Labs Manual

1.Write a program to take one list of string and check that in which vowel letter occur then print these string only.

**lis=['Noman','Arslan',"bb"]**

**for name in lis:**

**if ('a' or 'e' or 'i' or 'o' or 'u') in name:**

**print(name)**

2. Take 10 input from the user and divide input list index values equally to 3 new lists

inputs = []

for i in range(10):

user\_input = input(f"Enter input {i + 1}: ")

inputs.append(user\_input)

print("Inputs:", inputs)

my\_list = inputs

c1 = my\_list[0:3]

c2 = my\_list[3:7]

c3 = my\_list[7:10]

print("c1:", c1)

print("c2:", c2)

print("c3:", c3)

3.Count Number of vowel occurrence in string.

user\_input = input("Enter a name: ")

count = 0

for i in user\_input:

if ("a") in i or ("e") in i or ("i")in i or ("o")in i or ("u") in i:

count = count + 1

print(i)

print("Total vowels are : " , count)

*# 1.Print "Hello, World!".  
# 2. Print your name.  
# 3. Print a message using a variable.  
# 4. Take user input and print it.  
# 5. Perform addition, subtraction, multiplication, and division of two  
# numbers.  
# 6. Store and print a boolean.  
# 7. Convert a string to an integer.  
# 8. Check if a number is positive, negative, or zero.*

print("Hello, World!")

name = "Your Name"

print("My name is", name)

message = "Program No 03."

print(message)

user\_input = input("Enter something: ")

print("You entered:", user\_input)

*# Taking user input for two numbers*

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

*# Addition*

addition\_result = num1 + num2

print("Addition result:", addition\_result)

*# Subtraction*

subtraction\_result = num1 - num2

print("Subtraction result:", subtraction\_result)

*# Multiplication*

multiplication\_result = num1\* num2

print("Multiplication result:", multiplication\_result)

*# Division (check for division by zero)*

**if** num2 != 0:

    division\_result = num1 / num2

    print("Division result:", division\_result)

**else**:

    print("Cannot divide by zero.")

my\_boolean = **True**

print("Boolean:", my\_boolean)

my\_string = "123" *# A string containing a number*

my\_integer = int(my\_string)

print("Converted Integer:", my\_integer)

number = float(input("Enter a number: "))

**if** number > 0:

    print("Positive")

**elif** number< 0:

    print("Negative")

**else**:

    print("Zero")

*# 9. Check if a number is even or odd.  
# 10.Determine the largest of three numbers.  
# 11.Check if a year is a leap year.  
# 12.Check if a number is prime.  
# 13.Check if a number is a palindrome.  
# 14.Find the maximum of two numbers using a function.  
# 15.Check if a character is a vowel or consonant.*

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

**if** number % 2 == 0:

    print("Even")

**else**:

    print("Odd")

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

num3 = float(input("Enter the third number: "))

largest = max(num1, num2, num3)

print("The largest number is:", largest)

year = int(input("Enter a year: "))

**if** (year % 4 == 0 **and** year % 100 != 0) **or**(year % 400 == 0):

    print("Leap year")

**else**:

    print("Not a leap year")

**def** **is\_prime**(number):

**if** number <= 1:

**return** **False**

**for** i **in** range(2, int(number\*\*0.5) + 1):

**if** number % i == 0:

**return** **False**

**return** **True**

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

**if** is\_prime(number):

    print("Prime")

**else**:

    print("Not prime")

**def** **is\_palindrome**(string):

**return** string == string[::- 1]

word = input("Enter a word: ")

**if** is\_palindrome(word):

    print("Palindrome")

**else**:

    print("Not a palindrome")

**def** **max\_of\_two\_numbers**(num1, num2):

**return** max(num1, num2)

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

maximum = max\_of\_two\_numbers(num1, num2)

print("The maximum number is:", maximum)

*# 16.Find the roots of a quadratic equation.  
# 17.Print numbers from 1 to 10 using a while loop.  
# 18.Print even numbers from 1 to 20 using a for loop.  
# 19.Print the Fibonacci sequence up to a certain number.  
# 20.Calculate the factorial of a number.*

char = input("Enter a character: ")

**if** char.isalpha() **and** len(char) == 1:

**if** char **in** "aeiouAEIOU":

        print("Vowel")

**else**:

        print("Consonant")

**else**:

    print("Not a valid character")

**import** math  
a = float(input("Enter the coefficient a: "))

b = float(input("Enter the coefficient b: "))

c = float(input("Enter the coefficient c: "))

*# Calculate the discriminant*

D = b\*\*2 - 4\* a\* c

**if** D > 0:

    root1 = (-b + math.sqrt(D)) / (2\* a)

    root2 = (-b - math.sqrt(D)) / (2\* a)

    print("Two distinct real roots:", root1, "and", root2)

**elif** D == 0:

    root = -b / (2\* a)

    print("One real root:", root)

**else**:

    real\_part = -b / (2\* a)

    imaginary\_part = math.sqrt(-D) / (2\* a)

    print("Two complex roots:", real\_part, "+", imaginary\_part, "i and",

    real\_part, "-", imaginary\_part, "i")

    number = 1

**while** number <= 10:

    print(number)

    number += 1

**for** number **in** range(2, 21, 2):

        print(number)

**def** **fibonacci**(n):

    a, b = 0, 1

**while** a<n:

        print(a)

        a, b = b, a + b

*# Call the function with the desired upper limit*

fibonacci(100) *# This will print the Fibonacci sequence up to 100*

**def** **factorial**(n):

**if** n == 0:

**return** 1

**else**:

**return** n\* factorial(n - 1)

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

fact = factorial(number)

print("Factorial of", number, "is", fact)

*# 21.Display a multiplication table.  
# 22.Find the sum of natural numbers up to a given number.  
# 23.Reverse a number.  
# 24.Create and print a list of fruits.  
# 25.Access and print elements of a list.  
# 26.Append an item to a list.  
# 27.Sort a list of numbers.  
# 28.Check if an element exists in a list.*

num = int(input("Enter a number for the multiplication table: "))

**for** i **in** range(1, 11):

    product = num\* i

    print(num, "x", i, "=", product)

n = int(input("Enter a positive integer: "))

**if** n< 0:

    print("Please enter a positive integer.")

**else**:

    sum\_of\_natural\_numbers = (n\* (n + 1)) // 2

print("The sum of natural numbers up to", n, "is", sum\_of\_natural\_numbers)

**def** **reverse\_number**(number):

    reversed\_num = 0

**while** number > 0:

        digit = number % 10

        reversed\_num = reversed\_num\* 10 + digit

        number //= 10

**return** reversed\_num  
number = int(input("Enter a number to reverse: "))

reversed\_num = reverse\_number(number)

print("Reversed number:", reversed\_num)

fruits = ["apple", "banana", "cherry", "date", "fig", "grape"]

print("List of fruits:", fruits)

fruits = ["apple", "banana", "cherry", "date", "fig", "grape"]

print("First fruit:", fruits[0])

print("Second fruit:", fruits[1])

fruits = ["apple", "banana", "cherry"]

fruits.append("date")

print("Updated list of fruits:", fruits)

numbers = [4, 1, 8, 3, 10, 6]

numbers.sort()

print("Sorted list of numbers:", numbers)

fruits = ["apple", "banana", "cherry"]

**if** "banana" **in** fruits:

    print("Banana exists in the list of fruits.")

**else**:

    print("Banana does not exist in the list of fruits.")

*# 29.Create a function to add two numbers.  
# 30.Create a function to find the square of a number.  
# 31.Create a function to check if a number is even or odd.  
# 32.Create a function to calculate the area of a rectangle.  
# 33.Create a function to convert Celsius to Fahrenheit.  
# 34.Create a function to calculate the factorial of a number.  
# 35.Create a function to find the greatest common divisor (GCD) of two  
# numbers.  
# 36.Create a function to check if a string is a palindrome.*

**def** **add\_numbers**(a, b):

**return** a + b  
result = add\_numbers(5, 7)

print("Sum:", result)

**def** **square**(number):

**return** number\*\*2

result = square(5)

print("Square:", result)

**def** **is\_even\_or\_odd**(number):

**if** number % 2 == 0:

**return** "Even"

**else**:

**return** "Odd"

result = is\_even\_or\_odd(7)

print("Number is:", result)

**def** **rectangle\_area**(length, width):

**return** length\* width  
result = rectangle\_area(5, 4)

print("Rectangle Area:", result)

**def** **celsius\_to\_fahrenheit**(celsius):

**return** (celsius\* 9/5) + 32

result = celsius\_to\_fahrenheit(25)

print("Fahrenheit:", result)

**def** **factorial**(n):

**if** n == 0:

**return** 1

**else**:

**return** n\* factorial(n - 1)

result = factorial(5)

print("Factorial:", result)

**import** math

**def** **find\_gcd**(a, b):

**return** math.gcd(a, b)

result = find\_gcd(12, 18)

print("GCD:", result)

**def** **is\_palindrome**(s):

    s = s.lower() *# Convert to lowercase for case-insensitive comparison*

    s = s.replace(" ", "") *# Remove spaces*

**return** s == s[::- 1]

result = is\_palindrome("racecar")

print("Is Palindrome:", result)

*# 37.Concatenate two strings.  
# 38.Find the length of a string.  
# 39.Convert a string to uppercase and lowercase.  
# 40.Count the occurrences of a character in a string.  
# 41.Reverse a string.  
# 42.Check if a string is a palindrome.  
# 43.Replace a substring in a string.  
# 44.Create and print a dictionary of student names and ages.*

str1 = "Hello"

str2 = "World"

concatenated\_string = str1 + " " + str2

print("Concatenated String:", concatenated\_string)

my\_string = "This is a sample string."

length = len(my\_string)

print("Length of the string:", length)

my\_string = "Hello, World!"

uppercase\_string = my\_string.upper()

lowercase\_string = my\_string.lower()

print("Uppercase:", uppercase\_string)

print("Lowercase:", lowercase\_string)

my\_string = "Hello, World!"

char\_to\_count = "l"

count = my\_string.count(char\_to\_count)

print(f"The character '{char\_to\_count}' appears {count} times in the string.")

my\_string = "Python"

reversed\_string = my\_string[::- 1]

print("Reversed String:", reversed\_string)

**def** **is\_palindrome**(s):

    s = s.lower()

    s = s.replace(" ", "")

**return** s == s[::- 1]

    my\_string = "racecar"

**if** is\_palindrome(my\_string):

        print("Palindrome")

**else**:

        print("Not a palindrome")

my\_string = "I like programming in Python."

new\_string = my\_string.replace("Python", "Java")

print("Replaced String:", new\_string)

students = {

"Alice": 20,

"Bob": 22,

"Charlie": 19,

"David": 21

}

print("Dictionary of Students:", students)

**--------------------------------------------------------------------------------**

*# 45.Access and print values from a dictionary.  
# 46.Add a new key-value pair to a dictionary.  
# 47.Remove a key-value pair from a dictionary.  
# 48.Check if a key exists in a dictionary.  
# 49.Create a text file and write to it.  
# 50.Read from a text file.  
# 51.Append text to an existing file.*

students = {

"Alice": 20,

"Bob": 22,

"Charlie": 19,

"David": 21

}

print("Age of Alice:", students["Alice"])

print("Age of Charlie:", students["Charlie"])

students = {

 "Alice": 20,

 "Bob": 22,

 "Charlie": 19,

 "David": 21

}

students["Eva"] = 18

print("Updated Dictionary:", students)

students = {

"Alice": 20,

"Bob": 22,

"Charlie": 19,

"David": 21

}

**del** students["Bob"]

print("Updated Dictionary (without Bob):", students)

students = {

"Alice": 20,

"Bob": 22,

"Charlie": 19,

"David": 21

}

key\_to\_check = "Alice"

**if** key\_to\_check **in** students:

        print(f"{key\_to\_check} exists in the dictionary.")

**else**:

        print(f"{key\_to\_check} does not exist in the dictionary.")

*# Open a file for writing*

**with** open("example.txt", "w") **as** file:

        file.write("Hello, World!\n")

        file.write("This is a sample text file.")

*# Open a file for reading*

**with** open("example.txt", "r") **as** file:

        content = file.read()

print(content)

*# 52.Copy the contents of one file to another.  
# 53.Read and print the contents of a CSV file (assuming a CSV file with  
# comma-separated values).  
# 54.Calculate the square root of a number.  
# 55.Generate a random number.  
# 56.Find the power of a number.  
# 57.Calculate the area of a circle.  
# 58.Calculate the area of a triangle.  
# 59.Print a pyramid pattern.  
# Open a file for appending*

**with** open("example.txt", "a") **as** file:

    file.write("\nAppending some more text to the file.")

*# Open the source and destination files*

**with** open("example.txt", "r") **as** source\_file, open("destination.txt", "w") **as** destination\_file:

    content = source\_file.read()

    destination\_file.write(content)

**import** csv

*# Open a CSV file for reading*

*##with open("data.csv", newline= '') as csvfile:  
##    csvreader = csv.reader(csvfile)  
##    for row in csvreader:  
##        print(row)*

**import** math  
number = 16 *# Replace with the number you want to find the square root of*

square\_root = math.sqrt(number)

print(f"The square root of {number} is {square\_root:.2f}")

**import** random  
random\_number = random.randint(1, 100) *# Generates a random integer between 1 and 100*

print("Random number:", random\_number)

number = 2

exponent = 3

result = number\*\* exponent

print(f"{number} raised to the power of {exponent} is {result}")

**import** math  
radius = 5 *# Replace with the desired radius*

area = math.pi\* (radius\*\* 2)

print(f"The area of the circle with a radius of {radius} is {area:.2f}")

base = 6

height = 8

area = 0.5 \* base \* height

print(f"The area of the triangle is {area}")

rows = 5 *# Number of rows in the pyramid*

**for** i **in** range(1, rows + 1):

        print(" " \* (rows - i) + "\*" \* (2 \* i - 1))

*# 60.Print a rectangle pattern.  
# 61.Print a right-angled triangle pattern.  
# 62.Print a diamond pattern.  
# 63.Print an inverted pyramid pattern.  
# 64.Handle a division by zero exception.  
# 65.Handle a file not found exception.  
# 66.Display the current date and time.  
# 67.Find the difference between two dates.  
# 68.Create a countdown timer.*

width = 6 *# Width of the rectangle*

height = 4 *# Height of the rectangle*

**for** i **in** range(height):

     print("\*" \* width)

rows = 5 *# Number of rows*

**for** i **in** range(1, rows + 1):

    print("\*" \* i)

rows = 5 *# Number of rows*

**for** i **in** range(1, rows + 1):

    print(" " \* (rows - i) + "\*" \* (2 \* i - 1))

**for** i **in** range(rows - 1, 0, -1):

    print(" " \* (rows - i) + "\*" \* (2 \* i - 1))

rows = 5 *# Number of rows in the inverted pyramid*

**for** i **in** range(rows, 0, -1):

    print(" " \* (rows - i) + "\*" \* (2 \* i - 1))

**try**:

        numerator = 10

        denominator = 0

        result = numerator / denominator

**except** ZeroDivisionError:

        print("Division by zero is not allowed.")

**try**:

**with** open("example.txt", "r") **as** file:

            content = file.read()

**except** FileNotFoundError:

            print("File not found. Please check the file path or name.")

**import** datetime  
current\_datetime = datetime.datetime.now()

print("Current Date and Time:", current\_datetime)

**from** datetime **import** datetime

*# Define two dates*

date1 = datetime(2023, 10, 31)

date2 = datetime(2002, 9, 24)

*# Calculate the difference*

date\_difference = date1 - date2

print("Difference between the two dates:", date\_difference)

**import** datetime

**import** time

*# Set the target date and time for the countdown*

target\_datetime = datetime.datetime(2023, 12, 31, 23, 59, 59)

*##while True:  
##    current\_datetime = datetime.datetime.now()  
##    time\_remaining = target\_datetime - current\_datetime  
##    if time\_remaining.total\_seconds() <= 0:  
##        print("Countdown timer expired!")  
##        break  
##    else:  
##        print("Time remaining: ", time\_remaining)  
##    time.sleep(1)*