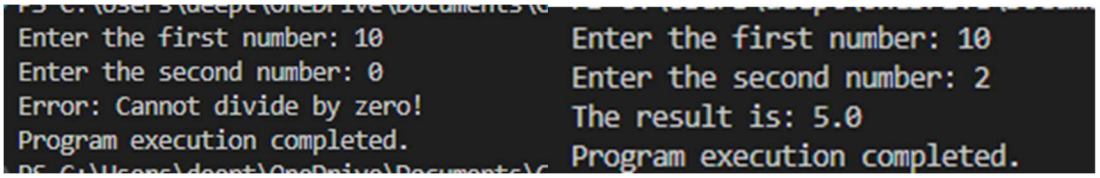
Code-

```
#exp-7
def main():
    try:
        num1 = int(input("Enter the first number: "))
        num2 = int(input("Enter the second number: "))
        result = num1 / num2
        print(f"The result is: {result}")
    except ZeroDivisionError:
        print("Error: Cannot divide by zero!")
    except ValueError:
        print("Error: Invalid input! Please enter numbers only.")
    finally:
        print("Program execution completed.")

if __name__ == "__main__":
    main()
```

OUTPUT-

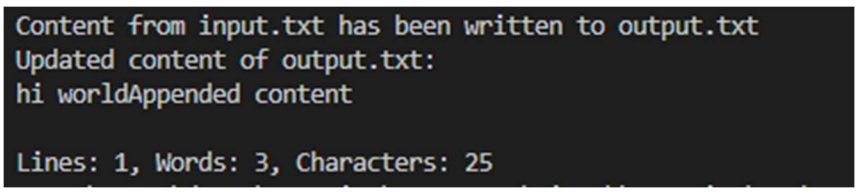


CODE-

```
#exp-8
def read_and_write_file(input_file, output_file):
    try:
        with open(input_file, 'r') as infile:
            content = infile.read()
        with open(output_file, 'w') as outfile:
            outfile.write(content)
        print(f"Content from {input_file} has been written to {output_file}")
    except FileNotFoundError:
        print(f"Error: {input_file} not found!")
    except Exception as e:
        print(f"An error occurred: {e}")

def append_and_display_file(file_name, data_to_append):
    try:
        with open(file_name, 'a') as file:
            file.write(data_to_append + '\n')
        with open(file_name, 'r') as file:
            content = file.read()
            print(f"Updated content of {file_name}:\n{content}")
    except FileNotFoundError:
        print(f"Error: {file_name} not found!")
    except Exception as e:
        print(f"An error occurred: {e}")

def count_file_content(file_name):
    try:
        with open(file_name, 'r') as file:
            lines = file.readlines()
```

```
            num_lines = len(lines)
            num_words = sum(len(line.split()) for line in lines)
            num_chars = sum(len(line) for line in lines)
        print(f"Lines: {num_lines}, Words: {num_words}, Characters: {num_chars}")
    except FileNotFoundError:
        print(f"Error: {file_name} not found!")
    except Exception as e:
        print(f"An error occurred: {e}")


def main():
    input_file = 'input.txt'
    output_file = 'output.txt'
    data_to_append = 'Appended content'
    read_and_write_file(input_file, output_file)
    append_and_display_file(output_file, data_to_append)
    count_file_content(output_file)


if __name__ == "__main__":
    main()
```

OUTPUT-

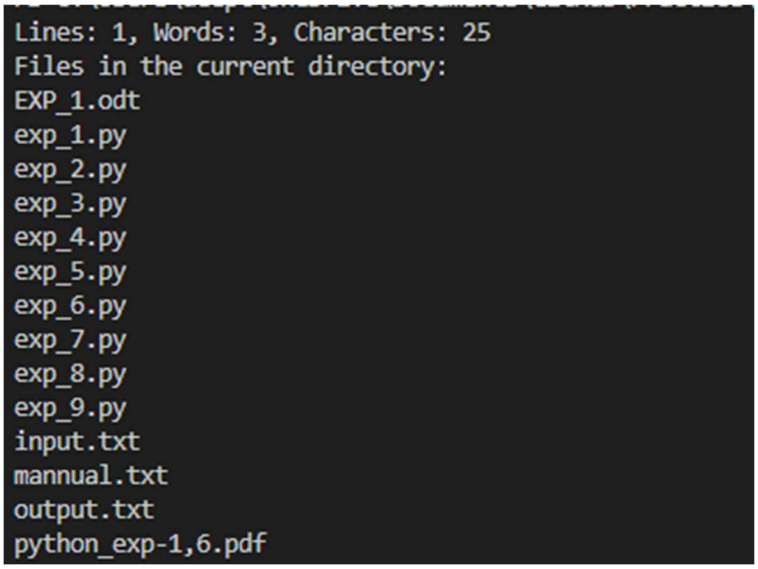


Code-

```
#exp-9
import os
def count_file_content(file_name):
    try:
        with open(file_name, 'r') as file:
            lines = file.readlines()
            num_lines = len(lines)
            num_words = sum(len(line.split()) for line in lines)
            num_chars = sum(len(line) for line in lines)
        print(f"Lines: {num_lines}, Words: {num_words}, Characters: {num_chars}")
    except FileNotFoundError:
        print(f"Error: {file_name} not found!")
    except Exception as e:
        print(f"An error occurred: {e}")

def display_files_in_directory():
    try:
        files = os.listdir('.')
        print("Files in the current directory:")
        for file in files:
            print(file)
    except Exception as e:
        print(f"An error occurred: {e}")
def main():
    file_name = 'output.txt'
    count_file_content(file_name)
    display_files_in_directory()
```

```
if __name__ == "__main__":
    main()
```

OUTPUT-



Code-

```
#exp-10
import sqlite3

# Creating a connection object
conn = sqlite3.connect('student_database.db')
# Creating a cursor object using the connection
cursor = conn.cursor()

# Creating a table in the database
def create_table():
    cursor.execute('''CREATE TABLE IF NOT EXISTS students (
                      id INTEGER PRIMARY KEY AUTOINCREMENT,
                      name TEXT NOT NULL,
                      age INTEGER NOT NULL,
                      grade TEXT NOT NULL)''')
    print("Table created successfully!")

# Inserting values into the table
def insert_values(name, age, grade):
    cursor.execute('''INSERT INTO students (name, age, grade)
                      VALUES (?, ?, ?)''', (name, age, grade))
    conn.commit()
    print("Values inserted successfully!")

# Displaying values from the table
def display_values():
    cursor.execute('SELECT * FROM students')
    rows = cursor.fetchall()
    for row in rows:
```

```
        print(row)

# Updating values in the table
def update_values(student_id, new_grade):
    cursor.execute('''UPDATE students SET grade = ? WHERE id = ?''', (new_grade, student_id))
    conn.commit()
    print("Values updated successfully!")

# Main function
def main():
    create_table()
    insert_values('John Doe', 20, 'A')
    insert_values('Jane Smith', 22, 'B')
    display_values()
    update_values(1, 'A+')
    display_values()

# Calling the main function
if __name__ == "__main__":
    main()
conn.close()
print("Connection closed successfully!")
```

OUTPUT-