Computer Assignment

Kovid Joshi

November 9, 2022

Contents

ACKNOWLEDGMENT

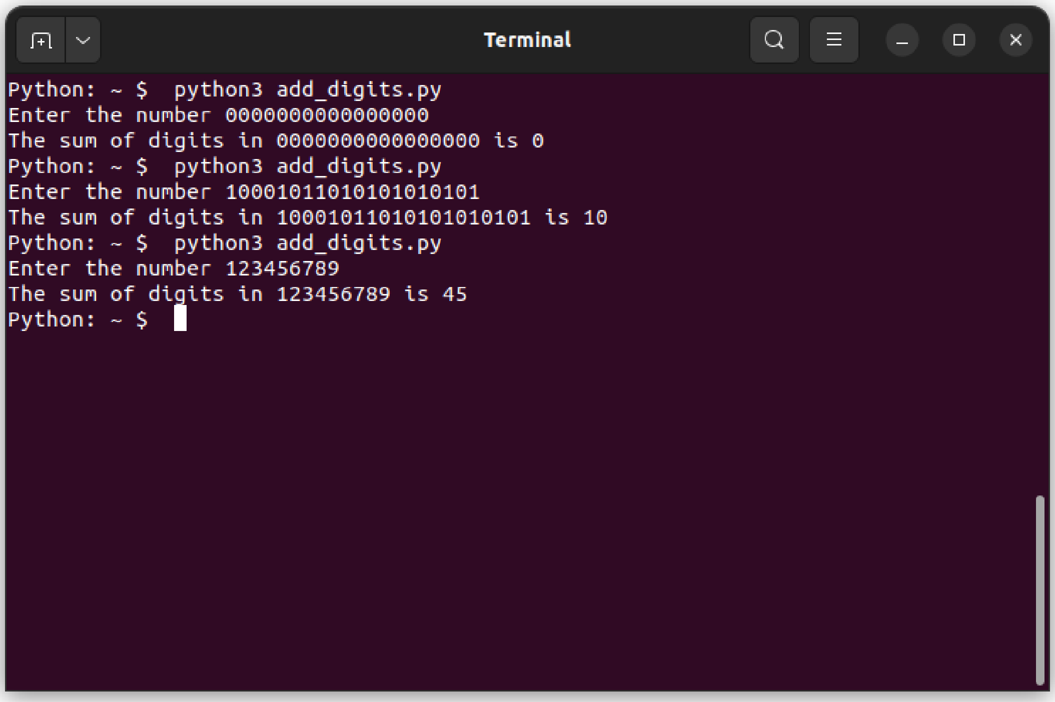
I would like to thank our computer teacher to give us such a opportunity to work and explore on this compilation of python programs using the concepts learned in our computer classes. These program are self made and with pure dedication would like to thank our school authority and the generous staff. This program's contribution is solely mine but a lot of bugs were reported thanks to the people who tested these programs and to point out the bugs, grammar and spelling mistakes.

Adding the digits to itself

```
"""
adding the numbers to the digits of the number itself
"""


def add_digits(number, digit_sum=0):
    """adding digits of a number
    using functions and loops"""
    if number.isnumeric() is False:
        return 'Enter a number not a string!'
    for digits in number:
        digits = int(digits)
        digit_sum += digits
    return f'The sum of digits in {number} is {digit_sum} '


ask_number = input("Enter the number ")
print(add_digits(ask_number))
```

add_digits.py

output



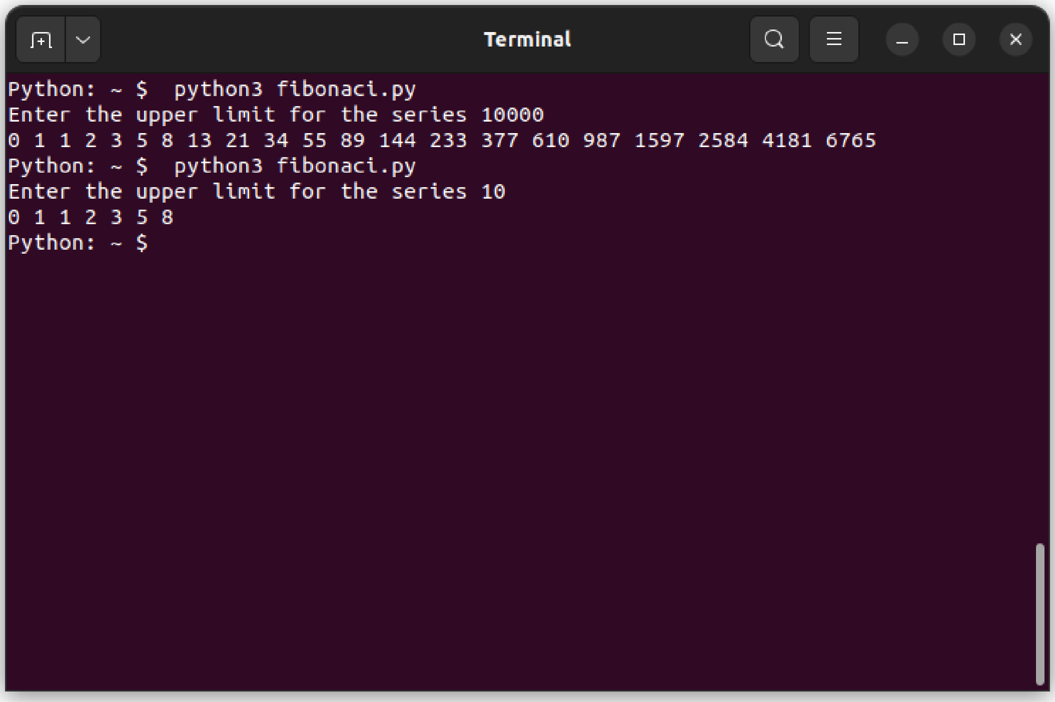
```
Python: ~ $  python3 add_digits.py
Enter the number 0000000000000000
The sum of digits in 0000000000000000 is 0
Python: ~ $  python3 add_digits.py
Enter the number 10001011010101010101
The sum of digits in 10001011010101010101 is 10
Python: ~ $  python3 add_digits.py
Enter the number 123456789
The sum of digits in 123456789 is 45
Python: ~ $
```

Fibonacci Series Program

```
"""
Fibonaci series of number using function till the specified number in
    the series
"""


def fib(number):
    """
    fibonaci series of the till number in the fibonaci series
    """
    a, b = 0, 1
    while a < number:
        print(a, end=' ')
        a, b = b, a+b
    print()


ask_number = input("Enter the upper limit for the series ")
while ask_number.isnumeric() is False:
    ask_number = input("Please enter a number ")

fib(ask_number)
```

fibonaci.py

output



```
Python: ~ $  python3 fibonaci.py
Enter the upper limit for the series 10000
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765
Python: ~ $  python3 fibonaci.py
Enter the upper limit for the series 10
0 1 1 2 3 5 8
Python: ~ $
```

File search text files

```
”””
 program for searching the data  (word) in the file
”””

import os.path  # for checking the existence of the file in path

count = 0        # word counter
found_flag = False  # word found flag
# asking the text file to search in
ask_file = input(”Enter the file to search ”)

# if the file does not exist
if os.path.exists(ask_file) is False:
    # exit the program
    print(”The file does not exists in the directory ”)
    exit()
else:
    # ask the search string
    search_string = input(”Enter the search string ”)

    # open the text file in read mode and search through the file
    with open(ask_file) as search_file:

        for lines in search_file:
            # breaking the file in line by line
            for words in lines.split():
                # breaking the line word by word
                # if found the word start counting
                if search_string == words:
                    found_flag = True
                    count += 1

# if found
if found_flag:
    print(f”Found the word {search_string!r} in the file {ask_file} found”)

# if not found
else:
    print(”Not Found the data in the file ”)
```

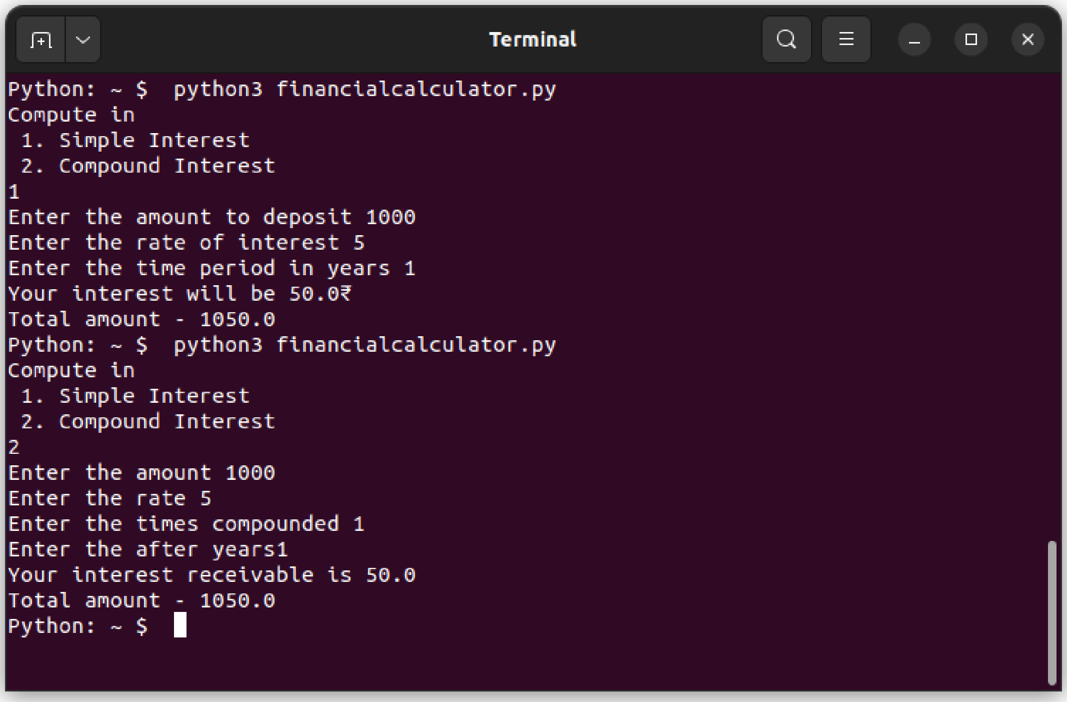
search.py

Financial Interest Calculator

```
"""
interest calculator
"""
amount = 0


def si(principal, rate, time):
    """
    calculate the simple interest
    :param principal: int
    :param rate: int
    :param time: int
    :return: value
    """
    value = (principal*rate*time)/100
    return value


def ci():
    """
    calculate the compound interest
    :return:
    """
    global amount
    amount = int(input("Enter the amount "))
    rate = int(input("Enter the rate "))
    times_compounded = int(input("Enter the times compounded "))
    time = int(input("Enter the after years"))

    value = amount*(1+(rate/100*times_compounded))**(times_compounded*
    time)
    return value


try:

    option = input("Compute in \n 1. Simple Interest\n 2. Compound
    Interest \n").strip().casefold()

    if option in ['1', 'si', 'simple interest']:
        ask_amount = int(input("Enter the amount to deposit "))
        ask_rate = int(input("Enter the rate of interest "))
        ask_time = int(input("Enter the time period in years "))

        print(f"Your interest will be {si(ask_amount, ask_rate,
    ask_time)}")
        print(f'Total amount - {si(ask_amount, ask_rate,ask_time)+
    ask_amount}')

    elif option in ['2', 'compound interest', 'ci']:
        compound_interest = ci()
        print(f"Your interest receivable is {compound_interest - amount
    }")
        print(f"Total amount - {compound_interest}")

except ValueError:
    print("Please enter a number in values given above ")
```

financialcalculator.py

output



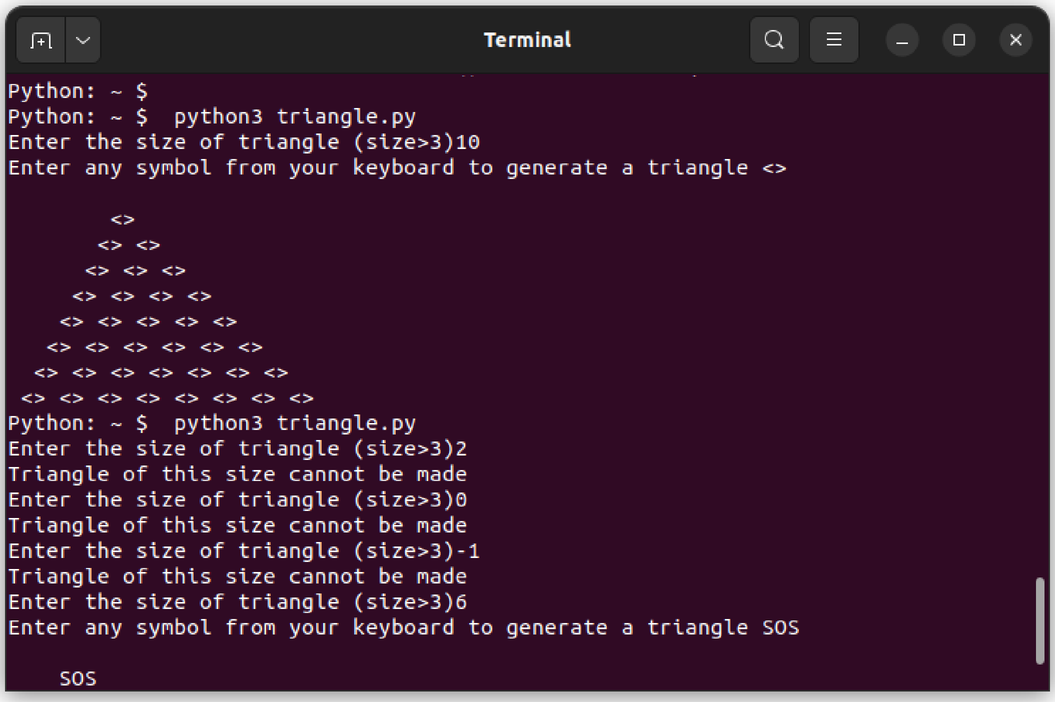
```
Python: ~ $  python3 financialcalculator.py
Compute in
 1. Simple Interest
 2. Compound Interest
1
Enter the amount to deposit 1000
Enter the rate of interest 5
Enter the time period in years 1
Your interest will be 50.0₹
Total amount - 1050.0
Python: ~ $  python3 financialcalculator.py
Compute in
 1. Simple Interest
 2. Compound Interest
2
Enter the amount 1000
Enter the rate 5
Enter the times compounded 1
Enter the after years1
Your interest receivable is 50.0
Total amount - 1050.0
Python: ~ $
```

Triangle maker

```
"""
printing a equilateral triangle
"""
while True:
    triangle_size = int(input("Enter the size of triangle (size>3)"))
    # asking the size of the triangle
    if triangle_size < 4:
        # checking for the size to be not very small
        print("Triangle of this size cannot be made")
        continue
    else:
        # asking for any symbol from which we can make a triangle
        triangle_symbol = input("Enter any symbol from your keyboard to
     generate a triangle ")
        break

# using the loop and given size to drive the loop further
for size in range(1, triangle_size):
    for size2 in range(size, triangle_size):
        # printing the adequate spaces required by the triangle
        print(" ", end="")
    for size3 in range(1, size):
        # printing the symbol provided by the user
        print(f"{triangle_symbol} ", end='')
    # for a new line at the end of a row
    print()
    # run the loop again
```

triangle.py

output



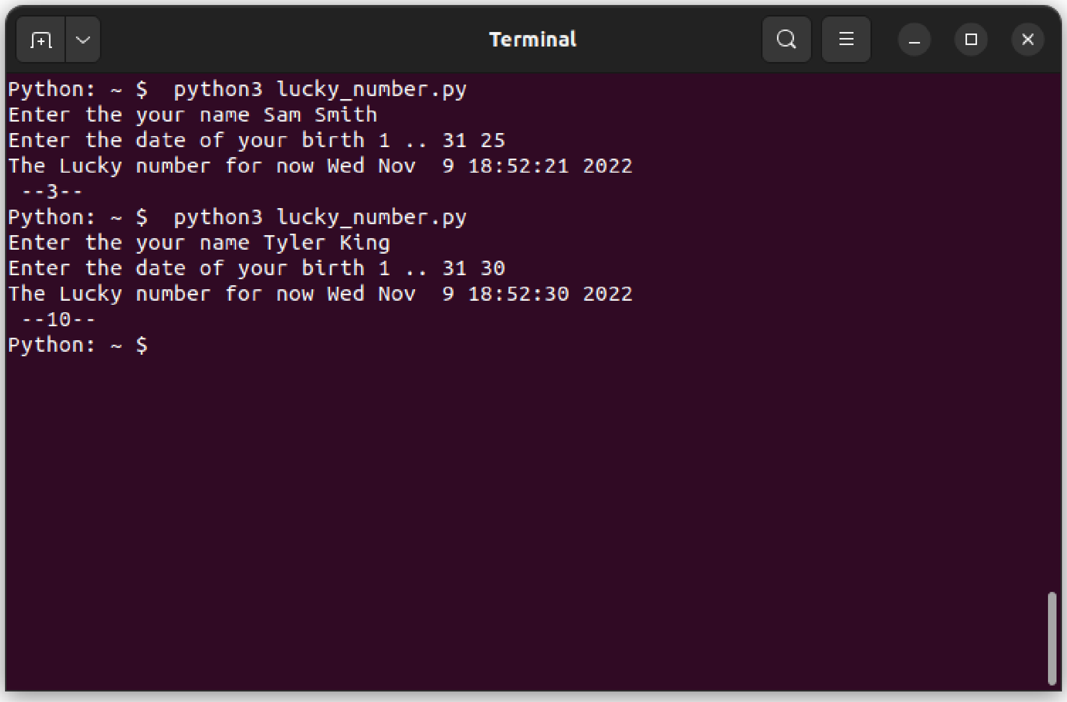
```
Python: ~ $
Python: ~ $  python3 triangle.py
Enter the size of triangle (size>3)10
Enter any symbol from your keyboard to generate a triangle <>

        <>
       <> <>
      <> <> <>
     <> <> <> <>
    <> <> <> <> <>
   <> <> <> <> <> <>
  <> <> <> <> <> <> <>
 <> <> <> <> <> <> <> <>
Python: ~ $  python3 triangle.py
Enter the size of triangle (size>3)2
Triangle of this size cannot be made
Enter the size of triangle (size>3)0
Triangle of this size cannot be made
Enter the size of triangle (size>3)-1
Triangle of this size cannot be made
Enter the size of triangle (size>3)6
Enter any symbol from your keyboard to generate a triangle SOS

    SOS
```

Lucky number calculator

```python
def check():
    """
    Checking for a valid data on
    name and date of birth of a person
    """
    while True:
        ask_name = input("Enter the your name ").title().strip()
        ask_date = input("Enter the date of your birth 1 .. 31 ")
        # checking for the data given to be a number
        if ask_name.isascii() is True and ask_date.isnumeric() is True:
            ask_date = int(ask_date)
            # date must be between 1 .. 31
            if 31 >= ask_date > 0:
                return ask_name, ask_date
            print("Enter a valid date")
        else:
            print("ERROR enter the values again")


def calc():
    """Calculation of the Lucky number of the person"""
    import hashlib
    import time
    time_now = time.localtime()      # get the time
    name, date = check()         # get the checked values returned by funciton
    sup_list = f'{name} {date} {time.monotonic()}'      # make a gobble
    value = hashlib.sha256(sup_list.encode()).hexdigest()    # compute the hash of the value in gobble
    hex_value = int(value, 16)  # convert the hexadecimal value to int
    lucky_number = round(hex_value*1e-76)        # round the value
    return f'The Lucky number for now {time.asctime(time_now)} \n ---{lucky_number}---'
    # print the lucky number

print(calc())
```

lucky_number.py

output


Terminal
```
Python: ~ $  python3 lucky_number.py
Enter the your name Sam Smith
Enter the date of your birth 1 .. 31 25
The Lucky number for now Wed Nov  9 18:52:21 2022
 --3--
Python: ~ $  python3 lucky_number.py
Enter the your name Tyler King
Enter the date of your birth 1 .. 31 30
The Lucky number for now Wed Nov  9 18:52:30 2022
 --10--
Python: ~ $
```

Notemaker program

```
from time import sleep

name = input(”Enter Your Name ”)
true = True
file_name = input(”Enter a New file Name or Load the previous file ”)
while true is True:
    print(f’’’\
    Hello, {name.split()[0].capitalize()}!
            Data Saving Software Version 0.00.02
            What do you want to do
            1. Add data  [1]
            2. delete previous data [2]
            3. Know your data [3]
            4. About the program [4]
            5. How to use the program [5]
            6. Exit [6]’’’)
    mode_opened = input().casefold().strip()
    if mode_opened in [’1’, ’add’]:
        # adding the data to the file opened
        y = ’a’
        x = True
        while x is True:
            with open(file_name + ’.tif’, y, encoding=’utf8’) as mem_file:
                print(”press enter to go to menu or type exit”)
                appending_input = input(”Enter Your data \n\t ”)
                mem_file.write(appending_input + ’\n’)
                if appending_input in [’’, ’exit’, ’quit’]:
                    break

    elif mode_opened == ’2’:
        # deleting the data from the user file
        y = ’w’
        del_choice = input(”Are you sure to delete the data ? ”).strip().casefold()
        if del_choice in [’yes’, ’y’, ’yep’]:
            with open(file_name + ’.tif’, y, encoding=’utf8’) as mem_file:
                print(”processing”)
                sleep(1)
                print(”you deleted your data! ”)
                mem_file.write(’’)

    elif mode_opened == ’3’:
        # display the opened file to the user
        y = ’r’
        with open(file_name + ’.tif’, y, encoding=’utf8’) as mem_file:
            reading = mem_file.read()
            print(f' Here is your data {name.split()[0].capitalize()}!\
    n'
                  f'{reading}')

    elif mode_opened in ['exit', 'quit', '6']:
        # exiting the program
        print("Thanks for visiting!")
        input(" Enter to exit ....")
        exit()
    elif mode_opened in ['help', '5']:
        print(f"""
        data Saving Software version Guide
            data saving program works to stay organised
                    """)
    elif mode_opened in ['4', 'version']:
        print('DATA Saving Software version 0.00.02')
    else:
        print('enter something valid among the above ')
input()
```

notingprogram.py

Displaying the reversed data of a text file

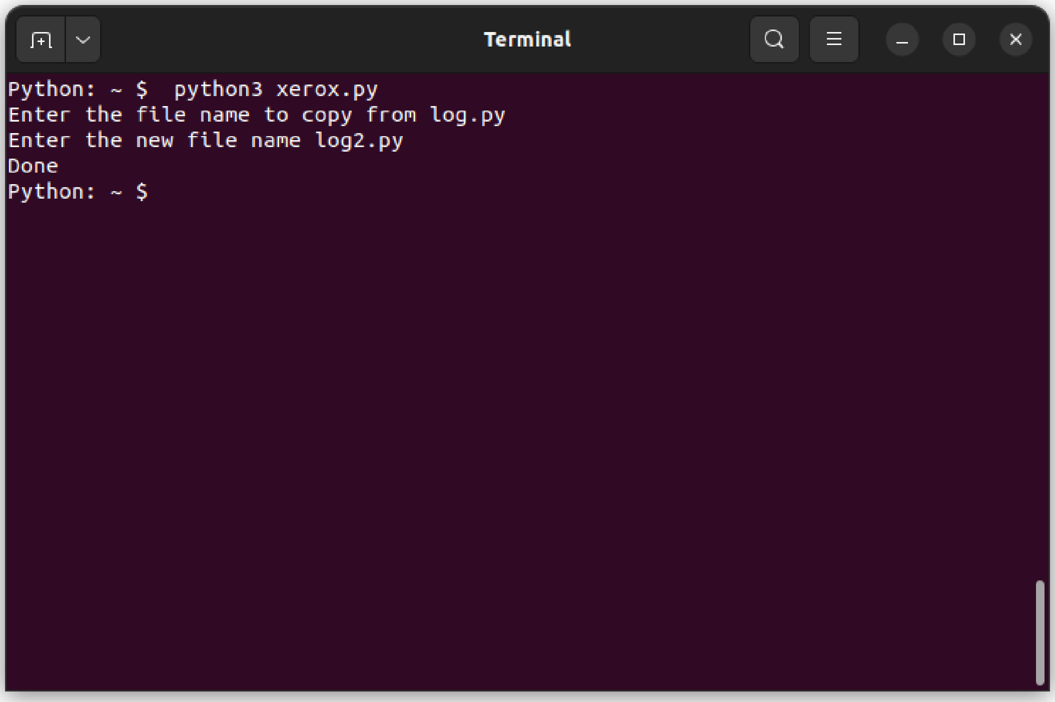
```
"""
# reverse the data of the file
"""

ask_file = input("Enter the file name ")
# asking the file name
reverse_list = []
# storing the file data in a list for taking care of files with lines
try:
    with open(ask_file) as data_file:
        # open the file for reading
        for lines in data_file:
            # read file line by line
            print(lines, end="")
            # reverse the contents of the line and store in the
    reversed list
            reverse_data = lines[::-1]
            # add the reversed data to the file
            reverse_list.append(reverse_data)
    print()
    # new line when the loop ends
    for reversed_lines in reverse_list:
        # print the reverse data of the file
        print(reversed_lines, end='')
    print()
    # new line at the end

except FileNotFoundError:
    print(f"The file {ask_file!r} is not found! ")
# if the file is not found print the following message
```

filedatareverse.py

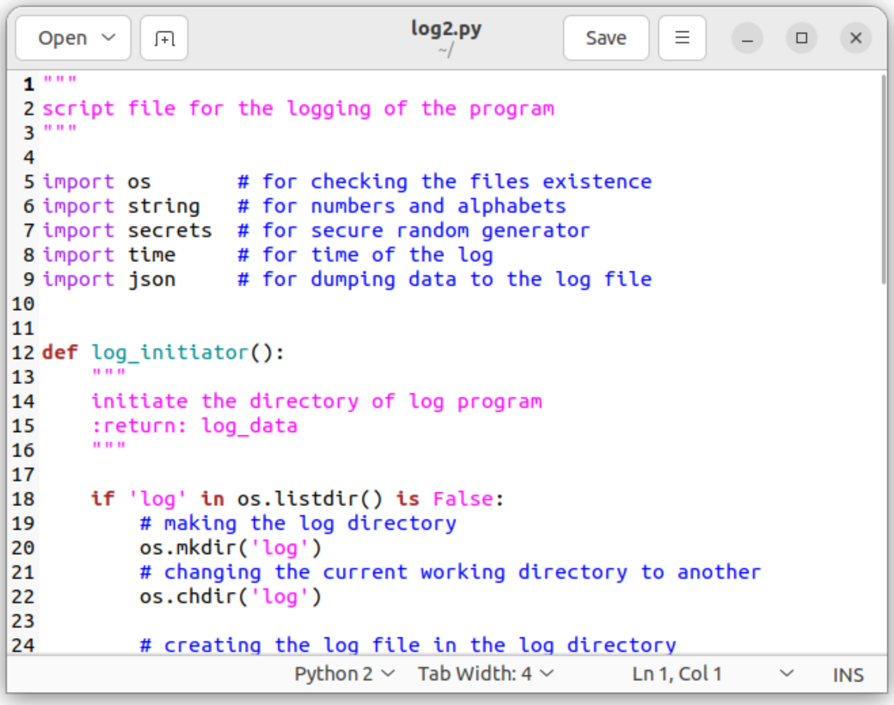
File copying program

```
"""
copy the file data from one to another
"""
import os.path

ask_file = input("Enter the file name to copy from ")
# asking for the file name

if os.path.exists(ask_file) is False:
    # checking for the files existence
    print("The file does not exists ")

# storing the file data using a list
file_data = []
with open(ask_file) as read_file:
    for file_lines in read_file:
        # adding the data to the list line by line
        file_data.append(file_lines)
    # asking the file to save the document which is copied
    ask_new_filename = input("Enter the new file name ")
# checking for the new files name file existence
while os.path.exists(ask_new_filename):
    print(f"The file {ask_new_filename} already exists overwriting may
    corrupt the file\n enter a newfile name")
    # if new file exists is True ask another name for the file
    ask_new_filename = input("Enter the new file name ")

# write to the new file the copied data
with open(ask_new_filename, 'w') as new_file:
    for new_lines in file_data:
        new_file.write(new_lines)

# print the conformation message
print("Done")
```

xerox.py



```
Python: ~ $  python3 xerox.py
Enter the file name to copy from log.py
Enter the new file name log2.py
Done
Python: ~ $
```



```
"""
script file for the logging of the program
"""

import os       # for checking the files existence
import string   # for numbers and alphabets
import secrets  # for secure random generator
import time     # for time of the log
import json     # for dumping data to the log file


def log_initiator():
    """
    initiate the directory of log program
    :return: log_data
    """

    if 'log' in os.listdir() is False:
        # making the log directory
        os.mkdir('log')
        # changing the current working directory to another
        os.chdir('log')

        # creating the log file in the log directory
```

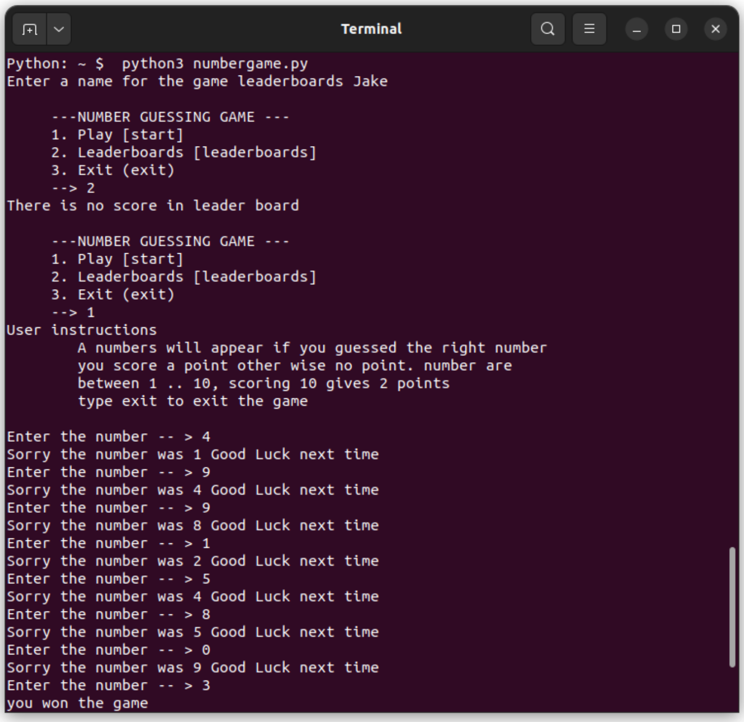
Random Number game

```
"""
# number guessing game
"""

import random   # required for random numbers
import csv      # for saving leader boards
import os       # for checking the existence of file

# asking a name for the leader boards
name = input("Enter a name for the game leaderboards ")

# initial score and count set to zero
score, count = 0, 0


def score_add(user_name, user_score, user_win_ratio):
    """
    for adding the score to a file as the leader board
    on exiting the program
    """
    # check for the file's existence
    if os.path.exists('score.csv') is False:
        with open("score.csv", 'x') as new_file:
            writer = csv.writer(new_file)
            # writing the initial header for the leader board
            writer.writerow(['name ', 'score', 'win ratio'])

    # writing the data given in the arguments of the function
    with open('score.csv', 'a') as add_score:
        write = csv.writer(add_score)
        write.writerow([user_name, user_score, user_win_ratio])


def leader_board():
    """
    show the leader board of the game
    :return:
    """
    # if no score file in folder
    if os.path.exists('score.csv') is False:
        print("There is no score in leader board")
    else:
        # printing the score file's data in leader boards
        with open("score.csv") as file:
            reader = csv.reader(file, dialect='excel', delimiter=',')
            for leader_score in reader:
                print(f'{leader_score[0]} -- {leader_score[1]} wins, --
    with win ratio-- {leader_score[2]}')
```

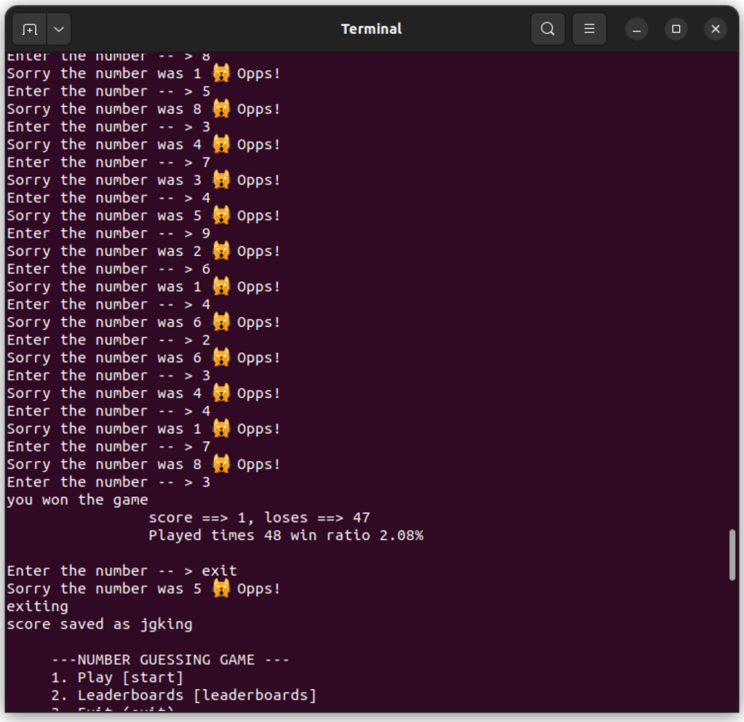
```
# main program execution
while True:
    ask_options = input("""
     ---NUMBER GUESSING GAME ---
     1. Play [start]
     2. Leaderboards [leaderboards]
     3. Exit (exit)
     --> """)    # ask options from the user

    if ask_options in ['play', 'start', '1']:
        print("""User instructions
        A numbers will appear if you guessed the right number
        you score a point other wise no point. number are
        between 1 .. 10, scoring 10 gives 2 points
        type exit to exit the game \n""")
        # game ^ instructions

        while True:
            ask_guess = input("Enter the number --> ")  # give the number
            number = random.randint(1, 10)
            if ask_guess == str(number):    # check the number from the random number
                if number == 10:
                    score += 2
                    count += 1
                else:
                    score += 1
                    count += 1
                print(f"""you won the game {'great' if score > 6 else ''}
                score ==> {score}, loses ==> {count - score}
                Played times {count} win ratio {round((score/count) * 100, 2)}%\n""")

            else:
                count += 1
                print(f"Sorry the number was {number} {' Opps! ' if count - score > 10 else 'Good Luck next time'}")

            if ask_guess in ['exit', 'quit', 'n']:
                print(f'exiting\nscore saved as {name}')
                score_add(name, score, f'{round(score/count * 100, 2)}%')
                break
            else:
                continue

    elif ask_options in ['leaderboards', 'leaderboard', '2', 'score card']:
        # checking the leaderboard
        leader_board()

    elif ask_options in ['exit', 'quit']:
        # exiting the program
        exit()
    else:
        # informing for an unknown object
        print("Sorry that is not valid enter from the menu ")
```

numbergame.py

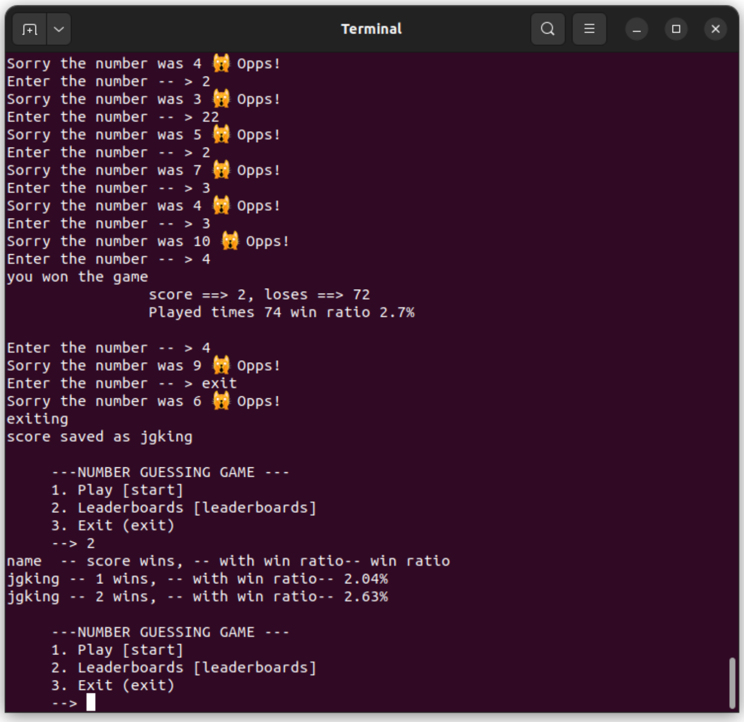
output



```
Python: ~ $  python3 numbergame.py
Enter a name for the game leaderboards Jake

     ---NUMBER GUESSING GAME ---
     1. Play [start]
     2. Leaderboards [leaderboards]
     3. Exit (exit)
     --> 2
There is no score in leader board

     ---NUMBER GUESSING GAME ---
     1. Play [start]
     2. Leaderboards [leaderboards]
     3. Exit (exit)
     --> 1
User instructions
        A numbers will appear if you guessed the right number
        you score a point other wise no point. number are
        between 1 .. 10, scoring 10 gives 2 points
        type exit to exit the game

Enter the number -- > 4
Sorry the number was 1 Good Luck next time
Enter the number -- > 9
Sorry the number was 4 Good Luck next time
Enter the number -- > 9
Sorry the number was 8 Good Luck next time
Enter the number -- > 1
Sorry the number was 2 Good Luck next time
Enter the number -- > 5
Sorry the number was 4 Good Luck next time
Enter the number -- > 8
Sorry the number was 5 Good Luck next time
Enter the number -- > 0
Sorry the number was 9 Good Luck next time
Enter the number -- > 3
you won the game
```



```
Enter the number -- > 8
Sorry the number was 1 🙀 Opps!
Enter the number -- > 5
Sorry the number was 8 🙀 Opps!
Enter the number -- > 3
Sorry the number was 4 🙀 Opps!
Enter the number -- > 7
Sorry the number was 3 🙀 Opps!
Enter the number -- > 4
Sorry the number was 5 🙀 Opps!
Enter the number -- > 9
Sorry the number was 2 🙀 Opps!
Enter the number -- > 6
Sorry the number was 1 🙀 Opps!
Enter the number -- > 4
Sorry the number was 6 🙀 Opps!
Enter the number -- > 2
Sorry the number was 6 🙀 Opps!
Enter the number -- > 3
Sorry the number was 4 🙀 Opps!
Enter the number -- > 4
Sorry the number was 1 🙀 Opps!
Enter the number -- > 7
Sorry the number was 8 🙀 Opps!
Enter the number -- > 3
you won the game
                score ==> 1, loses ==> 47
                Played times 48 win ratio 2.08%

Enter the number -- > exit
Sorry the number was 5 🙀 Opps!
exiting
score saved as jgking

     ---NUMBER GUESSING GAME ---
     1. Play [start]
     2. Leaderboards [leaderboards]
```



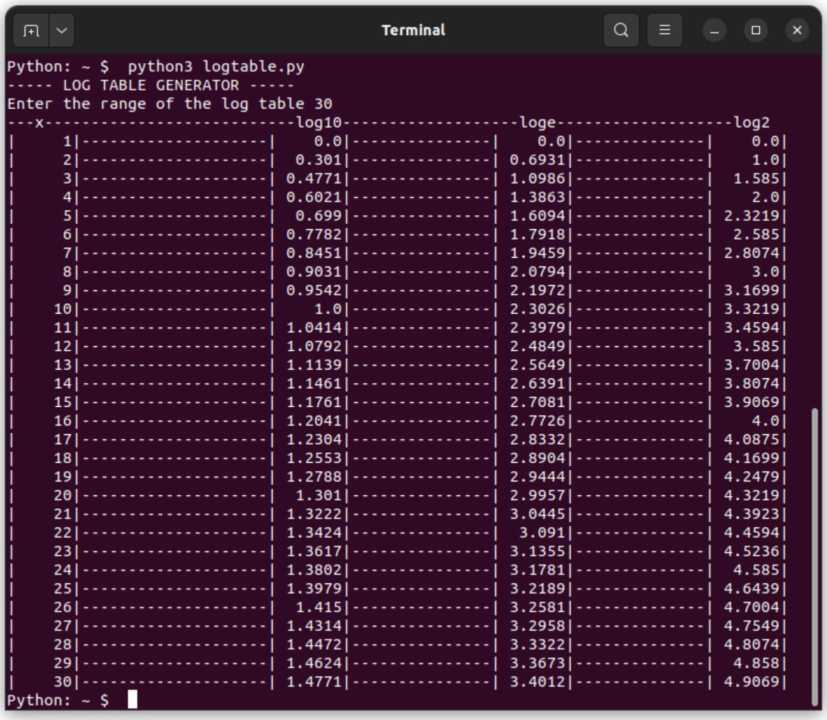
```
Sorry the number was 4 🙀 Opps!
Enter the number -- > 2
Sorry the number was 3 🙀 Opps!
Enter the number -- > 22
Sorry the number was 5 🙀 Opps!
Enter the number -- > 2
Sorry the number was 7 🙀 Opps!
Enter the number -- > 3
Sorry the number was 4 🙀 Opps!
Enter the number -- > 3
Sorry the number was 10 🙀 Opps!
Enter the number -- > 4
you won the game
                score ==> 2, loses ==> 72
                Played times 74 win ratio 2.7%

Enter the number -- > 4
Sorry the number was 9 🙀 Opps!
Enter the number -- > exit
Sorry the number was 6 🙀 Opps!
exiting
score saved as jgking

     ---NUMBER GUESSING GAME ---
     1. Play [start]
     2. Leaderboards [leaderboards]
     3. Exit (exit)
     --> 2
name  -- score wins, -- with win ratio-- win ratio
jgking -- 1 wins, -- with win ratio-- 2.04%
jgking -- 2 wins, -- with win ratio-- 2.63%

     ---NUMBER GUESSING GAME ---
     1. Play [start]
     2. Leaderboards [leaderboards]
     3. Exit (exit)
     -->
```

Mathematical Log table generator

```
"""
# log table making program till desired number
"""

import math
print("——— LOG TABLE GENERATOR ———")

ask_range = input("Enter the range of the log table ")

while ask_range.isnumeric() is False:
    # if the range is not a number
    ask_range = input("Enter the range of the log table ")

# print the header
print("——x———————————————————————log10———————————————————loge———————————————————log2")
for number in range(1, int(ask_range) + 1):
    # calculating the log of numbers
    log10 = round(math.log(number, 10), 4)       # base 10
    loge = round(math.log(number, math.e), 4)    # base e natural lo
    log2 = round(math.log(number, 2), 4)         # base 2
    # print the result row wise
    print(f"|{number:6}|———————————————————|{log10:7}|——————————————————|{loge:7}|———————————————————|{log2:7}|")
```

logtable.py

output



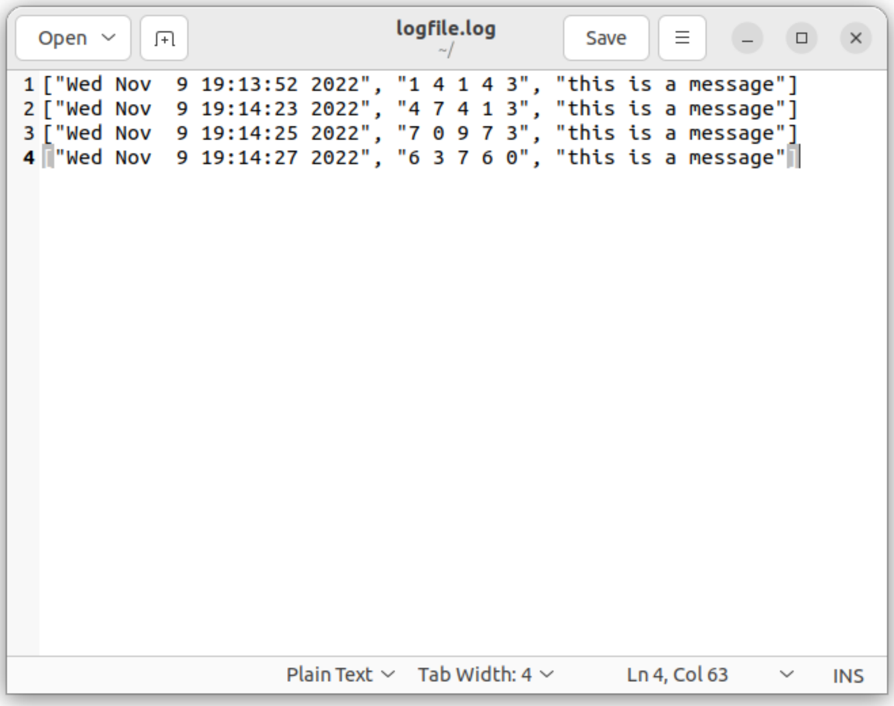
```
Python: ~ $  python3 logtable.py
----- LOG TABLE GENERATOR -----
Enter the range of the log table 30
---x---------------------------log10-------------------loge-------------------log2
|     1|--------------------|    0.0|---------------|    0.0|--------------|    0.0|
|     2|--------------------|  0.301|---------------| 0.6931|--------------|    1.0|
|     3|--------------------| 0.4771|---------------| 1.0986|--------------|  1.585|
|     4|--------------------| 0.6021|---------------| 1.3863|--------------|    2.0|
|     5|--------------------|  0.699|---------------| 1.6094|--------------| 2.3219|
|     6|--------------------| 0.7782|---------------| 1.7918|--------------|  2.585|
|     7|--------------------| 0.8451|---------------| 1.9459|--------------| 2.8074|
|     8|--------------------| 0.9031|---------------| 2.0794|--------------|    3.0|
|     9|--------------------| 0.9542|---------------| 2.1972|--------------| 3.1699|
|    10|--------------------|    1.0|---------------| 2.3026|--------------| 3.3219|
|    11|--------------------| 1.0414|---------------| 2.3979|--------------| 3.4594|
|    12|--------------------| 1.0792|---------------| 2.4849|--------------|  3.585|
|    13|--------------------| 1.1139|---------------| 2.5649|--------------| 3.7004|
|    14|--------------------| 1.1461|---------------| 2.6391|--------------| 3.8074|
|    15|--------------------| 1.1761|---------------| 2.7081|--------------| 3.9069|
|    16|--------------------| 1.2041|---------------| 2.7726|--------------|    4.0|
|    17|--------------------| 1.2304|---------------| 2.8332|--------------| 4.0875|
|    18|--------------------| 1.2553|---------------| 2.8904|--------------| 4.1699|
|    19|--------------------| 1.2788|---------------| 2.9444|--------------| 4.2479|
|    20|--------------------|  1.301|---------------| 2.9957|--------------| 4.3219|
|    21|--------------------| 1.3222|---------------| 3.0445|--------------| 4.3923|
|    22|--------------------| 1.3424|---------------|  3.091|--------------| 4.4594|
|    23|--------------------| 1.3617|---------------| 3.1355|--------------| 4.5236|
|    24|--------------------| 1.3802|---------------| 3.1781|--------------|  4.585|
|    25|--------------------| 1.3979|---------------| 3.2189|--------------| 4.6439|
|    26|--------------------|  1.415|---------------| 3.2581|--------------| 4.7004|
|    27|--------------------| 1.4314|---------------| 3.2958|--------------| 4.7549|
|    28|--------------------| 1.4472|---------------| 3.3322|--------------| 4.8074|
|    29|--------------------| 1.4624|---------------| 3.3673|--------------|  4.858|
|    30|--------------------| 1.4771|---------------| 3.4012|--------------| 4.9069|
Python: ~ $ 
```

System Log Generation Program

```python
"""
script file for the logging of the program
"""

import os       # for checking the files existence
import string   # for numbers and alphabets
import secrets  # for secure random generator
import time     # for time of the log
import json     # for dumping data to the log file


def log_initiator():
    """
    initiate the directory of log program
    :return: log_data
    """

    if 'log' in os.listdir() is False:
        # making the log directory
        os.mkdir('log')
        # changing the current working directory to another
        os.chdir('log')

        # creating the log file in the log directory
        with open('logfile.log', 'x') as _:
            pass

        os.chdir('..')


def logit(message=''):
    """
    Log the in a particular time and store the message provided
    default message ''
    """

    # checking the existence of the log file
    while os.path.exists('logfile.log') is False:
        print("The Log directory does not exists")
        print(os.getcwd())
        ask_create = input("Do you want to create 'log' ")
        # asking for creation
        if ask_create in ['yes', 'y', 'yep']:
            os.mkdir('log')
            os.chdir('log')

    number_id = ' '.join(secrets.choice(string.digits) for _ in range(5))
    log_data = [time.asctime(time.localtime()), number_id, message]
    # store the log data in the log file
    with open('logfile.log', 'a') as log_file:
        json.dump(log_data, log_file)
        log_file.write('\n')

    # change the directory back the initial
    os.chdir('..')
    # also print the current working directory
    print(os.getcwd())

    # return back a unique random number
    return number_id
```

log.py

output



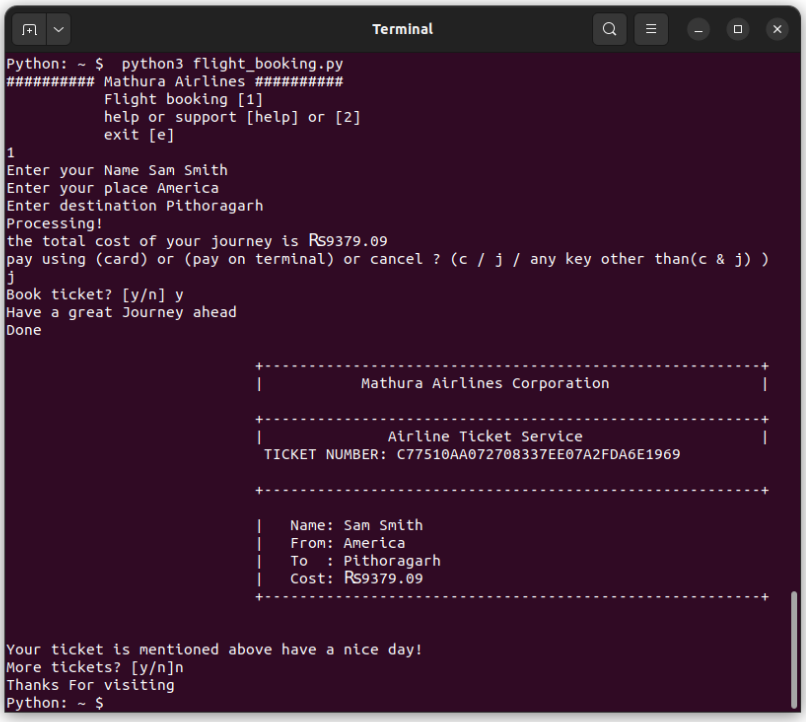
```
["Wed Nov  9 19:13:52 2022", "1 4 1 4 3", "this is a message"]
["Wed Nov  9 19:14:23 2022", "4 7 4 1 3", "this is a message"]
["Wed Nov  9 19:14:25 2022", "7 0 9 7 3", "this is a message"]
["Wed Nov  9 19:14:27 2022", "6 3 7 6 0", "this is a message"]
```

Flight Booking Program

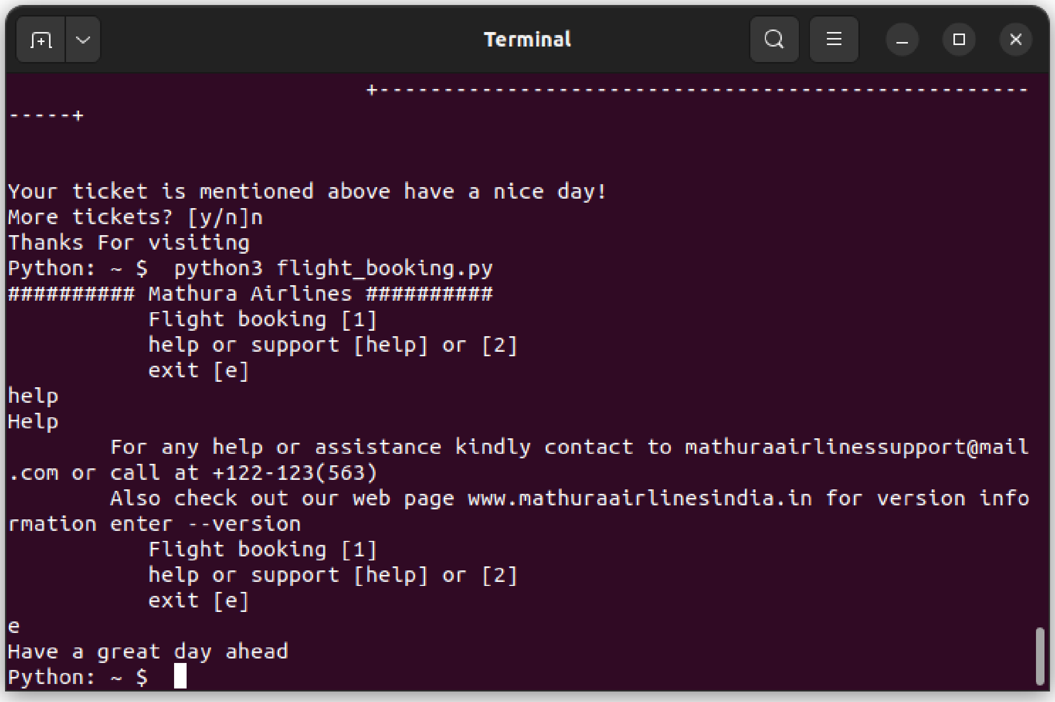
```
”””
# flight booking program
”””

from time import sleep  # for stopping the execution for some time
from random import choice  # for choosing random number for sleep command
from hashlib import md5     # for generating the ticket id

# ————————MISCELLANEOUS SOURCE ITEMS——————————————
sleep_time = [1.3, 2.4, 3, 0.9]
sleep_time2 = [1.3, 2.4, 3, 0.9]
time = choice(sleep_time)
check = 0
# ——————————————————————————————————————————————
print(f”{’#’*10} Mathura Airlines {’#’*10}”)     # ———————HEADING—————————
while True:
    print(f”””\
           Flight booking [1]
           help or support [help] or [2]
           exit [e]”””)
    ask_user = input().lower().strip()

    # ———————————————————ASKING USER DETAILS FOR BOOKING ONE TICKET AT A TIME————————————————
    if ask_user == ’1’:
        ask_name = input(”Enter your Name ”).strip().title()
        ask_from = input(”Enter your place ”).title().strip()
        ask_to = input(”Enter destination ”).title().strip()
        if len(ask_name) == 0 or ask_from.isspace() or ask_to.isspace() or ask_from == ’ ’ or ask_to == ’ ’:
            print(”No spaces, please enter them again ”)  # —————————— checking for spaces and empty part
            continue
        print(’Processing! ’)
        sleep(time)       # $$$$$$$$$$$$$$$$$$$$$$$$sleep$$$$$$$$$$$$$$$$$$$$$$
        # ——————————————CALCULATION OF THE COST————————————————————
        cost1 = hash(ask_from)
        cost2 = hash(ask_to)
        cost = (round(cost1) - round(cost2))
        temp_total = abs(cost)
        value = temp_total/10e14
        total = round(value, 2)
        # ——————————————————————————————————————————————————————————————
    print(f"the total cost of your journey is  {total} ")
    ask_payment = input("pay using (card) or (pay on terminal) or
cancel ? (c / j / any key other than(c & j) )"
                        " \n").casefold()
    if ask_payment == 'c' or ask_payment == 'card':  # PAY USING
CARD NUMBER ————————————————
        ask_detail = input("Enter your number ")        # asking for
 user's number
        print("Processing ")
        sleep(choice(sleep_time2))  #
$$$$$$$$$$$$$$$$$$$$$$$$$$$$sleep$$$$$$$$$$$$$$$$$$$$$
        total_money_conformation = input(f"amount {total} will be
credited from {ask_detail} proceed"
                                         f" or cancel? [y/c]").
casefold()
        if total_money_conformation == 'y':  # conform the payment
            sleep(time)      #
$$$$$$$$$$$$$$$$$$$$$$$$$$$sleep$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
            print("Transaction Successful")
        elif total_money_conformation == 'c':  # cancel the payment
            print("Process aborted ")
            print("Transaction UnSuccessful ")
            continue
        else:
            print("Please enter a valid value ")
    elif ask_payment == 'j':  # PAY IN TERMINAL
————————————————————————

        while True:
            # ———————————————— Pay on Terminal conformation
————————

            ask_final = input("Book ticket? [y/n] ").casefold()
            if ask_final == 'y':    #
                print("Have a great Journey ahead ")
                break
            elif ask_final == 'n':
                print("\n Booking Canceled ")   # Cancelling the
booking
                check += 1
                break
            else:
                print("Enter a proper value")
                continue
        if check > 0:
            continue
    else:
        print("Enter a acceptable payment option \n payment
canceled ")
        continue
# ———————————————Final Ticket Number making and printing the
ticket——————————————————
    hash_addition = [ask_name, ask_from, ask_to, time]
    ticket_code = md5(str(hash_addition).encode()).hexdigest().
capitalize()
    print('Done\n')
    print(f"""\

+————————————————————————————————————————+
                        |            Mathura Airlines Corporation
          |

+————————————————————————————————————————+
                    |                Airline Ticket Service
        |
                    TICKET NUMBER: {ticket_code.upper()}

+--------------------------------------------------------------+
                    |   Name: {ask_name}
                    |   From: {ask_from}
                    |   To  : {ask_to}
                    |   Cost:   {total}

+--------------------------------------------------------------+\n\n""")
    print("Your ticket is mentioned above have a nice day! ")
    ask = input("More tickets? [y/n]")
#
------------------------------------------------------------------------------------------------

    if ask.lower() == 'y':          # asking for more tickets else
breaking
        continue
    else:
        print("Thanks For visiting ")
        break
# ------------------------------HELP LEVEL
--------------------------------------
elif ask_user == '2' or ask_user == 'help':
    print("""Help
    For any help or assistance kindly contact to
mathuraairlinessupport@mail.com or call at +122-123(563)
    Also check out our web page www.mathuraairlinesindia.in for
version information enter --version """)
# ----------------------------EXITING THE APPLICATION
-------------------------
elif ask_user == 'e' or ask_user == 'exit':
    print("Have a great day ahead")
    exit()
# ----------------------------VERSION INFORMATION
------------------------------------
elif ask_user == '--version':
    print("Version 0.00.0.2 (previous release ->first version
0.00.01) \n What's new in 0.00.0.2"
          "Flight program by Kovid Joshi"
          "\n 1. Minor bug fixes "
          "\n 2. Source code improvement ")
# ----------------------------in case ask_user is not something
meaningful!--------------------
else:
    print("Enter a proper value ")
```

flight_booking.py

output



```
Python: ~ $  python3 flight_booking.py
########## Mathura Airlines ##########
           Flight booking [1]
           help or support [help] or [2]
           exit [e]
1
Enter your Name Sam Smith
Enter your place America
Enter destination Pithoragarh
Processing!
the total cost of your journey is ₨9379.09
pay using (card) or (pay on terminal) or cancel ? (c / j / any key other than(c & j) )
j
Book ticket? [y/n] y
Have a great Journey ahead
Done

                            +---------------------------------------------------------+
                            |           Mathura Airlines Corporation                  |

                            +---------------------------------------------------------+
                            |              Airline Ticket Service                     |
                             TICKET NUMBER: C77510AA072708337EE07A2FDA6E1969

                            +---------------------------------------------------------+

                            |   Name: Sam Smith
                            |   From: America
                            |   To  : Pithoragarh
                            |   Cost: ₨9379.09
                            +---------------------------------------------------------+


Your ticket is mentioned above have a nice day!
More tickets? [y/n]n
Thanks For visiting
Python: ~ $
```



```
                            +-----------------------------------------------------
-----+


Your ticket is mentioned above have a nice day!
More tickets? [y/n]n
Thanks For visiting
Python: ~ $  python3 flight_booking.py
########## Mathura Airlines ##########
           Flight booking [1]
           help or support [help] or [2]
           exit [e]
help
Help
        For any help or assistance kindly contact to mathuraairlinessupport@mail
.com or call at +122-123(563)
        Also check out our web page www.mathuraairlinesindia.in for version info
rmation enter --version
           Flight booking [1]
           help or support [help] or [2]
           exit [e]
e
Have a great day ahead
Python: ~ $
```

File Specification Program

```python
"""
# program for the file's data's specification
# number of numbers, symbols, letters and spaces
"""
# initial count of numbers are 0
numbers, alphabets, symbols, spaces = 0, 0, 0, 0

# asking the filename for checking the specifications
ask_file = input("Enter the file name ")
try:
    # opening the file for counting the specification
    with open(ask_file) as data_file:
        for lines in data_file:
            for letters in lines:
                # if a letter
                if letters.isalpha():
                    alphabets += 1
                # if a number
                elif letters.isnumeric():
                    numbers += 1
                # if a space
                elif letters.isspace():
                    spaces += 1
                # if a symbol
                else:
                    symbols += 1
    # printing the specification of the file
    print(f"""\
    The Following file {ask_file} has
        Numbers : {numbers}
        Alphabets : {alphabets}
        Symbols : {symbols}
        Spaces : {spaces}""")

except FileNotFoundError:
    print("There is no such file in the directory ")
    # if the file does not exist then print the following message
```

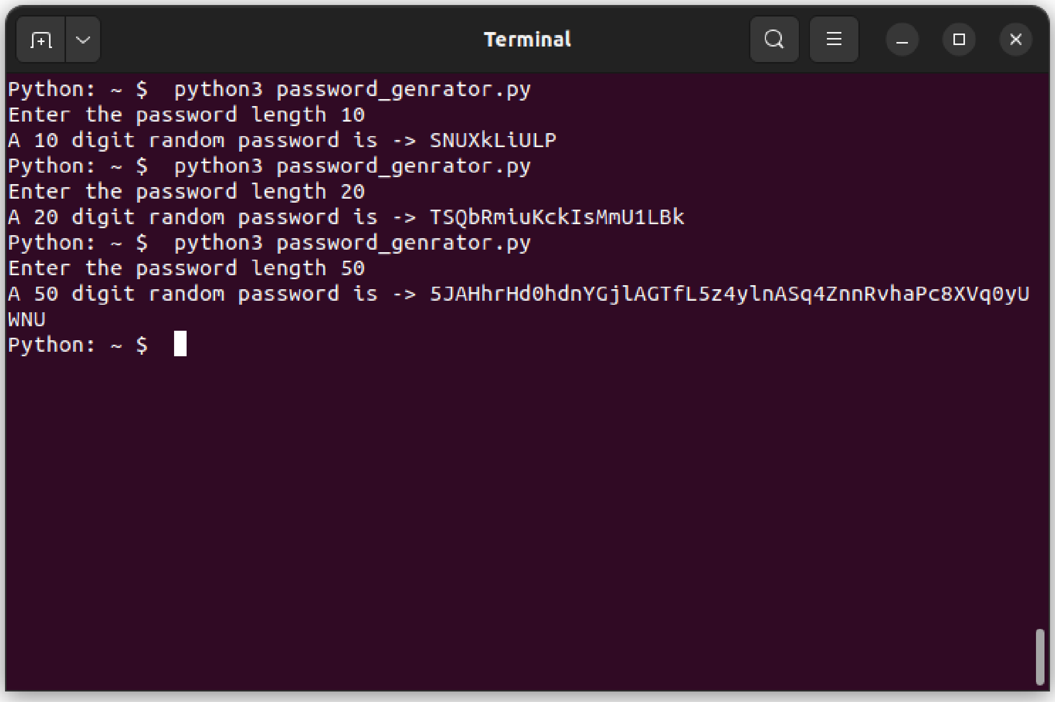
filespecs.py

Random Password generator

```python
"""
 random password generator
"""

import secrets  # secure random digits
import string  # words and letters

ask_pass_length = input("Enter the password length ")
# asking the password length

# checking for the password length to be a number
while ask_pass_length.isnumeric() is False:
    # if not a number ask again
    print("Please enter a number ")
    ask_pass_length = input("Enter the password length ")

# type cast the password length to a
ask_pass_length = int(ask_pass_length)
# generate the password using the join method, secrets' choice function and string letters and digits
# then by running till the for loop is complete
password = ''.join(secrets.choice(string.ascii_letters+string.digits) for lines in range(ask_pass_length))

# printing the password
print(f"A {ask_pass_length} digit random password is -> {password}")
```

password_genrator.py

output



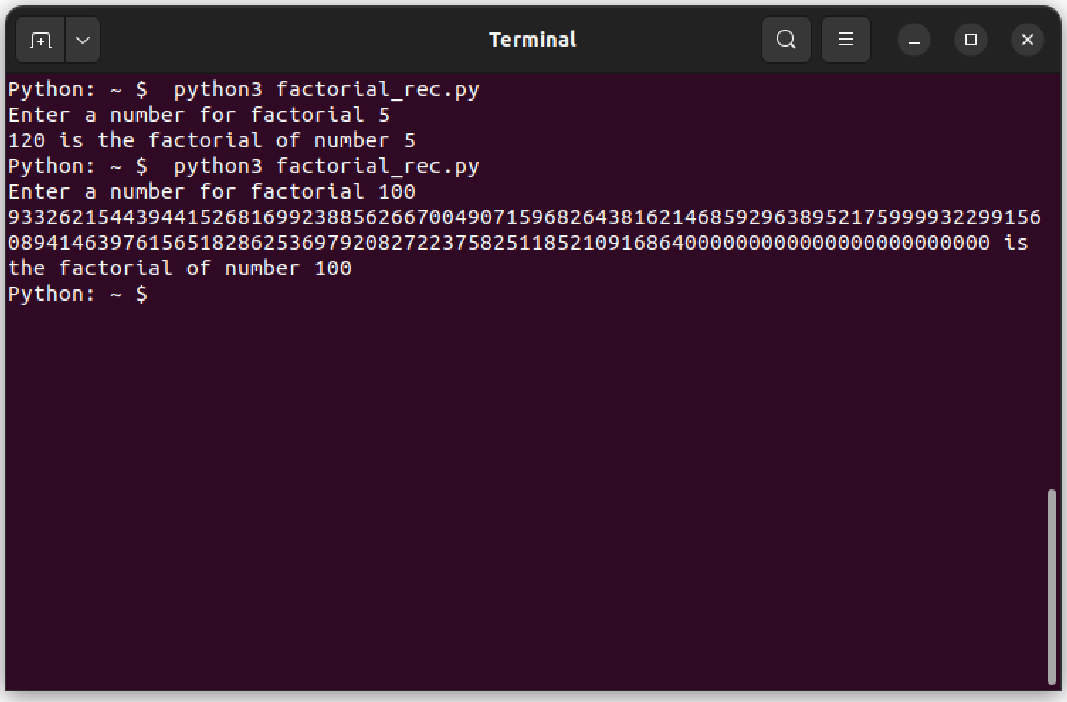
```
Python: ~ $  python3 password_genrator.py
Enter the password length 10
A 10 digit random password is -> SNUXkLiULP
Python: ~ $  python3 password_genrator.py
Enter the password length 20
A 20 digit random password is -> TSQbRmiuKckIsMmU1LBk
Python: ~ $  python3 password_genrator.py
Enter the password length 50
A 50 digit random password is -> 5JAHhrHd0hdnYGjlAGTfL5z4ylnASq4ZnnRvhaPc8XVq0yU
WNU
Python: ~ $
```

Factorial Using Recursion

```python
def factorial(number):
    """
    compute the factorial using recursion
    """
    if number <= 1:
        return number
    else:
        # recursion >
        return number*factorial(number-1)

try:
    ask_number = int(input("Enter a number for factorial "))
    print(f'{factorial(ask_number)} is the factorial of number {
    ask_number}')
except ValueError:
    print("Please enter a number ")
    # handling the exception
```

factorial_rec.py

output



```
Python: ~ $  python3 factorial_rec.py
Enter a number for factorial 5
120 is the factorial of number 5
Python: ~ $  python3 factorial_rec.py
Enter a number for factorial 100
933262154439441526816992388562667004907159682643816214685929638952175999932299156
08941463976156518286253697920827223758251185210916864000000000000000000000000 is
the factorial of number 100
Python: ~ $
```

Display a table from MySQL

```
"""
using the mysql-connector to display the file
"""

import mysql.connector   # for MySQL connection
import getpass       # for passwords

ask_pass = getpass.getpass()
# asking the password for the MySQL
ask_database = input("Enter the database ")
# asking the database for tables
try:
    cnx = mysql.connector.connect(user='root', passwd=ask_pass,
    database=ask_database)
    # setting up a connection to the MySQL
    cursor = cnx.cursor()
    # asking for the table
    ask_table = input("Enter the table ")

    # executing the MySQL query
    cursor.execute(f"select * from {ask_table}")
    # for new line
    print()
    # traversing over the cursor
    for lines in cursor:
        for words in lines:
            # printing the data fetched by the cursor
            print(words, '\t\t', end='')
        print()
        # for new line


except mysql.connector.errors.ProgrammingError:
    print("Wrong password or database does not exists ")
    # if the following error occurs then print this message
```

mysqldisplay.py

Updating a certain MySQL Table

```
"""
# adding data to given mysql table with defined attributes
"""

import mysql.connector      # getting the connection to MySQL
import getpass  # getting the password


ask_password = getpass.getpass()        # asking the database password
ask_database = input("Enter the database of the table ")        #
    asking the database
try:
    cnx = mysql.connector.connect(user='root', passwd=ask_password,
    database=ask_database)
    # initiating the connection
    cursor = cnx.cursor()

    # asking the details required for the table:music
    ask_song = input("Enter the song name ").strip()
    ask_artist = input("Enter the artist name ").strip().title()
    ask_album = input("Enter the album name ").strip()

    # executing the commands in mysql
    cursor.execute(f"insert into music values({ask_artist!r}, {
    ask_album!r}, {ask_song!r})")

    # committing the changes in the sql
    cnx.commit()
    print("Successfully added the data ")
    # printing the success message
except mysql.connector.errors.ProgrammingError:
    print("The password or database is wrong.")
    # if the following error is encountered print this message
```

mysqldataupdating.py

Deleting the tables in mysql

```python
"""
using the mysql connector to delete tables
"""

import mysql.connector
import getpass

ask_pass = getpass.getpass()
ask_database = input("Enter the database ")

try:
    cnx = mysql.connector.connect(user='root', passwd=ask_pass,
    database=ask_database)

    cursor = cnx.cursor()
    table_name = input("Enter the table name ")
    cursor.execute(f"drop table if exists {table_name}")
    conform = input(f"Completely delete the table {table_name!r} ").
    strip().casefold()
    if conform in ['yes', 'y', 'yep']:
        cnx.commit()
        print(f"the table {table_name} is permanently deleted")
    else:
        print("The table is not deleted")
except ValueError:
    print("The password or the database is wrong ")
```

mysqldeletetables.py

Searching the mysql table

```python
"""
searching the table with mysql
"""

import mysql.connector      # connecting to the mysql database
import getpass      # for getting the password of the database

try:
    ask_pass = getpass.getpass()
    ask_database = input("Enter the database of the table ")
    cnx = mysql.connector.connect(user='root', passwd=ask_pass,
    database=ask_database)

    cursor = cnx.cursor()
    ask_pet_name = input("Enter the pet name to search for their owner
    ")
    cursor.execute(f'select owner from pet where name like {
    ask_pet_name+"%"!r}')

    lines = cursor.fetchall()

    if not lines:
        print(f"There is no pet like {ask_pet_name}")
    else:
        if len(lines) == 1:
            print('The owner of the pet is {lines[0][0]}')

        else:
            print("The owner of the pets can be -- ")
            for owners in lines:
                print(owners[0])

except mysql.connector.errors.ProgrammingError:
    print("The password or the database is wrong")
```

mysqltablesearch.py