Darshan
UNIVERSITY
योग: कर्मसु कौशलम्

Python Programming - 2301CS404

Lab - 9

23010101161 - Smit Maru - 260

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 [1]:
```
fp = open('text.txt')
print(fp.read())
fp.close()
```

```
in the form of a string
linelinelineline by line
in the form of a list
in the form of a string
line by line
in the form of a list
```

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

In [3]:
```
fp = open('abc.txt','x')
print(fp.write("File is Create"))
fp.close()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
14
```

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

In [4]:
```
fp = open('text.txt')
data = fp.readlines()
for i in range(0,5):
    print(data[i][:len(data[i])-1])
fp.close()
```

```
in the form of a string
linelinelineline by line
in the form of a list
in the form of a string
line by line
```

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

In [5]:
```
fp = open('text.txt')
data = fp.readlines()
long = ''

for line in data:
    words = line.split()
    for word in words:
        if len(word) > len(long):
            long = word
print(long)
fp.close()
```

```
linelinelineline
```

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

In [35]:
```
fp = open('text.txt')
data = fp.readlines()
print(f"No. of lines = {len(data)}")

word_len = 0
char_Count = 0
for line in data:
    word_len += len(line.split())
    char_Count += len(line)
print(word_len)
print(char_Count)
fp.close()
```

```
No. of lines = 6
30
129
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

ontent of a file to the another file.

In [36]:
```
fp = open('text.txt')
fp2 = open('text-copy.txt','x')
fp2.write(fp.read())
fp.close()
fp2.close()
```

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

In [40]:
```
fp = open('text.txt')
fp.seek(0,2)
print(fp.tell())
```

```
129
```

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

In [ ]:
```
def occurances_of_char():
    fp = open('text.txt')
    data = fp.readlines()
    x = 
    char_count = 0
    for line in data:
        word_len = line.split()
        for j in word_len:
            if(x = j):
                char_Count += 1
    fp.close()
    print()
```

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 [6]:
```
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 [2]:

```
l1=[]
count = 0
while count < 100:
    num = 0
    for j in range(1,count):
        if(count%j==0):
            num+=1
    if(num==1):
        l1.append(count)
    count+=1
l2=[str(i)+"\n" for i in l1]
with open("primenumbers.txt", "w") as file:
    file.writelines(l2)
print(l1)
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79,
83, 89, 97]
```

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 [1]:

```
def replace_word(input_file, output_file, word1, word2):
    """Replace all occurrences of word1 with word2 in input_file and save to output
    with open(input_file, "r", encoding="utf-8") as infile:
        data = infile.read()

    updated_data = data.replace(word1, word2)

    with open(output_file, "w", encoding="utf-8") as outfile:
        outfile.write(updated_data)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
    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 [12]:

```
fp = open("text.txt","rb")
fp.read(5)
print(fp.tell())
fp.seek(0,2)
fp.seek(-4,1)
print(fp.tell())
fp.close()
```

```
5
125
```

In [ ]:

```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js