1.Write a python function that copies file reading and writing upto 50 characters at a time

In [6]:
```
def copy_file():
    f=open('demo.txt','r')
    count=0
    content=f.read()
    raw=[]
    for i in content:
        if count<50:
            raw.append(i)
            count+=1
        print(content.read)
    f.close()
copy_file()
```

2.Print all numbers present in a text file and print number of blank spaces present in that file

In [36]:
```
f1=(open('first.txt','r'))
content=f1.read()
raw1=[]
count1=0
count2=0
for i in content:
    raw1.append(i)
    if i==' ':
        count1+=1
import re
numbers=re.findall(r'\d+',content)
print('Numbers are : ', numbers)


print('Number of spaces are ',count1)
```

```
Numbers are :  []
Number of spaces are  3
```

3. Write a function called sed that takes as argument a pattern string, a replacement string, and two fienames;. It should read the file first and write the contents into the second file. If the

pattern string appears anywhere in the file, it should be replaced with a replacement string. If an error occurs while opening, reading, writing, or closing strings, your program should catch the exception, print an error message, and exit

In [51]:
```
file=open('input.txt','w')
file.write('Allah is Omnipotent, and He is Above anything else!')

file.close()
```

In [52]:
```
file1=open('output.txt','w')
file1.write('The Prophet (PBUH) was the best human in realizing the denotation

file1.close()
```

In [65]:
```
def sed(pattern, replacement, in_f, out_f):
   try:
        with open(in_f, "r") as infile, open(out_f, "w") as outfile:
         for line in infile:
             modified_line=line.replace(pattern, replacement)
             outfile.write(modified_line)
   except Exception as e:

       print("An error occurred:", str(e))

sed("old pattern", "new pattern", "input.txt", "output.txt")
```

4.Lock file analysis you have a lot well containing records of user activities on a website each line in the file represents a long entry with details like a time stamp user id and action performed your task is to analyse this log file

1.write python go to read the lock file and extract specific informati on such as a number of unique users of the most common actions
2.how would you handle files efficiently without without loading the e ntire file into memory

In [28]:
```
#Function to parse a log file and extract relevant information

def parse_log_file(log_file_path):

# Create a dictionary to store Log entries by date

    log_entries_by_date = {}

    with open(log_file_path, 'r') as log_file:

        for line in log_file:


#Split each Line into timestamp and message
            parts= line.strip().split(' ', 1)

            if len(parts) == 2:

                timestamp, message = parts

                date =timestamp[:10] # Extract the date portion

                if date in log_entries_by_date:

                    log_entries_by_date [date].append(message)

                else:

                    log_entries_by_date[date] = [message]

    return log_entries_by_date

 # Function to analyze Log data (you can customize this based on your needs)

def analyze_log_data(log_entries_by_date):


    for date, entries in log_entries_by_date.items():

        print("Date: (date)")

        print("Total Entries: (lep(entries)}")

        print("Sample Entries:")

        for i, entry in enumerate(entries[:5], start=1):

            print("(i). (entry)")

        print("------")
```

In [29]:
```
if __name__=='__main__':
    log_file_path = 'C:/Users/admin/Desktop/Practice Exercises/Python-s1/Day-3/
    # Replace with the path to your log file
    log_entries_by_date = parse_log_file(log_file_path)
    analyze_log_data(log_entries_by_date)
```

```
Date: (date)
Total Entries: (lep(entries)}
Sample Entries:
(i). (entry)
(i). (entry)
(i). (entry)
(i). (entry)
(i). (entry)
------
```

5 write a python code to search for and replace with in the text file

In [68]:
```
f4=open('large_file.txt','w')
f4.write('Our Prophet Muhammad (sallallahu ‘alayhi wa sallam) said: “The seven
```

Out[68]: 160

In [81]:
```
file_path = 'large_file.txt'
search_text = (input('Enter the text that you want to replace'))
replace_text = 'Great'
with open(file_path, 'r') as file:
    file_content = file.read()
updated_content = file_content.replace(search_text, replace_text,4)
with open(file_path, 'w') as file:
    file.write(updated_content)
print(f"Text '{search_text}' has been replaced with '{replace_text}' in '{file_
```

```
Enter the text that you want to replacethe
Text 'the' has been replaced with 'Great' in 'large_file.txt'.
```

In [84]:
```
##multiple replacements
import re
file_path = 'large_file.txt'
replacements = {
    'is': 'am',
    'for': 'that'
}
with open(file_path, 'r') as file:
    file_content = file.read()
def multiple_replace(text, replacements):
    pattern = re.compile("|".join([re.escape(key) for key in replacements.keys(
    return pattern.sub(lambda x: replacements[x.group()], text)
updated_content = multiple_replace(file_content, replacements)
with open(file_path, 'w') as file:
    file.write(updated_content)
print("Multiple replacements have been performed in '{file_path}'.")
```

```
Multiple replacements have been performed in '{file_path}'.
```

6.write a python script that concatenates the contents of multiple text files in to a single output file.Allowthe user to specify output file and input file

In [14]:
```
def concatenate_files (input_files, output_file):
    try:
        with open(output_file, 'w') as output:
            for input_file in input_files:
                with open(input_file,'r' ) as file:
                    output.write(file.read())
        print(f"Concatenated {len(input_files)} files into {output_file}")
    except Exception as e:
        print("An error occurred: (str(e)}")
if __name__=="__main__":
    input_files = []
    while True:

        file_name = input("Enter an input file (or type 'empty' to finish): ")
        if file_name.lower() == 'empty':
            break
        input_files.append(file_name)

    output_file = input("Enter the output file name: ")

    concatenate_files (input_files, output_file)
```

```
Enter an input file (or type 'empty' to finish): empty
Enter the output file name: tauheer
Concatenated 0 files into tauheer
```

7.You are given a test file memory input.txt containing a list of one word for line you are a talkies to create a python program that tell you the contents of input.t processes the words and the rights the result to another or put file named output.txt

In [20]:
```
def process_words(input_file, output_file):

    try:
        word_length = {}

        with open(input_file, "r") as inp:
            words=inp.read().split()

            for word in words:
                word_length[word] = len(word)

        with open(output_file, 'w') as out:
             for word, length in word_length.items():
                out.write(f" {word}: {length}\n")

        print(f"Word lengths written to '{output_file}'' successfully.\n")

    except FileNotFoundError:
        print(f"Error: File '{input_file}' not found.\n")

    except Exception as e:
        print("An error occurred: {e}\n")

input_file = "input.txt"

output_file ='output.txt'

process_words(input_file, output_file)

with open(output_file, 'r') as f:
    data = f.read()
print(data)
```

```
Word lengths written to 'output.txt'' successfully.

 Allah: 5
 is: 2
 Omnipotent,: 11
 and: 3
 He: 2
 Above: 5
 anything: 8
 else!: 5
```

In [ ]:

In [ ]:

In [ ]:

8.Assume that you are developing a student gradebook system for a school. The system should allow teachers to input student grades for various subjects, store the data in files, and provide students with the ability to view their grades.

Design a Python program that accomplishes the following tasks.
1.teacher should be able to input grades for students in different sub jects.
2.store the student grade data in separate text files for each subjec t.
3. subjects should be able to view their grades for each subject.
4.Implement error handling for file operations such as file not found or permission issues

In [8]:
```
import os


def input_grades(student_id, subject, grade):
    filename = f"{subject}_grades.txt"
    with open(filename, 'a') as file:
        file.write(f"{student_id}: {grade}\n")

def view_grades(student_id, subject):
    filename = f"{subject}_grades.txt"
    with open(filename, 'r') as file:
        lines = file.readlines()
        student_grades = [line.strip() for line in lines if line.startswith(f"{
        if student_grades:
            print(f"Grades for {student_id} in {subject}:")
            for grade in student_grades:
                print(grade)
        else:
            print(f"No grades found for {student_id} in {subject}.")


while True:
    print("Gradebook Menu:")
    print("1. Input Grades")
    print("2. View Grades")
    print("3. Exit")
    choice = input("Enter your choice (1/2/3): ")

    if choice == "1":
        student_id = input("Enter Student ID: ")
        subject = input("Enter Subject: ")
        grade = input("Enter Grade: ")
        input_grades(student_id, subject, grade)
    elif choice == "2":
        student_id = input("Enter Student ID: ")
        subject = input("Enter Subject: ")
        view_grades(student_id, subject)
    elif choice == "3":
        print("Exiting.")
    break
```

```
Gradebook Menu:
1. Input Grades
2. View Grades
3. Exit
Enter your choice (1/2/3): 3
Exiting.
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: