24BIT167 - VANDITA NAWANI Aim : Two write simple python programs to perfrom basic file handling operations including working with CSV, excel compatiblity, direct manipulation, text processing, file merging and serilzation of objects

Hardware and Software requirements : hardware-16GB RAM, Intel Processor(i9), software: Python (Version 3.x.), Google Colab

System Configration : Operating System: Windows 11, IDE: Google Colab

Theory : Python file handling read, write and manages files like Excel, CSV and binary. It is useful for storing data, creating reports etc

Write a program to create a csv file that we can directly open in MS-Excel.

```
import csv
with open('data.csv', 'w') as file:
    file.write('Name,Age,City\n')
    file.write('Alice,30,New York\n')
    file.write('Bob,25,Los Angeles\n')
    file.write('Charlie,35,Chicago\n')

print("CSV file created!")
```

```
CSV file created!
```

Read the data stored in MS-Excel file and convert it into a dictionary. The record contains rollno, name of student, marks of three subjects. Also calculate total. Display the dictionary data on the monitor.

```
import csv

student={}
with open('student.csv', 'r') as file:
    reader = csv.DictReader(file)
    for row in reader:
        total = int(row['marks1']) + int(row['marks2']) + int(row['marks3'])
        row['total'] = str(total)
        student.append(row)

print(student)
```

```
{}
```

Accept contact details from the user and create a vcard that we can directly store in our mobile.

```
name = input("Enter full name: ")
phone = input("Enter phone number: ")
email = input("Enter email address: ")


vcard = f"""BEGIN:VCARD
VERSION:3.0
FN:{name}
TEL;TYPE=CELL:{phone}
EMAIL:{email}
END:VCARD
"""

with open("contact.vcf", "w") as file:
    file.write(vcard)

print("vCard saved as 'contact.vcf'. You can open this on your mobile to save the contact.")
```

```
Enter full name: VANDITA NAWANI
Enter phone number: 12323445567
Enter email address: gmail.com
vCard saved as 'contact.vcf'. You can open this on your mobile to save the contact.
```

Create a specific subdirectory and copy one file from another subdirectory to this newly created subdirectory.

```
source_path = 'old_folder/file.txt'
target_folder = 'new_folder'
target_path = target_folder + '/file.txt'

with open(source_path, 'r') as f:
    data = f.read()


with open(target_path, 'w') as f:
    f.write(data)

print("File copied!")
```

Write a program to copy contents of one file to another. While doing so, replace all lowercase characters into uppercase characters

```
source_file = 'input.txt'
destination_file = 'output.txt'


with open(source_file, 'r') as f:
    content = f.read()
    content_upper = content.upper()

with open(destination_file, 'w') as f:
    f.write(content_upper)

print("File copied with lowercase converted to uppercase.")
```

Write a program that merges lines alternatively from two files and writes the results to new file. If one file has less number of lines than the other, the remaining lines from the larger file should be simply copied into the target file.

```
file1 = 'file1.txt'
file2 = 'file2.txt'
output_file = 'merged_file.txt'


with open(file1, 'r') as f1, open(file2, 'r') as f2, open(output_file, 'w') as output:

    lines1 = f1.readlines()
    lines2 = f2.readlines()


    max_len = max(len(lines1), len(lines2))


    for i in range(max_len):
        if i < len(lines1):
            output.write(lines1[i])
        if i < len(lines2):
            output.write(lines2[i])

print("Files merged successfully.")
```

If an Employee object contains following details: empcode, empname, Date of Joining, Salary Write a program to serialize and deserialize this data.

```
import pickle
from datetime import datetime


class Employee:
    def __init__(self, empcode, empname, date_of_joining, salary):
        self.empcode = empcode
        self.empname = empname
        self.date_of_joining = date_of_joining
        self.salary = salary
```

```
    def __str__(self):
        return f"Employee Code: {self.empcode}, Name: {self.empname}, Date of Joining: {self.date_of_joining}, Salary: {self.salary}"


emp1 = Employee(101, "John Doe", datetime(2020, 5, 10), 55000.0)


with open("employee.pkl", "wb") as file:
    pickle.dump(emp1, file)

print("Employee object serialized successfully.")

with open("employee.pkl", "rb") as file:
    emp2 = pickle.load(file)

print("Employee object deserialized successfully.")
print(emp2)
```

Given a text file, write a program to create another text file deleting the words ‘a’, ‘the’, ‘an’ and replacing each one of them with a blank space.

```
input_file = 'input.txt'
output_file = 'output.txt'


words_to_remove = ['a', 'the', 'an']

with open(input_file, 'r') as infile, open(output_file, 'w') as outfile:
    content = infile.read()
    for word in words_to_remove:
        content = content.replace(f' {word} ', ' ')
    outfile.write(content)

print("Words removed and file saved.")
```