Experiment -8

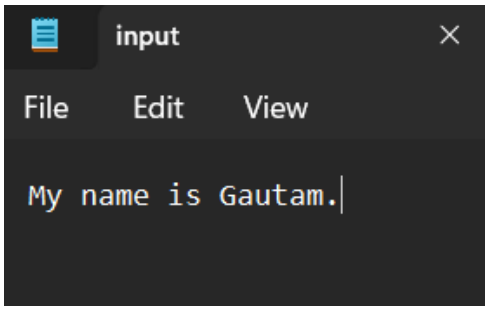
Aim: Create a program that reads data from a file and writes it to another file in a different format.

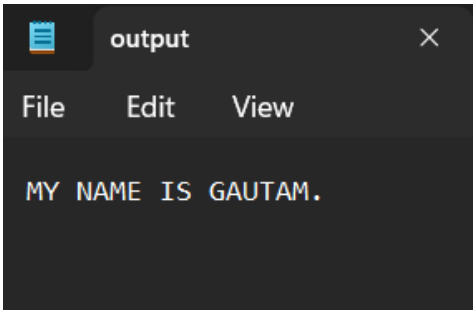
Theory:

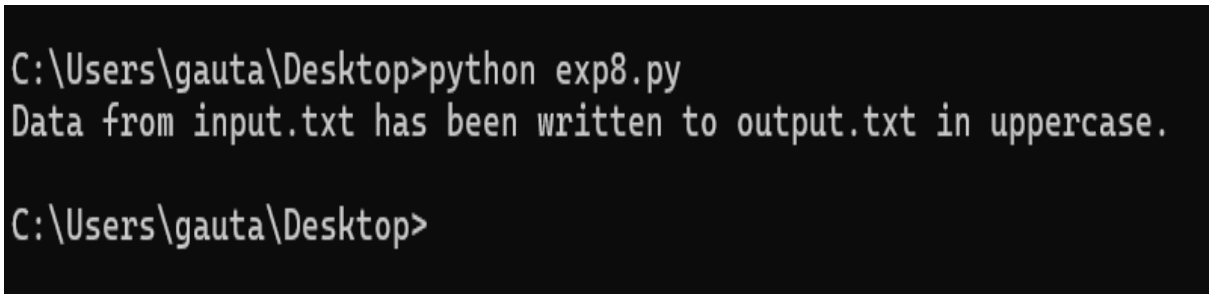
Source Code:

```
input_file_name = "input.txt"
output_file_name = "output.txt"
try:
    with open(input_file_name, 'r') as input_file:
        data = input_file.read()
except FileNotFoundError:
    print(f"Error: {input_file_name} not found.")
    exit()
modified_data = data.upper()
with open(output_file_name, 'w') as output_file:
    output_file.write(modified_data)
print(f"Data from {input_file_name} has been written to {output_file_name} in uppercase.")
```

Output:







Experiment -9

Aim: Create a program that uses regular expressions to find all instances of a specific pattern in a text file.

Theory:

Experiment -10

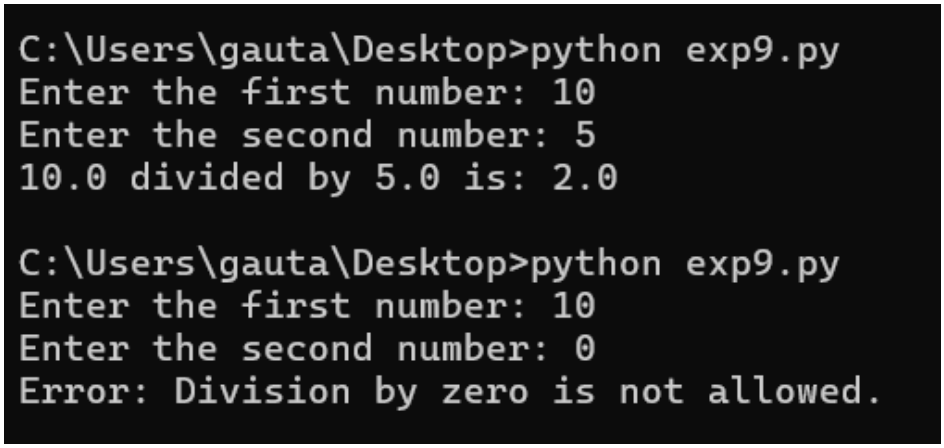
Aim: Create a program that prompts the user for two numbers and then divides them, handling any exceptions that may arise.

Theory:

Source Code:

```
try:
    num1 = float(input("Enter the first number: "))
    num2 = float(input("Enter the second number: "))
    result = num1 / num2
    print(f"{num1} divided by {num2} is: {result}")
except ZeroDivisionError:
    print("Error: Division by zero is not allowed.")
except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

Output:



Experiment -11

Aim: Create a program that uses a graphical user interface (GUI) to allow the user to perform simple calculations.

Theory: