

Python Programming - 2301CS404

Lab - 10

Roll No : 352

Name : Smit Gohe1

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 [11]: # fp = open('first.txt', 'w')
# fp.write('smit\nsmit')

fp = open('first.txt', 'r')
first_line = fp.readline()
fp.seek(0)
all_data = fp.read()
fp.seek(0)
list_data = fp.readlines()

print('First line:', first_line)
print('Read data:', all_data)
print('List data:', list_data)

fp.close()
```

First line: smit

Read data: smit
smit

List data: ['smit\n', 'smit']

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

```
In [12]: fp = open('new.txt', 'w')
print('File Created')
fp.close()
```

File Created

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

```
In [14]: fp = open('first.txt', 'r')

for _ in range(6):
    print(fp.readline(),end='')

fp.close()
```

smit
smit
smit
smit
smit

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

```
In [21]: fp = open('new.txt', 'r')
file = fp.read()

list_data = file.split(' ')

print(f'Longest Word in file : {max(list_data)}')
fp.close()
```

Longest Word in file : versatility.

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

```
In [36]: fp = open('new.txt', 'r')
list_data = len(fp.read().split(' '))
fp.seek(0)

lines = len(fp.readlines())
fp.seek(0)

characters = len(fp.read())

print(f'No of Words : {list_data}')
print(f'No of Lines : {lines}')
print(f'No of Characters : {characters}')
fp.close()
```

No of Words : 14
No of Lines : 1
No of Characters : 117

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

```
In [39]: fp = open('new.txt', 'r')
data = fp.read()

fp2 = open('copy.txt', 'w')
fp2.write(data)
print('Data Copy')
fp.close()
fp2.close()
```

Data Copy

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

```
In [38]: fp = open('new.txt', 'r')
size = len(fp.read())
print(size)
```

117

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

```
In [79]: def name_freq(data, word):
        return data.count(word)

word = input('Enter word: ')

fp = open('new.txt', 'r')
data = fp.read()
res = name_freq(data, word)
fp.close()

print(res)
```

1

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 [77]: subjects = []
for i in range(5):
    mark = input(f'Enter marks of subject {i+1}')
    subjects.append(mark)

res = max(subjects)

fp = open('subjects.txt', 'w')

for mark in subjects:
    fp.write(str(mark) + '\n')

fp.write(str(res))
fp.close()
```

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

(Note: each number should be in new line)

```
In [73]: res = []
i = 2
while len(res) < 100:
    for j in range(2,i):
        if i % j == 0:
            break
    else:
        res.append(i)
    i+=1

fp = open('primenumbers.txt', 'w')

for i in res:
    fp.write(str(i) + '\n')

fp.close()
```

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

```
In [46]: fp = open('first.txt', 'r')
data = fp.read()

fp2 = open('new.txt', 'r')
data += fp2.read()

fp3 = open('merge.txt', 'w')
fp3.write(data)
print('data merge')

fp.close()
fp2.close()
fp3.close()
```

data merge

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

```
In [3]: # Open and read the original file
fp = open('new.txt', 'r')
data = fp.read()
fp.close()

old_word = input('Enter word to replace: ')
new_word = input('Enter new word: ')

updated_data = data.replace(old_word, new_word)

new_fp = open('updated.txt', 'w')
```

```
new_fp.write(updated_data)
new_fp.close()
```

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

```
In [2]: with open('test.txt', 'w') as f:
        f.write("ABCDEFGHJKLMNOPQRSTUVWXYZ")
        f.close()

# Open in BINARY mode to allow all seek operations
file = open('test.txt', 'rb') # Note: 'rb' for binary read

print("File content: ABCDEFGHJKLMNOPQRSTUVWXYZ")

# tell() - Get current position
print(f"Initial position: {file.tell()}")

# Read 5 characters (bytes in binary mode)
data = file.read(5).decode() # Decode bytes to string
print(f"Read: {data}, Position: {file.tell()}")

# seek from BEGINNING
file.seek(10, 0) # Go to 10th byte
print(f"seek(10, 0) - Position: {file.tell()}, Char: {file.read(1).decode()}")

# seek from CURRENT - NOW WORKS in binary mode
file.seek(5, 1) # Move 5 bytes forward from current
print(f"seek(5, 1) - Position: {file.tell()}, Char: {file.read(1).decode()}")

file.seek(-3, 1) # Move 3 bytes backward from current
print(f"seek(-3, 1) - Position: {file.tell()}, Char: {file.read(1).decode()}")

# seek from END
file.seek(0, 2) # Go to end
end_pos = file.tell()
print(f"seek(0, 2) - End position: {end_pos}")

file.seek(-5, 2) # Go 5 bytes before end
print(f"seek(-5, 2) - Position: {file.tell()}, Char: {file.read(1).decode()}")

file.close()
```

```
File content: ABCDEFGHJKLMNOPQRSTUVWXYZ
Initial position: 0
Read: ABCDE, Position: 5
seek(10, 0) - Position: 10, Char: K
seek(5, 1) - Position: 16, Char: Q
seek(-3, 1) - Position: 14, Char: O
seek(0, 2) - End position: 26
seek(-5, 2) - Position: 21, Char: V
```