Python Programming - 2301CS404

Lab - 11

Roll No : 352

Name : Smit Gohel

Progrmas based on Python Data Structure + UDF + File Handling

Practical 1: Count Lines, Words and Characters

Problem Definition:
Write a Python program to read text from a file and count the total number of lines, words, and characters. The result should be written to an output file.

In [51]:

```python
fp = open('file1.txt', 'r')
data = fp.read()
characters = len(data)
fp.seek(0)

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

words = [i for i in fp.read().split()]

print(f'Characters: {characters}')
print(f'Lines: {len(lines)}')
print(f'Words: {len(words)}')

fp.close()
```

```
Characters: 23
Lines: 2
Words: 4
```

Practical 2: Search a Word in a File

Problem Definition:

Write a Python program to search a given word in a text file and count its occurrences. Store the result in an output file.

In [38]:

```
fp = open('file1.txt', 'r')

word = input('Enter word: ')
data = fp.read().count(word)

fp1 = open('count.txt', 'w')
fp1.write(str(data))

fp.close()
fp1.close()
```

Practical 3: Student Marks and Average

Problem Definition:

Write a Python program to read roll number, name, and marks of students from a file. Store marks in a list, calculate the average using a user-defined function, and write student details along with average marks to an output file.

In [12]:

```
fp = open('students.txt', 'r')
outfile = open('avg_results.txt', 'w')

all_marks = []

for line in fp:
    line = line.strip()
    if line:
        parts = line.split()
        roll = parts[0]
        name = parts[1]
        marks = [int(parts[i]) for i in range(2, len(parts))]

        student_avg = sum(marks) / len(marks)

        marks_str = ' '.join(str(m) for m in parts[2:])
        outfile.write(f"{roll} {name} {marks_str} Average: {student_avg:.2f}\n")

fp.close()
outfile.close()

print("Results saved to avg_results.txt")
```

```
Results saved to avg_results.txt
```

Practical 4: Maximum and Minimum using Tuple

Problem Definition:

Write a Python program to read numbers from a file, store them in a tuple, and find the maximum and minimum values. Write the result to an output file.

In [110…

```
fp = open('values.txt')
t1 = tuple(num for num in fp.read().split())

fp1 = open('min_max.txt', 'w')
max_num = str(max(t1))
min_num = str(min(t1))

fp1.write(max_num)
fp1.write(min_num)

fp.close()
fp1.close()
```

Practical 5: Remove Duplicate Words using Set

Problem Definition:

Write a Python program to read words from a file and remove duplicate words using a set. Store the unique words in an output file.

In [118…

```
fp = open('file1.txt')
data = fp.read().split()

remove_dup = set()
for i in data:
    remove_dup.add(i)

fp1 = open('unique_num.txt', 'w')
for i in remove_dup:
    fp1.write(str(i))

fp.close()
fp1.close()
```

Practical 6: Word Frequency using Dictionary

Problem Definition:

Write a Python program to count the frequency of each word in a text file using a dictionary and write the result to an output file.

In [123…

```
fp = open('file1.txt')
data = fp.read().split()
print(data)
freq = {}
for i in data:
    freq[i]=0
for i in data:
    freq[i]=freq[i]+1
print(freq)
```

```
['hello', 'world', 'hello', 'world']
{'hello': 2, 'world': 2}
```

Practical 7: Student Result and Grade Calculation

Problem Definition:

Write a Python program to read roll number, name, and marks of students from a file. Use a user-defined function to determine grade (A/B/C/Fail) and write the student result to an output file.

Grade Criteria:

- Marks ≥ 75 : Grade A
- Marks ≥ 60 : Grade B
- Marks ≥ 40 : Grade C
- Marks < 40 : Fail

In [10]:

```python
def get_grade(marks):
    marks = float(marks)
    if marks >= 75:
        return "A"
    elif marks >= 60:
        return "B"
    elif marks >= 40:
        return "C"
    else:
        return "Fail"

infile = open('marks.txt', 'r')

outfile = open('results.txt', 'w')

outfile.write("Roll No\tName\t\tMarks\tGrade\n")
outfile.write("-" * 40 + "\n")

for line in infile:
        roll, name, marks = line.strip().split(',')

        grade = get_grade(marks)

        # Write result
        outfile.write(f"{roll:<10}{name:<15}{marks:<10}{grade}\n")

infile.close()
outfile.close()

print("Results saved to results.txt")
```

```
Results saved to results.txt
```