Python Programming Lab

Manav Rachna International Institute of Research and Studies

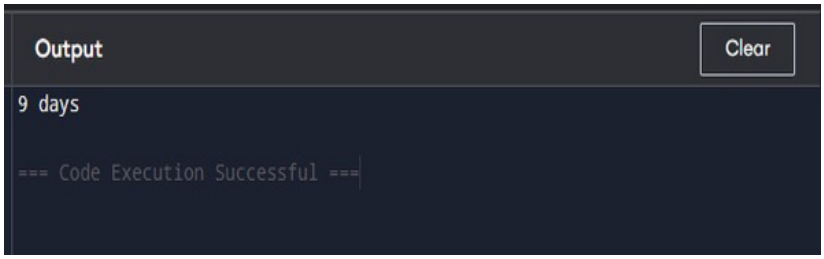
School of Computer Applications

	Submitted By
Student Name	Rachit Gautam
Roll No	25/SCA/MCAN/021
Programme	Masters of Computer Applications
Semester	1st Semester
Section	B
Department	School of Computer Applications
Batch	2025-27
	Submitted To
Faculty Name	Dr. Anupama Luthra

Manav Rachna Campus Rd, Gadakhor Basti Village, Sector 43, Faridabad, Haryana 121004

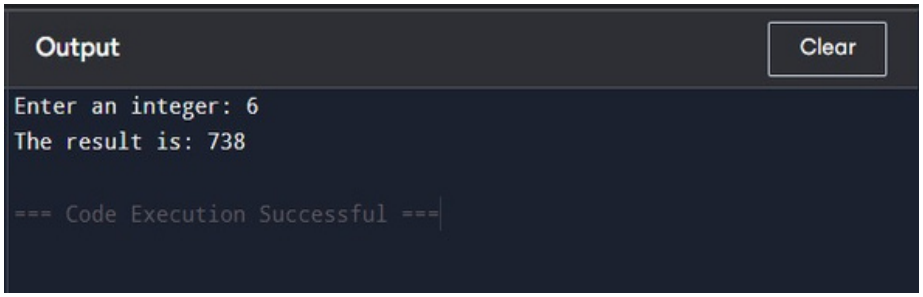
1. Write a Python program to calculate number of days between two dates.

```
from datetime import date

d1 = date(2014, 7, 2)
d2 = date(2014, 7, 11)
delta = d2 - d1
print(f"{delta.days} days")
```



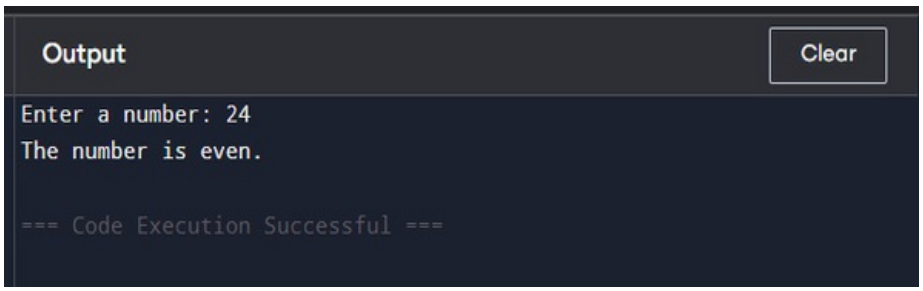
2. WriteaPython program that accepts an integer (n) and computes the value of n+nn+nnn

```
n=input("Enter an integer: ")

nn=n+n
nnn=n+n+ n

result=int(n) + int(nn) + int(nnn)

print("Theresult is:", result)
```



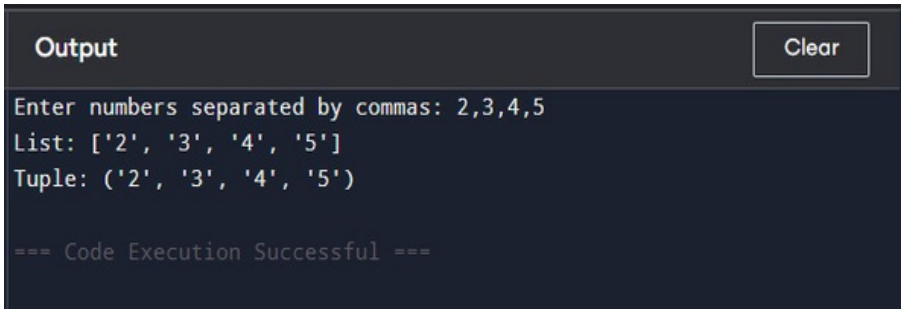
3. Ask the user for a number. Depending on whether the number is even or odd, print out an appropriate message to the user.

```
num = int(input("Enter a number: "))

ifnum % 2 == 0:
   print("The number is even.")
else:
   print("The number is odd.")
```



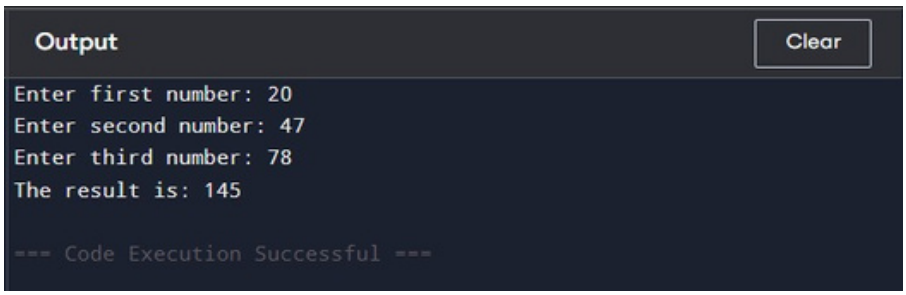
4. Write a Python program which accepts a sequence of comma-separated numbers from user and generate a list and a tuple with those numbers.

```
numbers = input("Enter numbers separated by commas: ")

num_list = numbers.split(",")
num_tuple = tuple(num_list)

print("List:", num_list)
print("Tuple:", num_tuple)
```



5. Write a Python program to calculate the sum of three given numbers, if the values are equal then return thrice of their sum.

```
a=int(input("Enter first number: "))
b=int(input("Enter second number: "))
c=int(input("Enter third number: "))

total = a + b + c

ifa== b == c:
   total *= 3

print("The result is:", total)
```

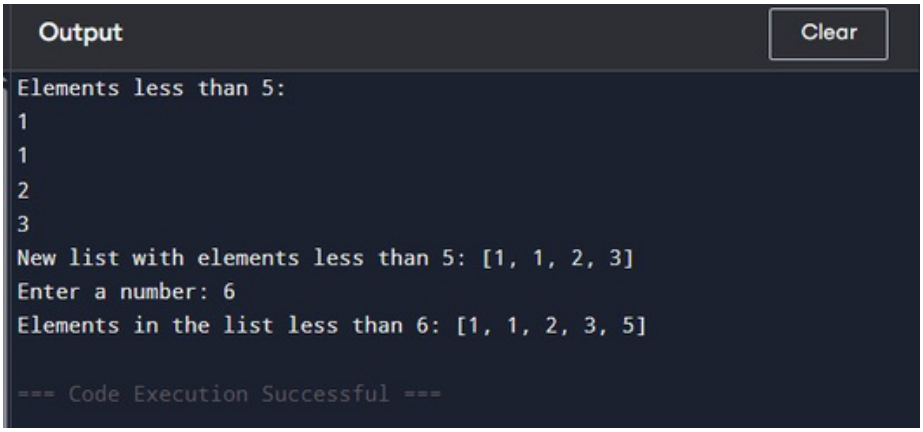


6. Write a Python program to test whether a passed letter is a vowel or not

```
letter = input("Enter a letter: ").lower()

ifletter in 'aeiou':
   print("The letter is a vowel.")
else:
   print("The letter is not a vowel.")
```

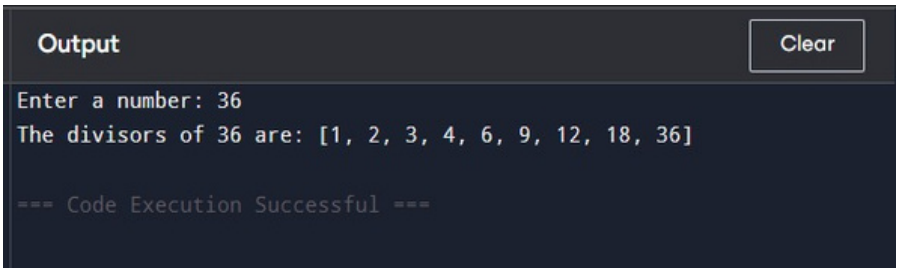
7. Takealist,sayforexamplethisone:a=[1,1,2,3,5,8,13, 21, 34, 55, 89] and write a programthatprintsoutalltheelementsofthelistthatare less than 5. Extras:
a)Insteadofprintingtheelementsonebyone,makeanew list that has all the elements less than 5 from this list in it and print out this new list.
b) Write this in one line of Python.
c)Asktheuserforanumberandreturnalistthatcontains only elements from the originallistathataresmallerthanthatnumbergivenbythe user

```
a = [1,1,2,3,5,8,13,21,34,55,89]

# Part 1: Print elements less than 5 one by one
print("Elements less than 5:")
for number in a:
if number < 5:
      print(number)

#Part2:Createanewlistwithelementslessthan5andprint it
new_list = [number for number in a if number < 5]
print("New list with elements less than 5:", new_list)


#Part 3: Ask the user for a number and filter the list based on that
n= int(input("Enter a number: "))
filtered_list = [number for number in a if number < n]
print(f"Elements in the list less than {n}:", filtered_list)
```



8. Create a program that asks the user for a number and then prints out a list of all the divisors of that number.

```
num= int(input("Enter a number: "))
divisors = []

foriin range(1, num + 1):
   ifnum % i == 0:
      divisors.append(i)

print("The divisors of", num, "are:", divisors)
```

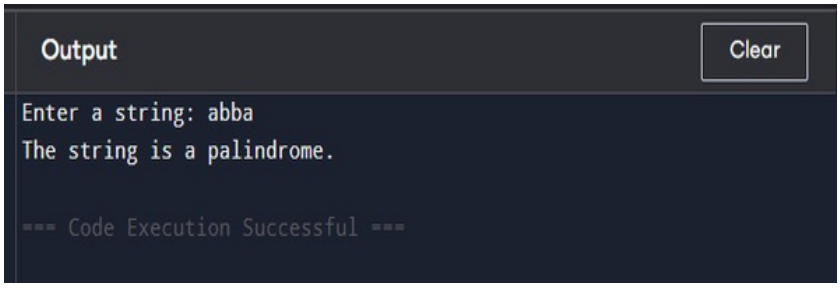


9. Take two lists, say for example these two: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes

```
a=[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
b=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

common_elements = []

forelement in a:
   ifelement in b and element not in common_elements:
      common_elements.append(element)

print("Common elements:", common_elements)
```

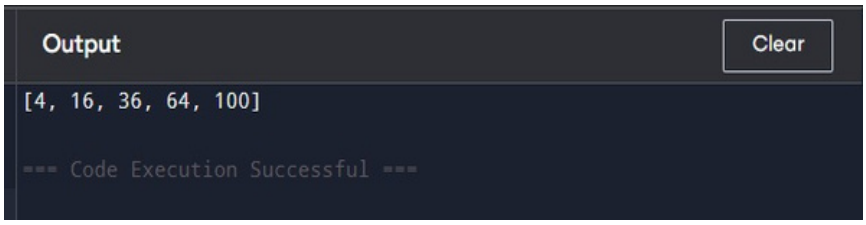
10. Ask the user for a string and print out whether this string is a palindrome or not.

```
text = input("Enter a string: ")

iftext == text[::-1]:
   print("The string is a palindrome.")
else:
   print("The string is not a palindrome.")
```



11. Let's say I give you a list saved in a variable: a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. Write one line of Python that takes this list a and makes a new list that has only the even elements of this list in it

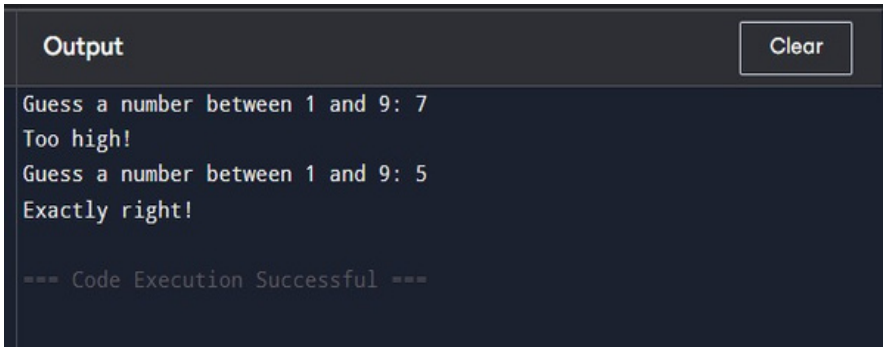
```
a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
even_elements = [x for x in a if x % 2 == 0]
print(even_elements)
```



12. Generate a random number between 1 and 9 (including 1 and 9). Ask the user to guess the number, then tell them whether they guessed too low, too high, or exactly right.

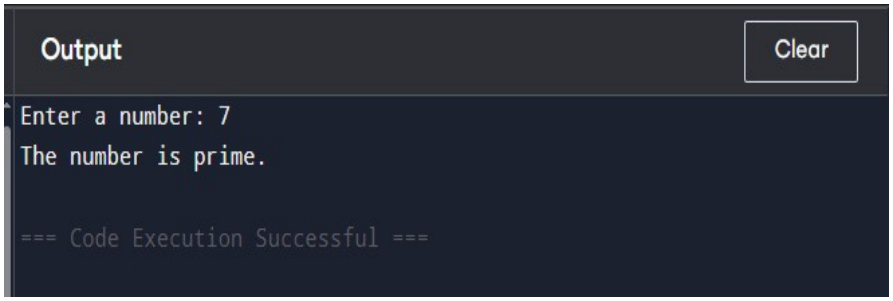
```
import random

number = random.randint(1, 9)
guess = 0

while guess != number:
   guess = int(input("Guess a number between 1 and 9: "))
   if guess < number:
```

```
    print("Too low!")
  elifguess > number:
    print("Too high!")
  else:
    print("Exactly right!")
```



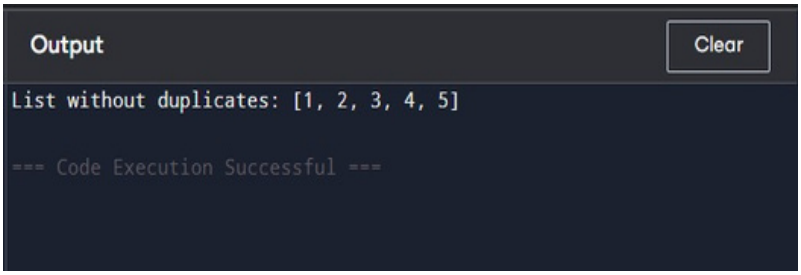
13. Asktheuser for a number and determine whether the number is prime or not.

```
num=int(input("Enter a number: "))

ifnum<=1:
  print("The number is not prime.")
else:
  is_prime = True
  foriinrange(2, num):
    ifnum % i == 0:
      is_prime = False
      break
  ifis_prime:
    print("The number is prime.")
  else:
    print("The number is not prime.")
```



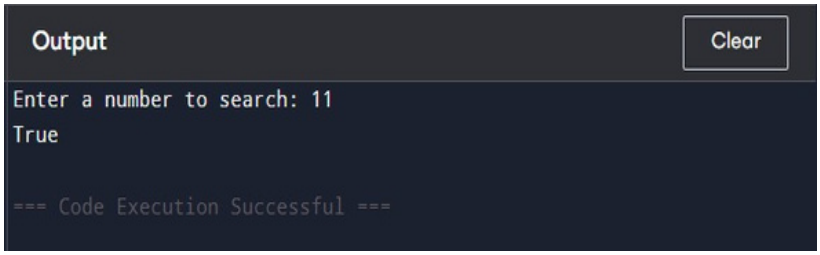
14. Write a program (function!) that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates

```
def remove_duplicates(lst):
    new_list = []
    for item in lst:
        if item not in new_list:
            new_list.append(item)
    return new_list

a = [1, 2, 2, 3, 4, 4, 5]
result = remove_duplicates(a)
print("List without duplicates:", result)
```



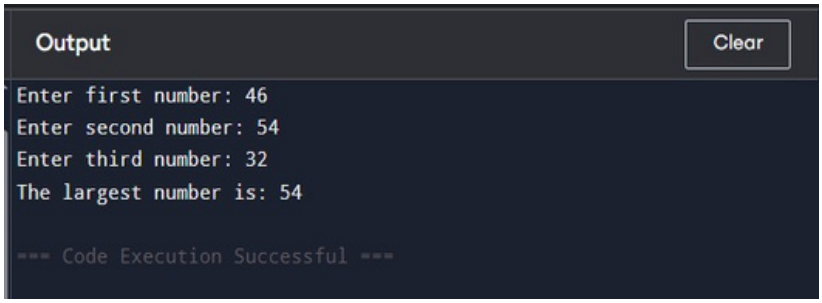
15. Write a function that takes an ordered list of numbers (a list where the elements are in order from smallest to largest) and another number. The function decides whether or not the given number is inside the list and returns (then prints) an appropriate Boolean

```
def check_number(ordered_list, number):
    for item in ordered_list:
        if item == number:
            return True
    return False

# Example usage
numbers = [1, 3, 5, 7, 9, 11, 13, 15]
num = int(input("Enter a number to search: "))

result = check_number(numbers, num)
print(result)
```



16. Implement a function that takes as input three variables, and returns the largest of the three.

```
deflargest(a, b, c):
   ifa>=b and a >= c:
      return a
   elifb>= a and b >= c:
      return b
   else:
      return c

#Example usage
x=int(input("Enter first number: "))
y=int(input("Enter second number: "))
z=int(input("Enter third number: "))

print("The largest number is:", largest(x, y, z))
```



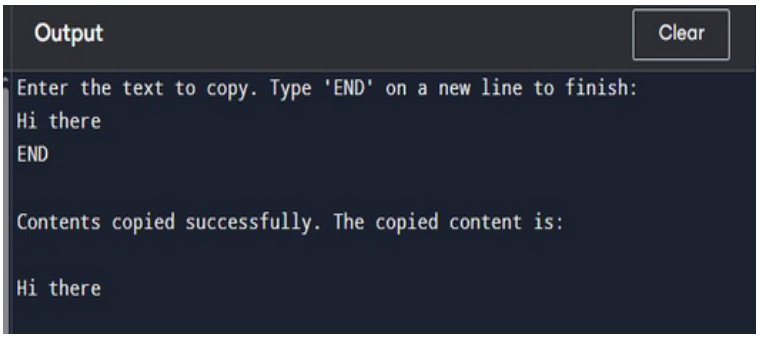
17. Python program to perform read and write operations on a file

```
#Writing to the file
file = open("example.txt", "w")
file.write("Hello, this is a sample file.\n")
file.write("We are performing read and write operations.")
file.close()

# Reading from the file
file = open("example.txt", "r")
content = file.read()
file.close()

print("File content:")
```

```
print(content)
```

18. Python program to copy the contents of a file to another file.

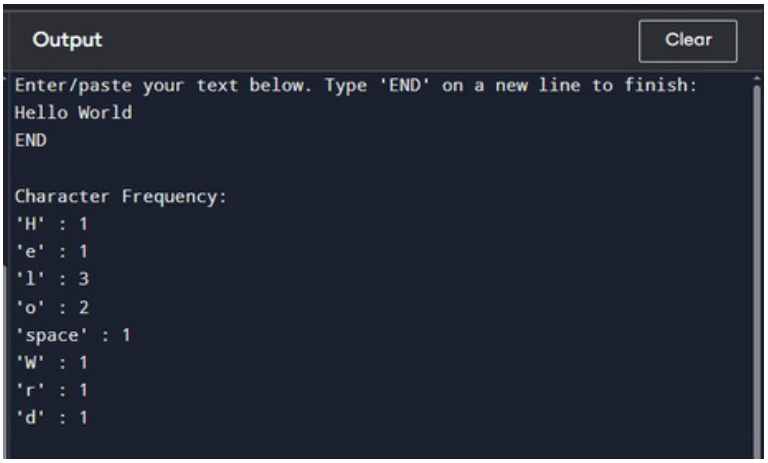
```
print("Enter the text to copy. Type 'END' on a new line to finish:")

lines = []
while True:
    line = input()
    if line == "END":
        break
    lines.append(line)

text = "\n".join(lines)

# Simulate copying to another "file" using a variable
copied_text = text

print("\nContents copied successfully. The copied content is:\n")
print(copied_text)
```



19. Python program to count frequency of characters in a given file.

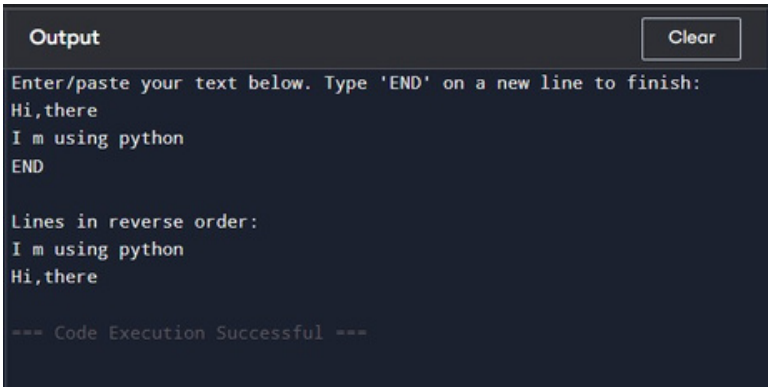
```
print("Enter/paste your text below. Type 'END' on a new line to finish:")

lines = []
while True:
    line = input()
    if line == "END":
        break
    lines.append(line)

text = "\n".join(lines)
char_freq = {}
```

```
forchar in text:
    ifchar in char_freq:
        char_freq[char] += 1
    else:
        char_freq[char] = 1

print("\nCharacter Frequency:")
forchar, freq in char_freq.items():
    ifchar == "\n":
        print("'\\n' :", freq)
    elifchar == " ":
        print("'space' :", freq)
    else:
        print(f"'{char}' :", freq)
```



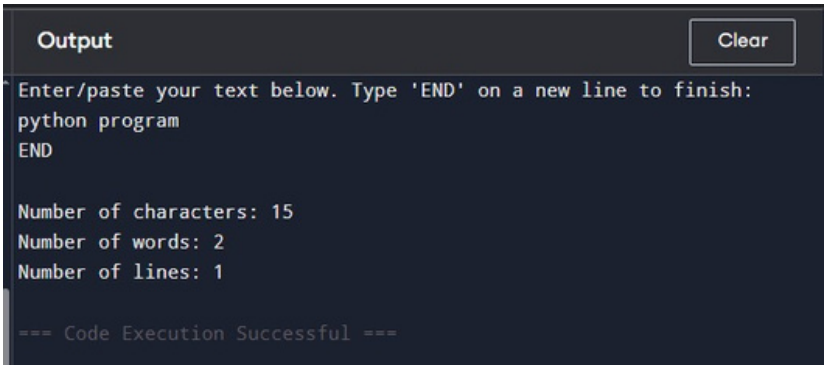
20. Python program to print each line of a file in reverse order

```
print("Enter/paste your text below. Type 'END' on a new line to finish:")
lines = []
while True:
    line=input()
    ifline=="END":
        break
    lines.append(line)
print("\nLines in reverse order:")
forlineinreversed(lines):
    print(line)
```



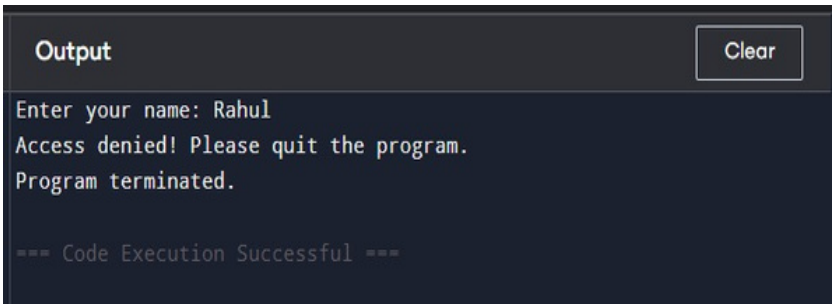
21. Python program to compute the number of characters, wordsand lines in a file

```
print("Enter/paste your text below. Type 'END' on a new line to finish:")

lines = []
while True:
    line = input()
    if line == "END":
        break
    lines.append(line)

text = "\n".join(lines)

num_chars = len(text)

num_words = len(text.split())


num_lines = len(lines)
print(f"\nNumber of characters: {num_chars}")
print(f"Number of words: {num_words}")
print(f"Number of lines: {num_lines}")
```



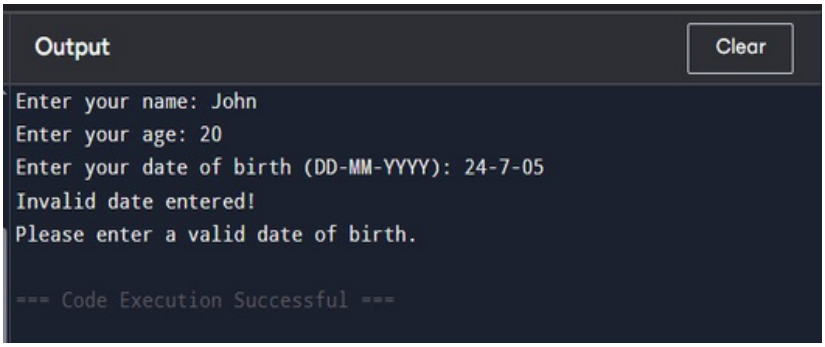
22. Write a program that prompts the user to enter his name. The program then greets the person with his name. But if the person's name is 'Rahul' and exception is thrown and he is asked to quit the program.

```
classQuitProgramException(Exception):
    pass

try:
    name = input("Enter your name: ")
    ifname == "Rahul":
        raise QuitProgramException("Access denied! Please quit the program.")
    print(f"Hello, {name}! Welcome!")
except QuitProgramException as e:
    print(e)
    print("Program terminated.")
```



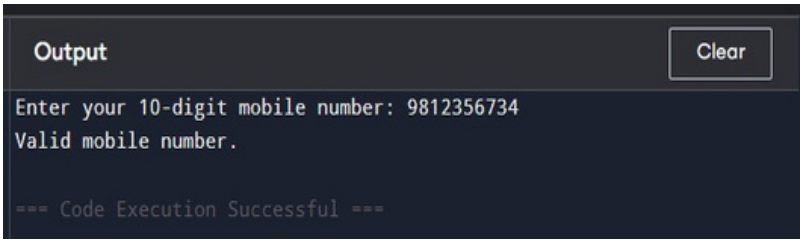
23. Writeaprogram that accepts date of birth along with the other personal details of a person.Throw an exception if an invalid date is entered

```
fromdatetime import datetime

classInvalidDateException(Exception):
    pass
try:
    name=input("Enter your name: ")
    age=input("Enter your age: ")
    dob_input = input("Enter your date of birth (DD-MM-YYYY): ")
    try:
        dob=datetime.strptime(dob_input, "%d-%m-%Y")
    exceptValueError:
        raiseInvalidDateException("Invalid date entered!")

    print("\nPersonal Details:")
    print(f"Name: {name}")
    print(f"Age: {age}")
    print(f"Date of Birth: {dob.strftime('%d-%m-%Y')}")

exceptInvalidDateException as e:
    print(e)
    print("Please enter a valid date of birth.")
```

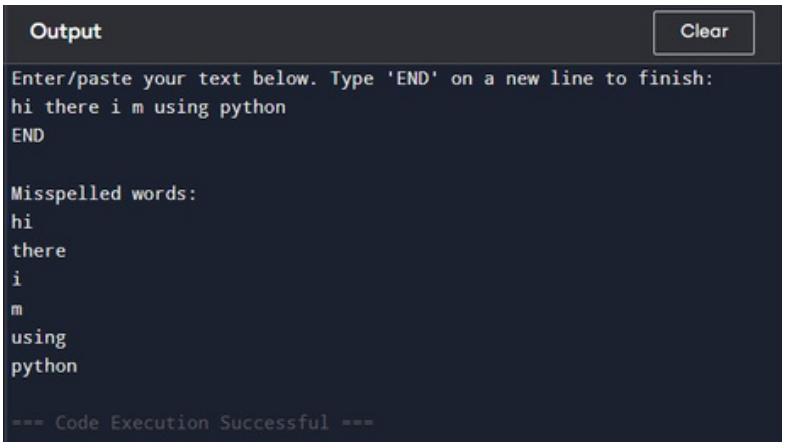


24. Write a Regular Expression to represent all 10 digit mobile numbers. Rules: 1. Every number should contains exactly 10 digits. 2. The first digit should be 7 or 8 or 9 Write a Python Program to check whether the given number is valid mobile number or not?

```
import re

pattern = r'^[789]\d{9}$'
mobile = input("Enter your 10-digit mobile number: ")

ifre.match(pattern, mobile):
   print("Valid mobile number.")
else:
   print("Invalid mobile number.")
```



25. A spell checker can be a helpful tool for people who struggle to spell words correctly. In this exercise, you will write a program that reads a file and displays all of the words in it that are misspelled. Misspelled words will be identified by checking each word in the file against a list of known words. Any words in the user's file that do not appear in the list of known words will be reported as spelling mistakes. The user will provide the name of the file to check for spelling mistakes as a command line parameter. Your program should display an appropriate error message if the command line parameter is missing. An error message should also be displayed if your program is unable to open the user's file. Words followed by a comma, period or other punctuation mark are not reported as spelling mistakes. Ignore the capitalization of the words when checking their spelling.

```
import string

known_words = {"this", "is", "a", "sample", "text", "with", "some", "words", "for", "testing"}
```

```
print("Enter/paste your text below. Type 'END' on a new line to finish:")

lines = []
while True:
   line = input()
   if line == "END":
      break
   lines.append(line)

text = "\n".join(lines)

translator = str.maketrans(string.punctuation, ' '*len(string.punctuation))
clean_text = text.translate(translator).lower()
words= clean_text.split()
misspelled = [word for word in words if word not in known_words]

ifmisspelled:
   print("\nMisspelled words:")
   forword in misspelled:
      print(word)
else:
   print("\nNo spelling mistakes found.")
```