# PYTHON PROGRAMMING LAB

## School of Computer Applications

## Department of Computer Applications

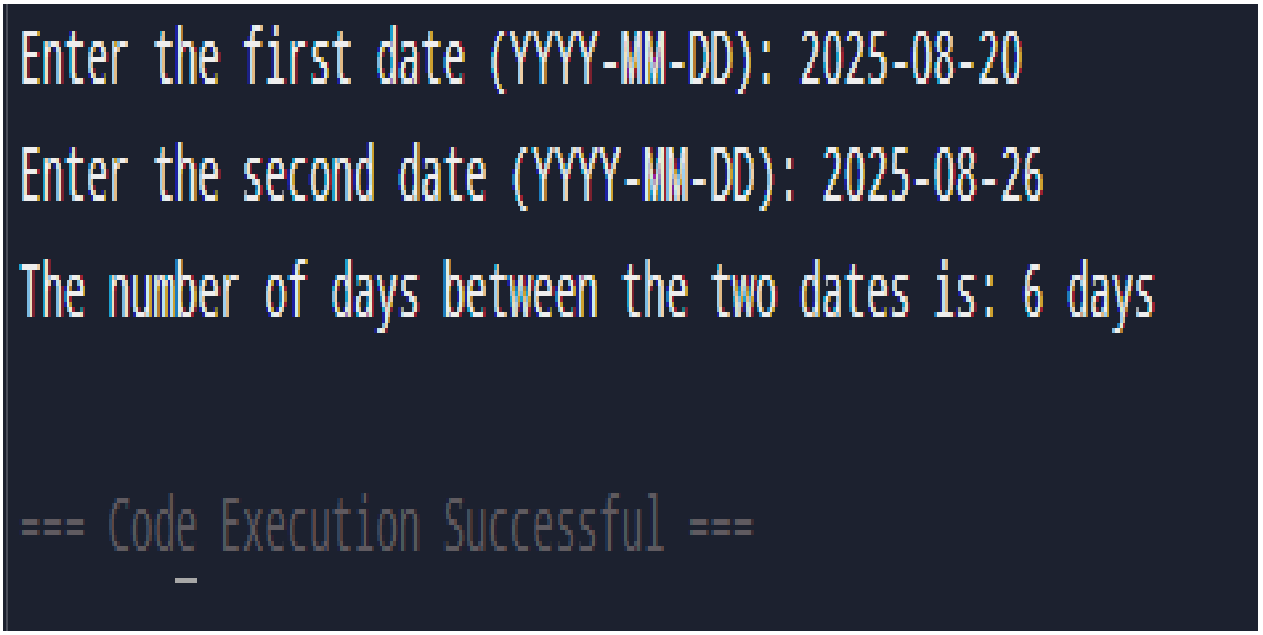
| SUBMITTED BY | |
|---|---|
| **Student Name** | Deepak Mangla |
| **Roll No** | 24/SCA/BCA(AI&ML)/015 |
| **Programme** | BCA(AI&ML) |
| **Semester** | 3rd Semester |
| **Section/Group** | 3rd C |
| **Department** | Computer Applications |
| **Batch** | 2024-2028 |

| SUBMITTED TO | |
|---|---|
| **Faculty Name** | Dr. Sakshi Gupta |

Ques 1)W.a.p. In python to calculate number of days between two dates.
Input:

```
from datetime import date

date1 = input("Enter the first date (YYYY-MM-DD): ")
date2 = input("Enter the second date (YYYY-MM-DD): ")

date1 = date.fromisoformat(date1)
date2 = date.fromisoformat(date2)

dd = abs((date2 - date1).days)

print(f"The number of days between the two dates is: {dd} days")
```

Output:

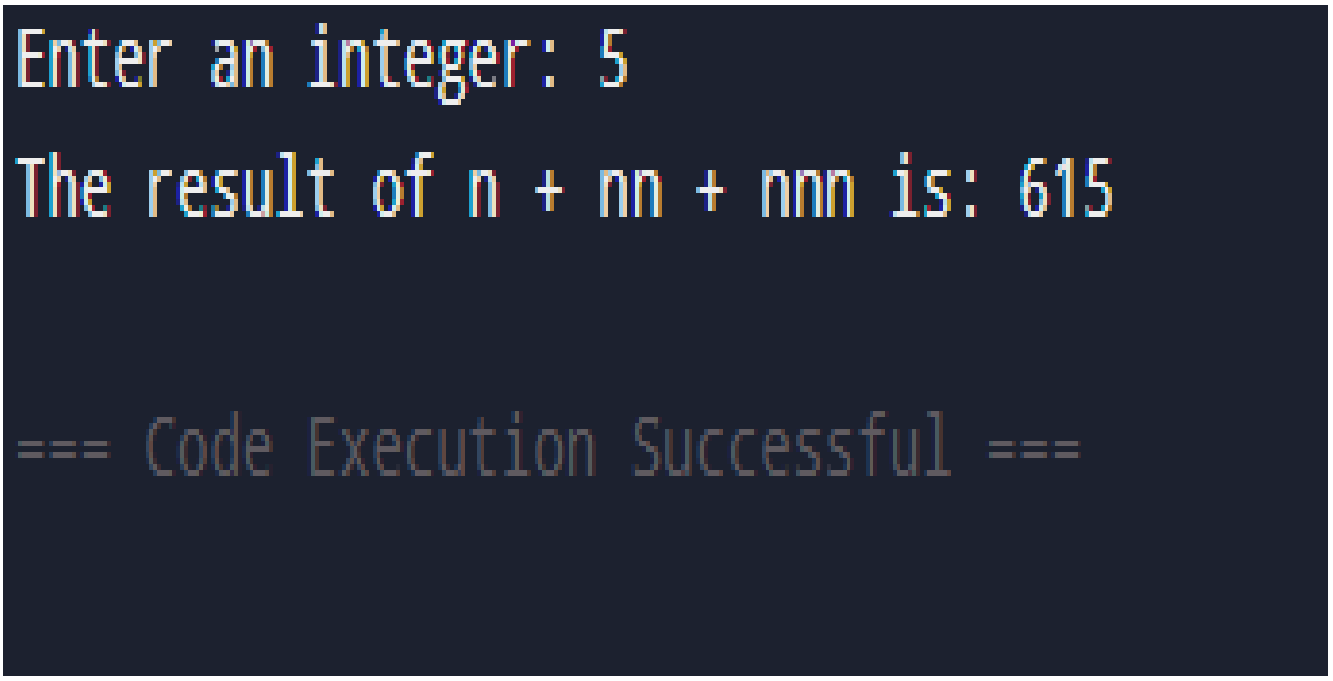


Ques 2)W.a.p. In python that accepts an integer(n) and computes the value of n+nn+nnn.

Input:

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

nn = int(str(n) * 2)
nnn = int(str(n) * 3)

result = n + nn + nnn

print(f"The result of n + nn + nnn is: {result}")
```

Output:



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

Input:

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

if number % 2 == 0:
    print(f"{number} is even.")
else:
    print(f"{number} is odd.")
```

Output:

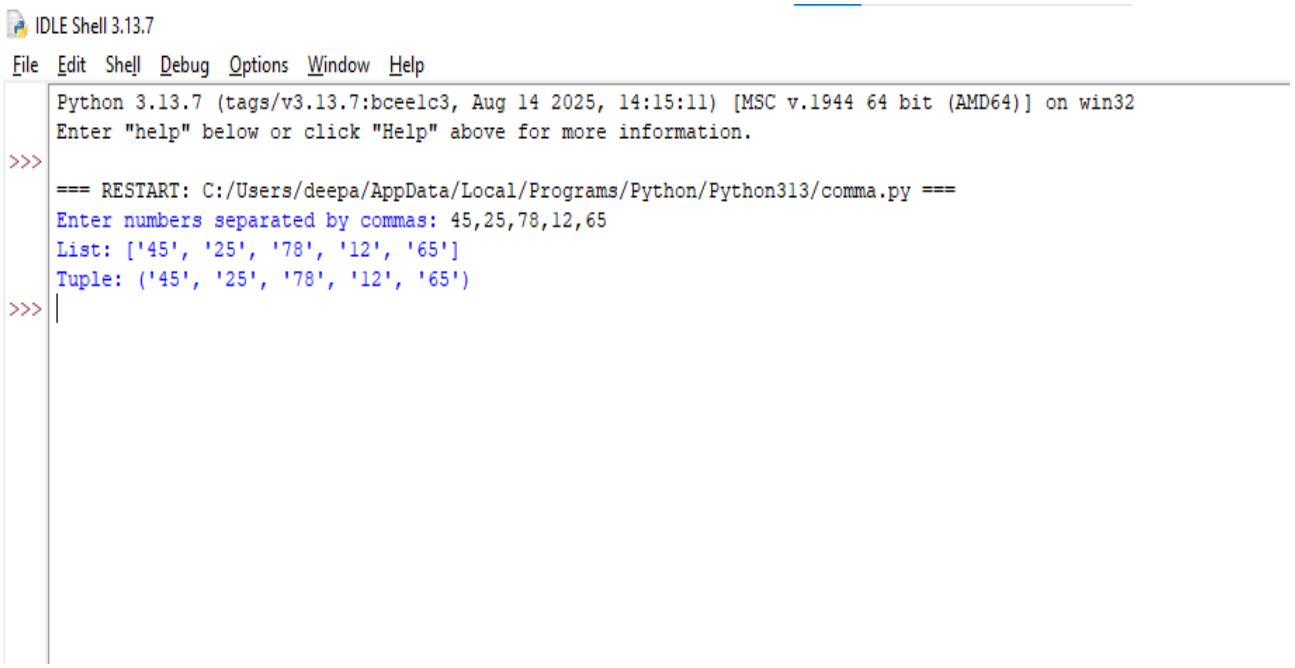
```
Enter a number: 5
5 is odd.

=== Code Execution Successful ===
```

Ques 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.

Input:

```
numbers = input("Enter numbers separated by commas: ")
num_list = numbers.split(",")
num_tuple = tuple(num_list)

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

Output:

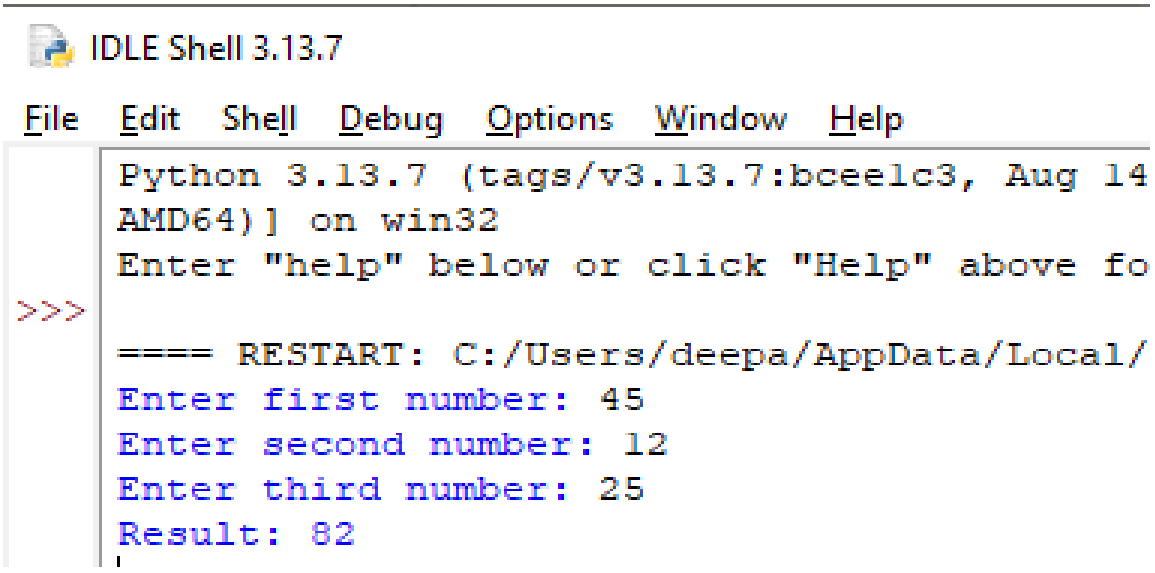


Ques 5)Write a Python program to calculate the sum of three given numbers, if the values are equal then return thrice of their sum.

Input:

```
def sum_of_three(a, b, c):
    total = a + b + c
    if a == b == c:
        return 3 * total
    else:
        return total

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

result = sum_of_three(x, y, z)
print("Result:", result)
```

Output:

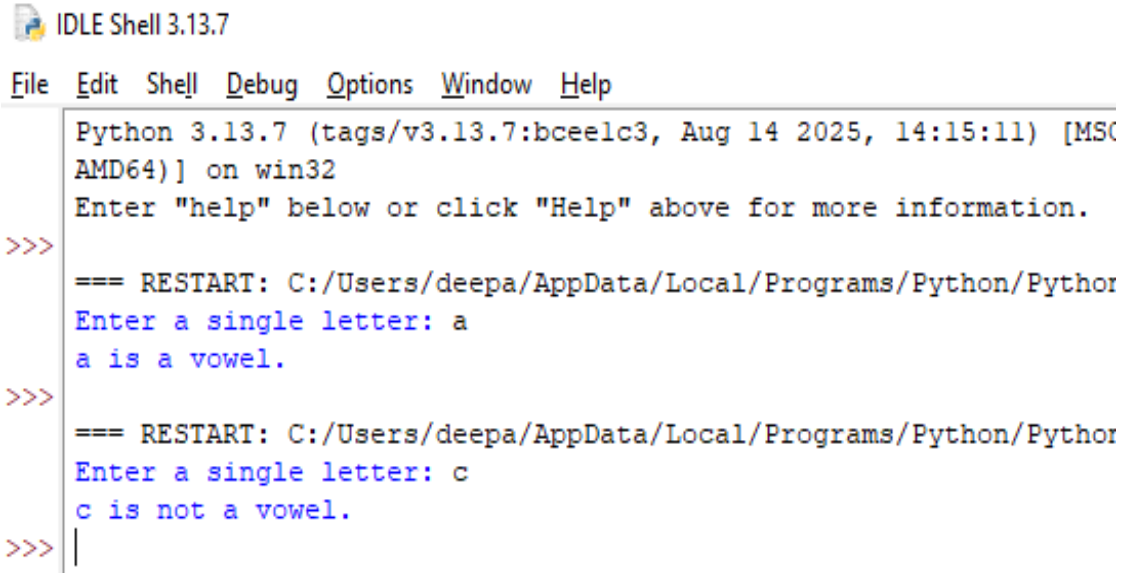


Ques 6) Write a Python program to test whether a passed letter is a vowel or not

Input:

```
def check_vowel(letter):
    vowels = "aeiouAEIOU"
    if letter in vowels:
        return True
    else:
        return False


ch = input("Enter a single letter: ")

if len(ch) == 1 and ch.isalpha():
    if check_vowel(ch):
        print(f"{ch} is a vowel.")
    else:
        print(f"{ch} is not a vowel.")
else:
    print("Please enter a valid single alphabet letter.")
```

Output:



Ques7)Take a list, say for example this one:
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
and write a program that prints out all the elements of the list that are less than 5.
Extras:

a. Instead of printing the elements one by one, make a new 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. Ask the user for a number and return a list that contains only elements from the original list a that are smaller than that number given by the user.

Input:

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
for number in a:
    if number < 5:
        print(number)
```

Output

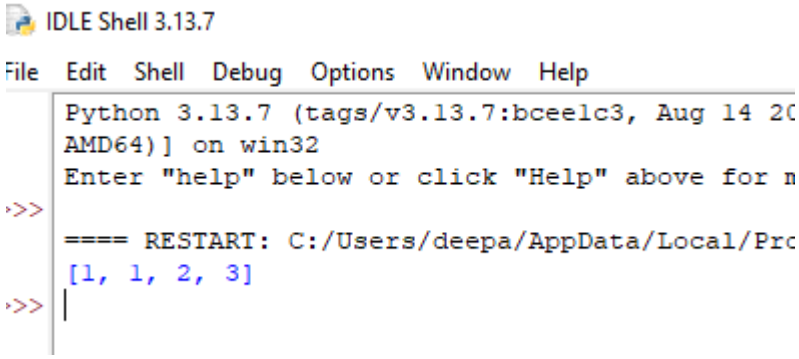
```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
for number in a:
    if number < 5:
        print(number)
```

```
1
1
2
3
```

1) Input:

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

new_list = []

for x in a:
    if x < 5:
        new_list.append(x)
print(new_list)
```

Output:

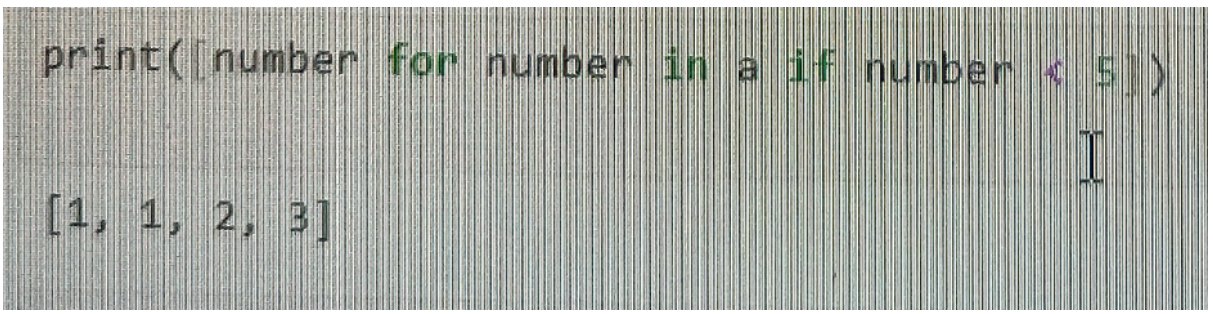



b)

Input:

```
print([i for i in a if i < 5])
```

Output:



C)
Input:

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
num = int(input("Enter a number: "))
filtered_list = [x for x in a if x < num]
print("Elements less than", num, ":", filtered_list)
```

Output:

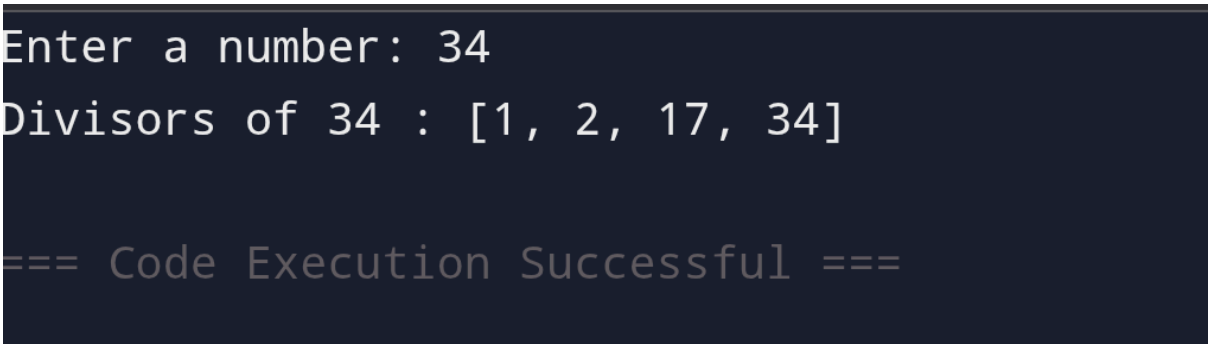
```
Enter a number: 12
Elements less than 12 : [1, 1, 2, 3, 5, 8]

=== Code Execution Successful ===
```

Ques 8) Create a program that asks the user for a number and then prints out a list of all the divisors of that number. (If you don’t know what a divisor is, it is a number that divides
evenly into another number. For example, 13 is a divisor of 26 because 26 / 13 has no remainder.)

Input:

```
num = int(input("Enter a number: "))
divisors = [i for i in range(1, num + 1) if num % i == 0]
print("Divisors of", num, ":", divisors)
```

Output:

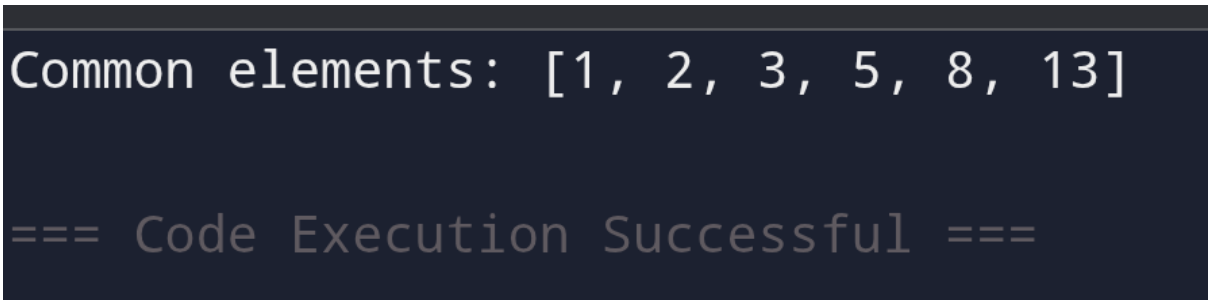


Ques 9)  Take two lists, say for example these two: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on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

Input:
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 = list(set(a) & set(b))
print("Common elements:", common)
```

Output:

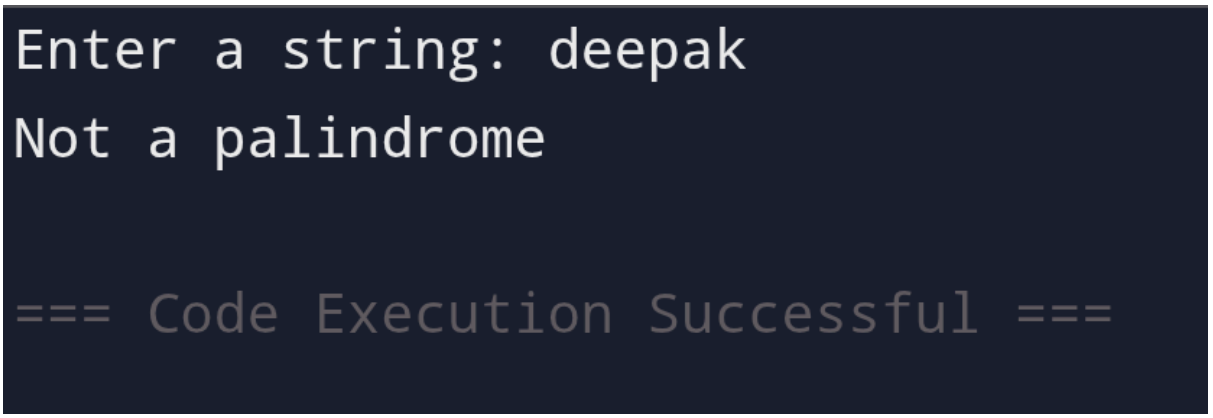


Ques 10). Ask the user for a string and print out whether this string is a palindrome or not. (A palindrome is a string that reads the same forwards and backwards.)

Input:

```
s = input("Enter a string: ")
if s == s[::-1]:
    print("Palindrome")
else:
    print("Not a palindrome")
```

Output:

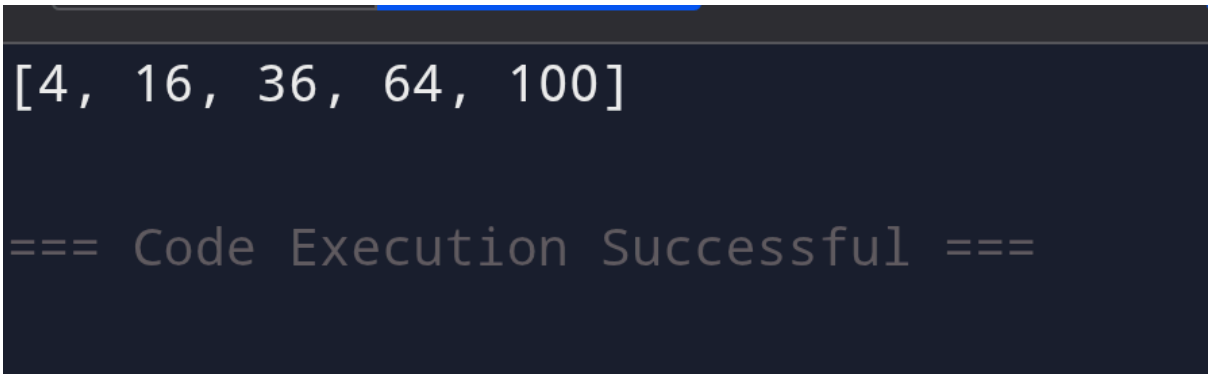


Ques 11) Let’s say I give you a list saved in a variable: a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. Write one line of Python that takes this list a and makes a new list that has only the even elements of this list in it.

Input:

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

Output:

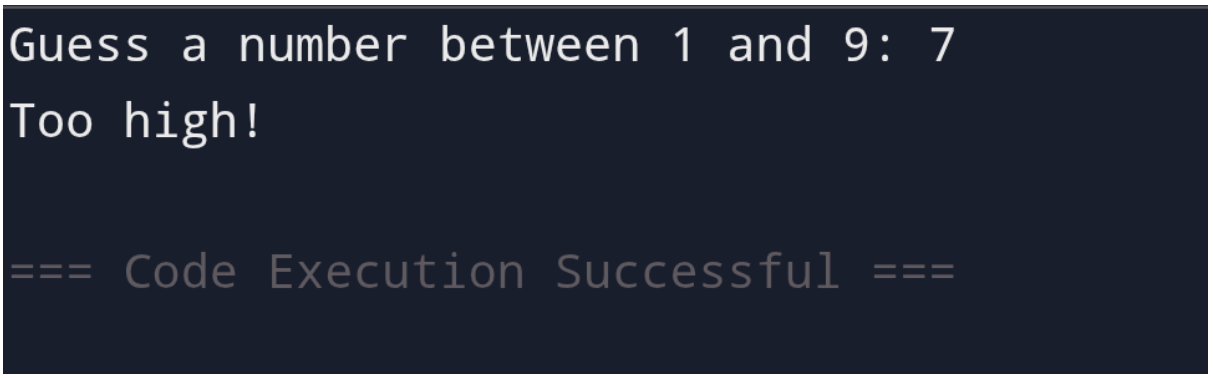


Ques 12) Generate a random number between 1 and 9 (including 1 and 9). Ask the user to guess the number, then tell them whether they guessed too low, too high, or exactly right. (Hint: remember to use the user input lessons from thevery first exercise)

Input:

```
import random

num = random.randint(1, 9)
guess = int(input("Guess a number between 1 and 9: "))

if guess < num:
    print("Too low!")
elif guess > num:
    print("Too high!")
else:
    print("Exactly right! ")
```

Output:

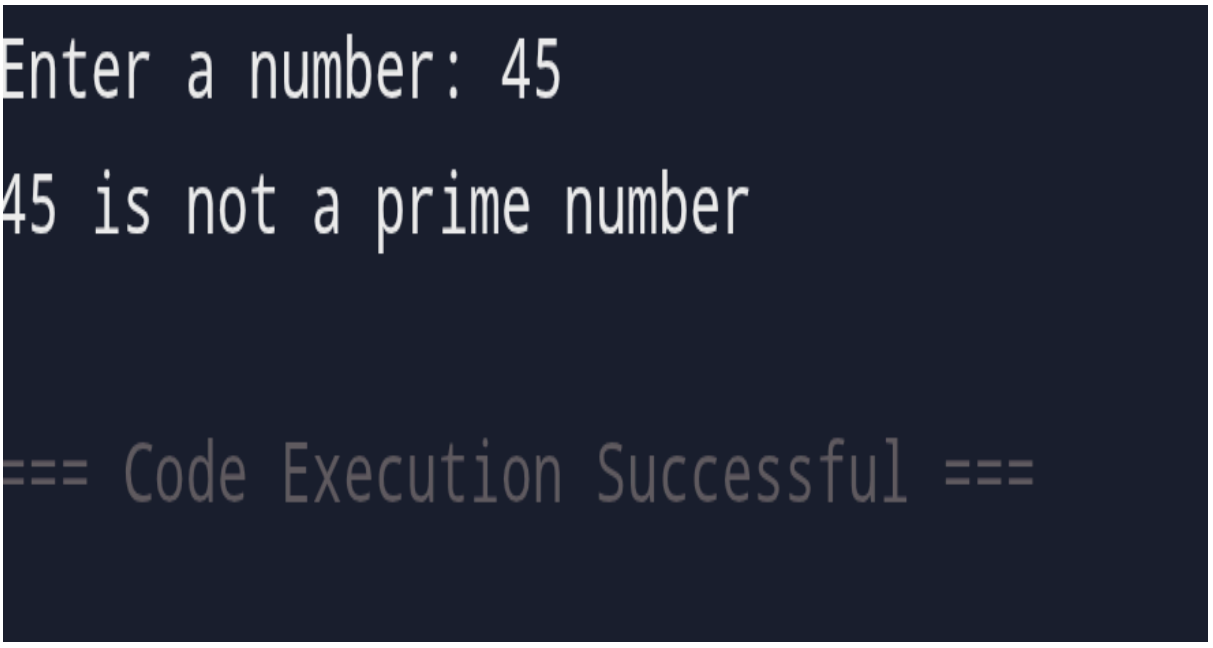


Ques 13) Ask the user for a number and determine whether the number is prime or not. (For those who have forgotten, a prime number is a number that has no divisors.).

Input:

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

if num > 1:
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            print(num, "is not a prime number")
            break
    else:
        print(num, "is a prime number")
else:
    print(num, "is not a prime number")
```

Output:



Ques 14)Write a program to remove all duplicate elements from a list nums = [1, 2, 2, 3, 4, 4, 5] and print the unique list in ascending order.

Input:
nums = [1, 2, 2, 3, 4, 4, 5]

```
unique_nums = sorted(set(nums))

print("Unique list in ascending order:", unique_nums)
```

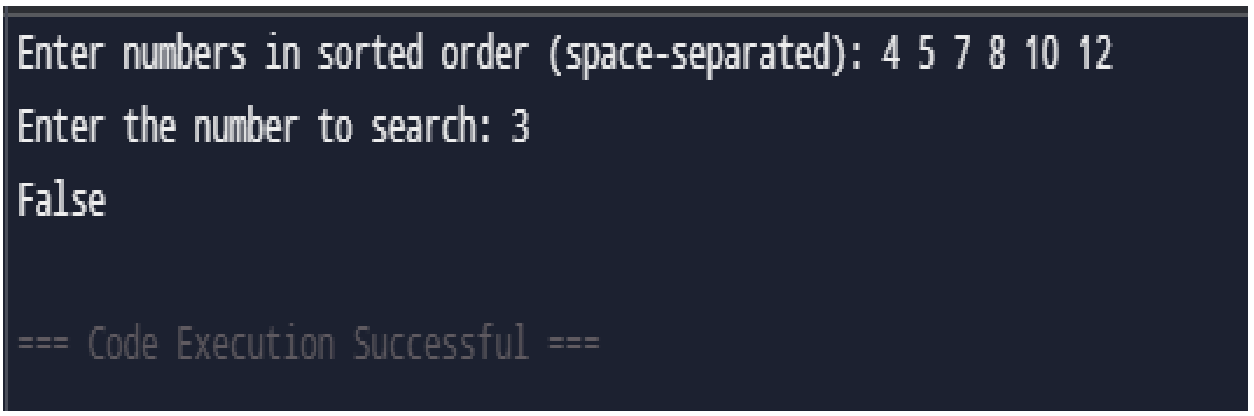
Output:

Ques 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.

Input:

```
def find_number(num_list, target):
    result = target in num_list
    print(result)
    return result
nums = input("Enter numbers in sorted order (space-separated): ")
num_list = list(map(int, nums.split()))

target = int(input("Enter the number to search: "))

find_number(num_list, target)
```

Output:

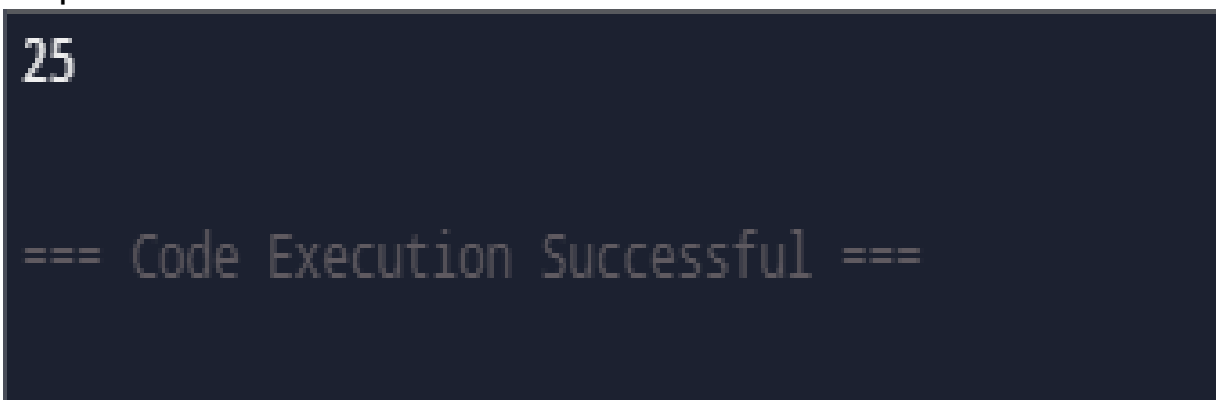


Ques 16)Implement a function that takes as input three variables, and returns the largest of the three. Do this without using the Python max() function!

Input:

```
def largest_of_three(a, b, c):
    largest = a

    if b > largest:
        largest = b

    if c > largest:
        largest = c

    return largest

print(largest_of_three(10, 25, 15))
```

Output:



Ques 17)Python program to perform read and write operations on a file.
Input:

```
# Write to a file
with open("sample.txt", "w") as file:
    file.write("Hello, this is a sample file.\n")
    file.write("Python file handling example.\n")

# Read from the same file
with open("sample.txt", "r") as file:
    content = file.read()
    print("File Content:")
    print(content)
```

Output:

```
File Content:
Hello, this is a sample file.
Python file handling example.
```

Ques 18)Python program to copy the contents of a file to another file.
Input:

```
with open("source.txt", "r") as src:

    with open("destination.txt", "w") as dest:
        for line in src:
            dest.write(line)

print("File copied successfully!")
```

Ques 19)Python program to count frequency of characters in a given file.
Input:

```
with open("sample.txt", "r") as file:
    content = file.read()

frequency = {}

for char in content:
    if char in frequency:
        frequency[char] += 1
    else:
        frequency[char] = 1

for char, count in frequency.items():
    print(f"'{char}' : {count}")
```

Ques 20)Python program to print each line of a file in reverse order.
Input:

```
with open("sample.txt", "r") as file:
    for line in file:
                reversed_line = line.strip()[::-1]
        print(reversed_line)
```

Output:

```
Hello world
Python is fun
```

Output:

```
powershell

dlrow olleH
nuf si nohtyP
```

Ques 21)Python program to compute the number of characters, words and lines in a file.

Input:

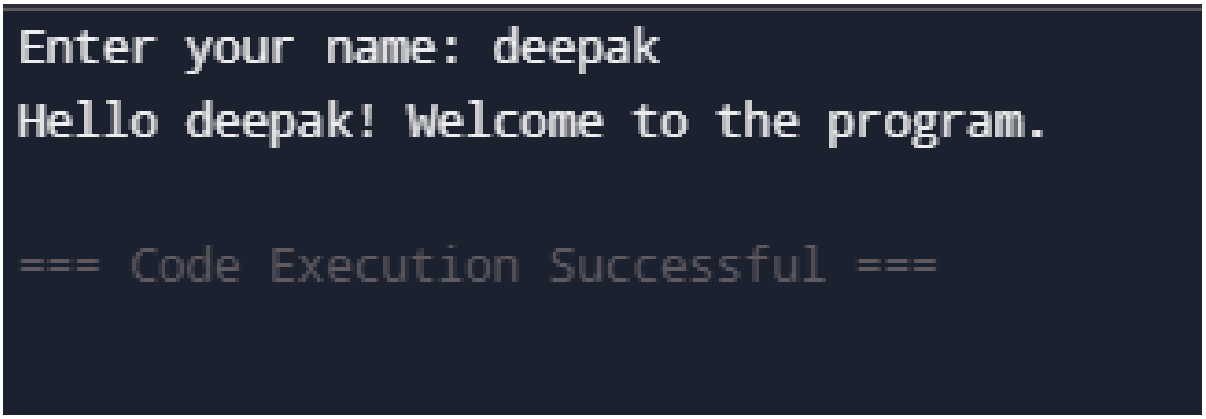
```
filename = input("Enter the file name: ")

try:
    with open(filename, "r") as file:
        text = file.read()

        num_characters = len(text)

        num_words = len(text.split())

        num_lines = len(text.splitlines())

        print("Number of characters:", num_characters)
        print("Number of words:", num_words)
        print("Number of lines:", num_lines)

except FileNotFoundError:
    print("File not found! Please check the file name.")
```

Ques 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.

Input:

```
class NameErrorException(Exception):
    pass

try:
    name = input("Enter your name: ")

    if name.lower() == "rahul":
        raise NameErrorException("Access denied for Rahul!")

    print("Hello", name + "! Welcome to the program.")

except NameErrorException as e:
    print(e)
    print("Please quit the program.")
```

Output:

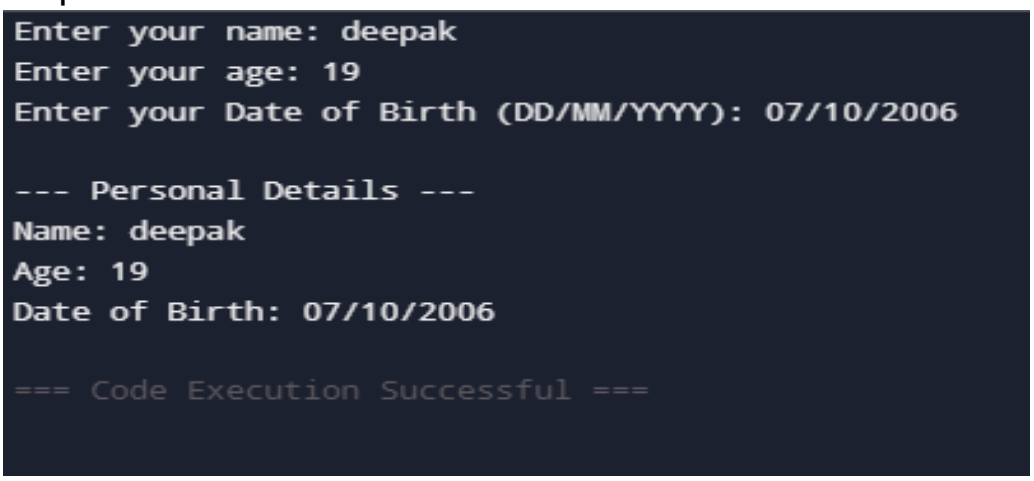


Ques 23)Write a program that accepts date of birth along with the other personal details of a person. Throw an exception if an invalid date is entered.

Input:

```
from datetime import datetime

def validate_date(dob):
    try:
        valid_date = datetime.strptime(dob, "%d/%m/%Y")
        return valid_date
    except ValueError:

        raise ValueError("Invalid date of birth entered!")
try:
    name = input("Enter your name: ")
    age = input("Enter your age: ")
    dob = input("Enter your Date of Birth (DD/MM/YYYY): ")
    valid_date = validate_date(dob)

    print("\n--- Personal Details ---")
    print("Name:", name)
    print("Age:", age)
    print("Date of Birth:", valid_date.strftime("%d/%m/%Y"))

except ValueError as e:
    print(e)
```

Output:

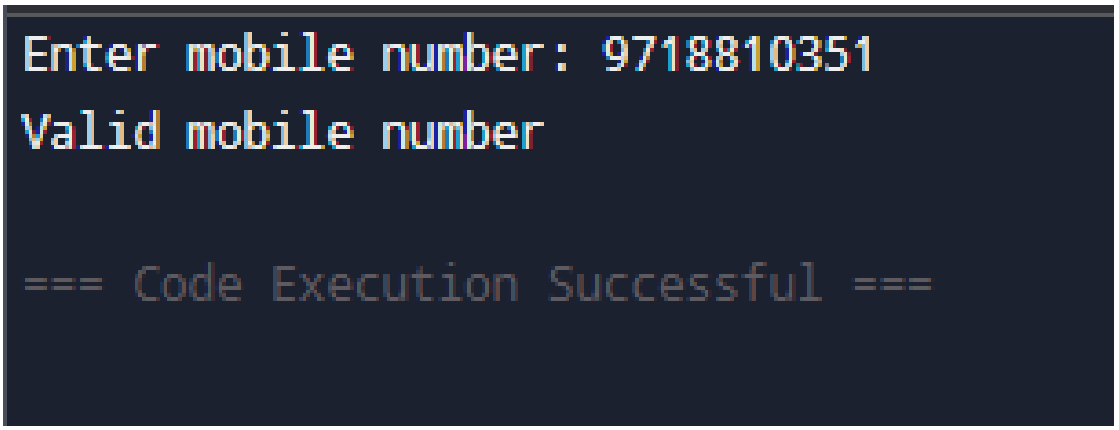


Ques 24)Write a Regular Expression to represent all 10 digit mobile numbers. Rules: 1. Every number should contains exactly 10 digits. 2. The first digit should be 7 or 8 or 9 Write a
Python Program to check whether the given number is valid mobile number or not?

Input:

```
import re

pattern = r'^[789]\d{9}$'

mobile = input("Enter mobile number: ")

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

Output:



Ques 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…

Input:

```
import sys
import string

def main():
    if len(sys.argv) < 2:
            print("Error: Please provide the filename as a command line argument.")
        return

    filename = sys.argv[1]

    known_words = {
        "the", "and", "is", "are", "you", "this", "that", "be", "to", "from",
        "with", "file", "check", "word", "words", "spell", "spelling", "for",
        "helpful", "tool", "people", "correctly", "list", "known", "will",
        "not", "appear", "reported", "mistakes", "ignore", "capitalization",
        "program", "reads", "user", "provide", "name", "open", "error"
    }

    try:
        with open(filename, "r") as f:
            text = f.read()
    except:
        print("Error: Unable to open the file.")
        return

    for p in string.punctuation:
        text = text.replace(p, "")

    words = text.split()

    print("Misspelled words:")
    found = False

    for w in words:
        word = w.lower()
        if word not in known_words:
```

```
            print(w)
            found = True

    if not found:
        print("No spelling mistakes found!")

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