Darshan UNIVERSITY

योगः कर्मसु कौशलम्

Python Programming - 2301CS404

Lab - 9

Dhol Namra

Enroll:23010101407

22-01-2025

File I/O

01) WAP to read and display the contents of a text file. (also try to open the file in some other directory)

- in the form of a string

- line by line

- in the form of a list

In [7]:
```
fp = open("abc.txt","r")
print(fp.read())
fp.close()
#Line by line
fp = open("abc.txt","r")
print(fp.readline())
fp.close()
# in the form of a list
fp = open("abc.txt","r")
print(fp.readlines())
fp.close()
```

```
asdfghjklqwertyuioj
sdfghjkl
asdfghjklqwertyuioj

['asdfghjklqwertyuioj\n', 'sdfghjkl']
```

02) WAP to create file named "new.txt" only if it doesn't exist.

In [1]:
```python
import os

filename = "new.txt"

if not os.path.exists(filename):
    with open(filename, "w") as file:
        print(f"File '{filename}' created successfully.")
else:
    print(f"File '{filename}' already exists.")
```

```
File 'new.txt' created successfully.
```

03) WAP to read first 5 lines from the text file.

In [21]:
```python
fp = open("new.txt","r")
for i in range(5):
        print(fp.readline())
fp.close()
```

```
sdfghj

asdfghj

sdfgh

sdfgh

aewertet
```

04) WAP to find the longest word(s) in a file

In [50]:
```python
f = open('new.txt','r')
s = f.read().split()
# st = list(s.split(' '))
# longest = ''
# for i in st:
#     if(len(longest) < len(i) ):
#         longest = i

# print(longest)
# fp.close()
longest=""
# print(s)
for i in s:
    if(len(longest) < len(i)):
        longest = i
print(longest)
```

```
aewertet
```

05) WAP to count the no. of lines, words and characters in a given text file.

In [56]:
```
fp = open("abc.txt","rt")
lines=fp.readlines()
num_lines = len(lines)
num_words = sum(len(line.split()) for line in lines)
num_chars = sum(len(line) for line in lines)
print(num_lines)
print(num_words)
print(num_chars)
fp.close()
```

```
6
7
50
```

06) WAP to copy the content of a file to the another file.

In [54]:
```
fp = open("new.txt","rt")
lines=fp.read()

fp = open("abc.txt","wt")
fp.write(lines)

fp.close()
fp.close()
```

07) WAP to find the size of the text file.

In [82]:
```
fp = open("abc.txt","rb")
size=len(fp.read())
print(size)
fp.close()
```

```
55
```

08) WAP to create an UDF named frequency to count occurances of the specific word in a given text file.

In [62]:
```
def count_word_in_file(words, word):
    count = words.count(word)
    return count
fp = open("abc.txt","rt")
data = fp.read()
word = "sdfgh"
words = data.split()
fp.close()

print(count_word_in_file(words, word))
```

```
2
```

09) WAP to get the score of five subjects from the user, store them in a file. Fetch those marks and find the highest score.

In [88]:
```
fp = open("score.txt", "w")
for i in range(1, 6):
    scores = input(f"Enter the score for subject {i}: ")
    fp.write(scores)
fp.close()

fp = open("score.txt", "r")
fp.readlines()
scores = [int(score.strip()) for score in scores]
highest_score = max(scores)
print(highest_score)
fp.close()
```

```
5
```

10) WAP to write first 100 prime numbers to a file named primenumbers.txt

(Note: each number should be in new line)

In [3]:
```
def is_prime(n):
    """Check if a number is prime."""
    if n < 2:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

def generate_primes(count):
    """Generate first 'count' prime numbers."""
    primes = []
    num = 2
    while len(primes) < count:
        if is_prime(num):
            primes.append(num)
        num += 1
    return primes


prime_numbers = generate_primes(100)

with open("primenumbers.txt", "w") as file:
    for prime in prime_numbers:
        file.write(str(prime) + "\n")

print("First 100 prime numbers have been written to 'primenumbers.txt'.")
```

```
First 100 prime numbers have been written to 'primenumbers.txt'.
```

11) WAP to merge two files and write it in a new file.

In [7]:
```
def merge_files(new, file2, output_file):
    """Merge contents of file1 and file2 into output_file."""
    with open(output_file, "w") as outfile:
        for file in [new, file2]:
            with open(new, "r") as infile:
                outfile.write(infile.read() + "\n")

# File names
new = "new.txt"
file2 = "file2.txt"
output_file = "merged.txt"

# Merge the files
merge_files(new, file2, output_file)

print(f"Files '{new}' and '{file2}' have been merged into '{output_file}'.")
```

```
Files 'new.txt' and 'file2.txt' have been merged into 'merged.txt'.
```

12) WAP to replace word1 by word2 of a text file. Write the updated data to new file.

In [11]:
```
def replace_word(new, output_file, word1, word2):
    """Replace all occurrences of word1 with word2 in input_file and save to out
    with open(input_file, "r") as infile:
        data = infile.read()

    updated_data = data.replace(word1, word2)

    with open(output_file, "w") as outfile:
        outfile.write(updated_data)

    print(f"Replaced '{word1}' with '{word2}' and saved to '{output_file}'.")


input_file = "new.txt"
output_file = "updated.txt"
word1 = "oldword"
word2 = "newword"

replace_word(input_file, output_file, word1, word2)
```

```
Replaced 'oldword' with 'newword' and saved to 'updated.txt'.
```

13) Demonstrate tell() and seek() for all the cases(seek from beginning-end-current position) taking a suitable example of your choice.

In [19]:
```
with open("new.txt", "w") as file:
    file.write("Hello, this is a test file.\nWelcome to file handling in Python.

with open("new.txt", "rb") as file:
    print(f"Initial Position: {file.tell()}")

    print("Reading 10 bytes:", file.read(10))
    print(f"Current Position: {file.tell()}")

    file.seek(0, 0)
```

```
    print(f"After seek(0, 0), Position: {file.tell()}")

    file.seek(5, 1)
    print(f"After seek(5, 1), Position: {file.tell()}")

    file.seek(-10, 2)
    print(f"After seek(-10, 2), Position: {file.tell()}")

    print("Remaining content:", file.read().decode("utf-8"))
```

```
Initial Position: 0
Reading 10 bytes: b'Hello, thi'
Current Position: 10
After seek(0, 0), Position: 0
After seek(5, 1), Position: 5
After seek(-10, 2), Position: 54
Remaining content: in Python.
```

In [ ]: