Name : ISHIKA SHARMA

Email id : ishikasharma.aug2001@gmail.com (mailto:ishikasharma.aug2001@gmail.com)

College : IIT Dharwad, Karnataka

MINI Project_1

Q1 : Write code to

```
1.1 Construct a list of name ‘COLORS’ having items red,green,blue,yello
w,white,black
1.2 Print List
1.3 Print First 3 item COLORS of list as list
1.4 Print last 3 item of list COLORS as list using minus indexing
1.5 append ‘Cyan’ to list COLORS and print list
1.6 Delete item index 2,3,4 from COLORS and print list and length of lis
t COLORS
1.7 check ‘Orange in the list of not.
```

In [1]:

```python
# 1.1
COLORS = ['red', 'green', 'blue', 'yellow', 'white', 'black']
```

In [2]:

```python
# 1.2
print(COLORS)
```

```
['red', 'green', 'blue', 'yellow', 'white', 'black']
```

In [3]:

```python
# 1.3
COLORS[ : 3]
```

Out[3]:

```
['red', 'green', 'blue']
```

In [4]:

```python
# 1.4
COLORS[-3 : ]
```

Out[4]:

```
['yellow', 'white', 'black']
```

In [5]:

```python
# 1.5
COLORS.append('Cyan')
print(COLORS)
```

```
['red', 'green', 'blue', 'yellow', 'white', 'black', 'Cyan']
```

In [6]:

```
# 1.6
del COLORS[2 : 5]
print(COLORS)
print(len(COLORS))
```

```
['red', 'green', 'black', 'Cyan']
4
```

In [7]:

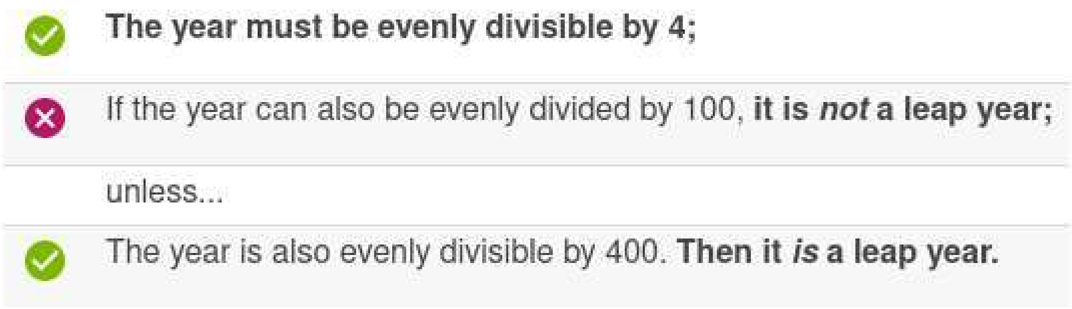
```
# 1.7
if 'Orange' in COLORS:
    print("Orange in list!")
else:
    print("Orange not found!")
```

```
Orange not found!
```

In [ ]:

Q2 : Write a code to find given year is a leap year or not using if else condition

Note:



✓ **The year must be evenly divisible by 4;**

✗ If the year can also be evenly divided by 100, **it is *not* a leap year;**

unless...

✓ The year is also evenly divisible by 400. **Then it *is* a leap year.**

In [8]:

```
# Python program to check if year is a leap year or not

year = int(input('Type Year: '))

n = 0
if year % 4 == 0:
    if year % 100 == 0:
        if year % 400 == 0:
            pass
        else:
            n = 1
            print('Not a Leap year!')
else:
    n = 1
    print('Not a leap year!')

if n == 0:
    print("It is a Leap Year!")
```

```
Type Year: 2986
Not a leap year!
```

```
In [ ]:
```

Q3 : Construct a dictionary having keys and value as following

- name: VIJAY
- income : 45K i.e. 45000 (Integer)
- child : Hema, Neha, Ranbeer (List)

```
Print all keys
Print all children name and no. of childern
Update income by multiplying with factor 1.5
Add a child name ‘ Ravi‘
```

```
In [9]: dict1 = {'name' : 'VIJAY',
                 'income': 45000,
                 'child' : ['Hema', 'Neha', 'Ranbeer']}

        dict1.keys()
```

```
Out[9]: dict_keys(['name', 'income', 'child'])
```

```
In [10]: count = 0
         for name in dict1['child']:
             print(name)
             count += 1
         print("Number of children: ", count)
```

```
Hema
Neha
Ranbeer
Number of children:  3
```

```
In [11]: dict1['income'] = 45000 * 5
         print(dict1)
```

```
{'name': 'VIJAY', 'income': 225000, 'child': ['Hema', 'Neha', 'Ranbeer']}
```

```
In [12]: dict1['child'].append('Ravi')
         dict1
```

```
Out[12]: {'name': 'VIJAY',
          'income': 225000,
          'child': ['Hema', 'Neha', 'Ranbeer', 'Ravi']}
```

```
In [ ]:
```

Q4 : Given a list, iterate it, and display numbers divisible by five, and if you find a number greater than 150, stop the loop iteration.

In [13]:

```
list1 = [12, 15, 32, 42, 55, 75, 122, 132, 150, 180, 200]
```

In [14]:

```
print("Numbers divisible by 5 in list 1 are: \n")
for num in list1:
    if num % 5 == 0:
        if num > 150:
            break
        print(num)
```

```
Numbers divisible by 5 in list 1 are:

15
55
75
150
```

In [ ]:

Q5 : Write a function in Python to count and display the total number of words in a text file.

Using split function and make your own txt file

In [15]:

```
import os
cwd = os.getcwd()
file = 'text_file.txt'
with open(os.path.join(cwd, file), 'w') as fp:
    pass

t1 = "Loss function:"
t2 = "\n\nWe need a way to evaluate how well our model is performing."
t3 = "We compare the model's predictions with the actual targets as follows:"
t4 = "\n\nCalculate the difference between the two matrices (preds and targets).'
t5 = "Square all elements of the difference matrix to remove negative values."
t6 = "Calculate the average of the elements in the resulting matrix."
t7 = "\n\nThe result is a single number, known as the mean squared error (MSE)."

str1 = [t1, t2, t3, t4, t5, t6, t7]

f = open("text_file.txt", "a")
f.writelines(str1)
f.close()

f = open("text_file.txt", "r")

y = []
for x in f:
    y += x.split()
print(len(y))

f.close()
```

```
66
```

Q6: Write a function display_words() in python to read lines from a text file "story.txt", and display those words, which are less than 4 characters.

In [16]:

```
def display_words():
    import re
    total_words = []
    char4 = []

    f = open("text_file.txt", "r")

    for line in f:
        # way of splitting - used when we don’t know the
        # exact characters we want to split upon
        y = re.findall(r"[\w']+", line)

        total_words.extend(y)

    print("Words, which are less than 4 characters are: \n")

    for word in total_words:
        if len(word) < 4:
            char4.append(word)

    print(char4)

    f.close()
```

In [17]:

```
file2 = 'story.txt'
with open(os.path.join(cwd, file2), 'w') as fp:
    pass

t1 = "Loss function:"
t2 = "\n\nWe need a way to evaluate how well our model is performing."
t3 = "We compare the model's predictions with the actual targets as follows:"
t4 = "\n\nCalculate the difference between the two matrices (preds and targets).'
t5 = "Square all elements of the difference matrix to remove negative values."
t6 = "Calculate the average of the elements in the resulting matrix."
t7 = "\n\nThe result is a single number, known as the mean squared error (MSE)."

str1 = [t1, t2, t3, t4, t5, t6, t7]

f2 = open("story.txt", "a")
f2.writelines(str1)
f2.close()


display_words()
```

```
Words, which are less than 4 characters are: 

['We', 'a', 'way', 'to', 'how', 'our', 'is', 'We', 'the', 'the', 'as', 'the', 
'the', 'two', 'and', 'all', 'of', 'the', 'to', 'the', 'of', 'the', 'in', 'the', 
'The', 'is', 'a', 'as', 'the', 'MSE']
```

In [ ]:

Q7: Use keywords Try, Except, Else, Finally

- Try: This block will test the excepted error to occur
- Except: Here you can handle the error
- Else: If there is no exception then this block will be executed
- Finally: Finally block always gets executed either exception is generated or not

https://www.geeksforgeeks.org/try-except-else-and-finally-in-python/ (https://www.geeksforgeeks.org/try-except-else-and-finally-in-python/)

To make a program use these keywords

In [18]:

```
try:
    val = int(input("Please enter an integer: "))

except ValueError:
    print("Looks like you did not enter an integer!")

    try:
        val = int(input("Try again-Please enter an integer: "))
    except ValueError:
        print("Go and read about an integer, and then Try again!")
    else:
        print("GOOD! This time you entered an integer! It is = ", val)

else:
    print("Your integer is: ", val)

finally:
    print("Finally, I executed!")
```

```
Please enter an integer: a
Looks like you did not enter an integer!
Try again-Please enter an integer: 4
GOOD! This time you entered an integer! It is =  4
Finally, I executed!
```

In [ ]:

Q8: Use math library

https://www.w3schools.com/python/module_math.asp (https://www.w3schools.com/python/module_math.asp)

Type code which uses at least 10 math library build function. Try to use constant also.

In [19]:

```python
import math

# Returns value of PI
print('PI = ', math.pi)

# Returns cosine of a number
print('\ncos(PI) = ', math.cos(math.pi))

# Returns e raised to the power of x
print('\ne^1 = ', math.exp(1))

# Returns the factorial of an number
print('\n7! = ', math.factorial(7))

# Returns the greatest common divisor of two integers
print('\nGCD(504, 4536) = ', math.gcd(504, 4536))

# Returns the base-2 logarithm of x
print('\nlog2(32) = ', math.log2(32))

# Returns the natural logarithm of a number
print('\nln(10) = ', math.log(10))

# Returns the no. of ways to choose k items from n items with order and
# without repetition
print('\nPermutation = ', math.perm(6, 4))

tuple1 = (19, 28, 34, 53)
# Returns the sum of all items in any iterable (tuples, arrays, lists, etc.)
print('\n\nSum = ', math.fsum(tuple1))

# Returns the product of all the elements in an iterable
print('\nProduct = ', math.prod(tuple1))
```

```
PI =  3.141592653589793

cos(PI) =  -1.0

e^1 =  2.718281828459045

7! =  5040

GCD(504, 4536) =  504

log2(32) =  5.0

ln(10) =  2.302585092994046

Permutation =  360


Sum =  134.0

Product =  958664
```

In [ ]:

```

```

Q9: Use io library:

https://www.geeksforgeeks.org/stringio-module-in-python/ (https://www.geeksforgeeks.org/stringio-module-in-python/)

Use four functions of the io library in your code comment its explanation also.

In [20]:

```python
from io import StringIO
string ='Hello and welcome to GeeksForGeeks.'

# Using the StringIO method to
# set as file object.
file = StringIO(string)

# This will returns whether the file
# is interactive or not.
print("Is the file stream interactive?", file.isatty())

# This will returns whether the file is
# readable or not.
print("Is the file stream readable?", file.readable())

# This will returns whether the file supports
# writing or not.
print("Is the file stream writable?", file.writable())

# This will returns whether the file is
# seekable or not.
print("Is the file stream seekable?", file.seekable())

# This will returns whether the file is
# closed or not.
print("Is the file closed?", file.closed)
```

```
Is the file stream interactive? False
Is the file stream readable? True
Is the file stream writable? True
Is the file stream seekable? True
Is the file closed? False
```

In [ ]:

```

```

Q10: Use NumPy library:

Use 10 functions of the NumPy library in your code comment its explanation also.

In [21]:

```
import numpy as np

A = [[[1, 2, 3],
      [3, 4, 5]],
     [[5, 6, 7],
      [7, 8, 9]]]

# Convert into array
A = np.array(A)
print('A = ', A)

# Gives the dimension of array
print('\nDimensions of A = ', A.ndim)

# Gives shape of array
print('\nShape of A = ', A.shape)

# Gives the size of array
print('\nSize of A = ', A.size)

# Returns an array containing the same data with a new shape
newA = A.reshape(3, 2, 2)
print('\nnewA = ', newA)
```

```
A =  [[[1 2 3]
  [3 4 5]]

 [[5 6 7]
  [7 8 9]]]

Dimensions of A =  3

Shape of A =  (2, 2, 3)

Size of A =  12

newA =  [[[1 2]
  [3 3]]

 [[4 5]
  [5 6]]

 [[7 7]
  [8 9]]]
```

In [22]:

```
# Return evenly spaced values within a given interval
B = np.arange(0, 12, 2)
print('\nB = ', B)

# In (0, 10) i want 50 numbers equally spaced.
print('\nC = ', np.linspace(0, 10, 50))

# Generate arrays of zeros or ones
print('\nD = ', np.ones(9))

print('\nE = ', np.zeros(9))

# Generate identity matrix
print('\nF = ', np.eye(3))
```

```

B =  [ 0  2  4  6  8 10]

C =  [ 0.          0.20408163  0.40816327  0.6122449   0.81632653  1.02040816
  1.2244898   1.42857143  1.63265306  1.83673469  2.04081633  2.24489796
  2.44897959  2.65306122  2.85714286  3.06122449  3.26530612  3.46938776
  3.67346939  3.87755102  4.08163265  4.28571429  4.48979592  4.69387755
  4.89795918  5.10204082  5.30612245  5.51020408  5.71428571  5.91836735
  6.12244898  6.32653061  6.53061224  6.73469388  6.93877551  7.14285714
  7.34693878  7.55102041  7.75510204  7.95918367  8.16326531  8.36734694
  8.57142857  8.7755102   8.97959184  9.18367347  9.3877551   9.59183673
  9.79591837 10.        ]

D =  [1. 1. 1. 1. 1. 1. 1. 1. 1.]

E =  [0. 0. 0. 0. 0. 0. 0. 0. 0.]

F =  [[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

In [ ]:

```
```

Q11: Use Errors and Exceptions:

Explain any 5 Errors and Exceptions in Python and use them in code

https://www.geeksforgeeks.org/errors-and-exceptions-in-python/ (https://www.geeksforgeeks.org/errors-and-exceptions-in-python/)

In [23]:

```
# ZeroDivisionError (logical error)
# Division by zero not defined

a = 3
b = 0
if a / b:
    print(b)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-23-bcbe47a3f235> in <module>
      4 a = 3
      5 b = 0
----> 6 if a / b:
      7     print(b)

ZeroDivisionError: division by zero
```

In [24]:

```
# IndentationError
# To specify a block of code,
# a tab space (4 spaces) is needed.

if a < b:
print('Yes!')

else:
print('No!')
```

```
  File "<ipython-input-24-d830c36f0b69>", line 6
    print('Yes!')
    ^
IndentationError: expected an indented block
```

In [25]:

```
# NameError
# Variable name is not defined

print(c)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-25-e9367ceaf8ae> in <module>
      2 # Variable name is not defined
      3 
----> 4 print(c)

NameError: name 'c' is not defined
```

```
In [26]: # IndexError
         # Index we asked for goes out of range.

         list1 = [1, 2, 2, 6]
         list1[4]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-26-368119ed68e4> in <module>
      3 
      4 list1 = [1, 2, 2, 6]
----> 5 list1[4]

IndexError: list index out of range
```

```
In [27]: # AttributeError
         # Attribute assignment fails

         t = (9, 8, 7, 6)
         t.append(2)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-27-46f0cde1bdcc> in <module>
      3 
      4 t = (9, 8, 7, 6)
----> 5 t.append(2)

AttributeError: 'tuple' object has no attribute 'append'
```

```
In [ ]: 
```

Q12: Write a Python class named Circle constructed by a radius and two methods that will compute the area and the perimeter of a circle.

In [28]:

```
class Circle():
    pi = 3.14

    def __init__(self, radius):
        self.radius = radius

    def area(self):
        """Calculte Area of a circle"""
        return self.pi * self.radius ** 2

    def perimeter(self):
        """Calculate perimeter of the circle"""
        return 2 * self.pi * self.radius

NewCircle = Circle(8)
print("Area is: ", NewCircle.area())
print("Perimeter is: ", NewCircle.perimeter())
```

```
Area is:  200.96
Perimeter is:  50.24
```

In [ ]: