as f:
    for key, value in data.items():
        f.write(f"{key}: {value}\n")

print("Converted data.json → data.txt")
```

In [ ]:
```python
# 43

def encrypt(text):
    return "".join(chr(ord(c) + 3) for c in text)
```

```
def decrypt(text):
    return "".join(chr(ord(c) - 3) for c in text)

choice = input("Encrypt (E) or Decrypt (D)? ").upper()
filename = input("Enter filename: ")

with open(filename, "r") as f:
    content = f.read()

if choice == "E":
    result = encrypt(content)
    with open("encrypted.txt", "w") as f:
        f.write(result)
    print("File encrypted → encrypted.txt")
else:
    result = decrypt(content)
    with open("decrypted.txt", "w") as f:
        f.write(result)
    print("File decrypted → decrypted.txt")
```

In [ ]:
```
# 44

filename = input("Enter filename: ")
word = input("Enter word to delete lines containing it: ")

with open(filename, "r") as f:
    lines = f.readlines()

with open(filename, "w") as f:
    for line in lines:
        if word not in line:
            f.write(line)

print("Lines containing the word deleted.")
```

In [ ]:
```
# 45

filename = input("Enter filename: ")
```

```
with open(filename, "r") as f:
    lines = f.readlines()

num_lines = len(lines)
num_words = sum(len(line.split()) for line in lines)
num_chars = sum(len(line) for line in lines)
avg_words = num_words / num_lines if num_lines else 0

print(f"Total Lines: {num_lines}")
print(f"Total Words: {num_words}")
print(f"Total Characters: {num_chars}")
print(f"Average Words per Line: {avg_words:.2f}")
```

In [ ]:

```
# 46

import os
import shutil

for file in os.listdir():
    if file.endswith(".txt") and not file.endswith("_backup.txt"):
        shutil.copy(file, file.replace(".txt", "_backup.txt"))

print("Backup files created.")
```

In [ ]:
```python
# 47

from collections import Counter

filename = input("Enter filename: ")

with open(filename, "r") as f:
    words = f.read().split()

top_words = Counter(words).most_common(10)

with open("top_words.txt", "w") as f:
    for word, count in top_words:
        f.write(f"{word}: {count}\n")

print("Top 10 words saved in top_words.txt")
```

In [ ]:
```python
# 48

import csv

with open("marks.csv", "r") as f:
    reader = csv.DictReader(f)
    with open("marks_report.txt", "w") as out:
        for row in reader:
            total = sum(int(v) for v in row.values() if v.isdigit())
            percentage = total / (len(row) - 1)
            out.write(f"{row['Name']}: Total = {total}, Percentage = {percentage:.2f}%\n")

print("Report saved to marks_report.txt")
```

In [ ]:
```python
# 49

import os
from datetime import date

today = date.today().strftime("%Y-%m-%d")

for file in os.listdir():
```

```
    if file.endswith(".txt"):
        os.rename(file, f"{today}_{file}")

print("All .txt files renamed with today's date prefix.")
```

In [ ]:
```
# 50

import os
from datetime import datetime

for file in os.listdir():
    if file.endswith(".txt"):
        size = os.path.getsize(file) / 1024
        created = datetime.fromtimestamp(os.path.getctime(file))
        print(f"{file} - {size:.2f} KB - Created: {created}")
```