

Practical 1

- A) Create a program that asks the user to enter their name and their age. Print out message addressed to them that tells them the year they will turn 100 years old.

Code:

```
name = input("Enter Name:")
age = int(input("Enter Age:"))
age = 2023 - age
year = str(age + 100)
print(name + " will turn 100 years
old in the year " + year)
```

3) Enter number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.

code:

```
num = int(input("Enter a Number:"))
if (num%2 == 0):
    num = str(num)
    print(num + " is Even.")
else:
    num = str(num)
    print(num + " is Odd.")
```

c) Write a program to generate the Fibonacci series.

code:

```
nTerms = int(input("Enter number  
of terms:"))  
n1, n2 = 0, 1  
count = 0  
if nTerms <= 0:  
    print("Please enter a positive  
integer.")  
else:  
    print("Fibonacci series:")  
    while count < nTerms:  
        print(n1)  
        nth = n1 + n2  
        n1 = n2  
        n2 = nth  
        count += 1
```

Q) Write a function that reverses the user defined value.

Code:

```
def reverse(x):  
    return x[::-1]  
myvalue = reverse(input("Enter a  
value:"))  
print(myvalue)
```

Write a function to check the input value is Armstrong and also write function for Palindrome.

Code:

```
def armstrong(num):
    return num
num=int(input("Enter a number:"))
sum=0
temp=num
while temp>0:
    digit=temp%10
    sum+=digit ** 3
    temp//=10
if num==sum:
    print(num," is an Armstrong
          number.")
else:
    print(num," is not an Armstrong
          number.")
```

```
def palindrome(num):
    return num
num=int(input("Enter a number:"))
temp=num
rev=0
while(temp!=0):
    rem=temp%10
    rev=rev*10+rem
    temp//=10
if num==rev:
    print(num," is Palindrome.")
else:
    print(num," is not Palindrome.")
```

E) Write a recursive function to print Factorial for a given number.

Code:

```
def fact(xc):
    if xc == 1:
        return 1
    else:
        return (xc * fact(xc - 1))
num = int(input("Enter a number:"))
print("The factorial of num is",
      fact(num))
```

~~150223~~

Practical 2

A) Write a function that takes a character and returns True if it is a vowel False otherwise.

Code:

```
def isVowel(ch):
    if(ch == 'a' or ch == 'A' or ch == 'e' or
       ch == 'E' or ch == 'i' or ch == 'I' or
       ch == 'u' or ch == 'U'):
        print(ch, " is a vowel.")
    else:
        print(ch, " is not a vowel.")
ch = input("Enter any char(A-Z/a-z)
only:")
isVowel(ch)
```

Define a function that computes the length of given list or string.

code:

```
def findLen(str):
    counter=0
    for i in str:
        counter+=1
    return counter
str=input("Enter a string:")
print(findLen(str))
```

Define a procedure histogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following:

* * * *
* * * * * * * *
* * * * * *

code:

```
def histogram(items):
    for n in items:
        output = ""
        times = n
        while(times > 0):
            output += "*"
            times = times - 1
        print(output)
histogram([4, 9, 7])
```

DS
25/07/23

Practical 3

A pangram is a sentence that contains all letters of English alphabet atleast once. Your task here is to write a function to check sentence to see if it is pangram or not.

code:

```
def isPangram(str,alphabet):
    flag = True
    for char in alphabet:
        if char not in str.lower():
            flag = False
    if (flag == True):
        print("It is a Pangram sentence")
    else:
        print("It is not a Pangram sentence")
```

~~str = "The brown quick fox jumps over
the lazy dog."}~~

~~alphabet = "abcdefghijklmnopqrstuvwxyz"~~

~~isPangram(str,alphabet)~~

$q = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$

Write a program to list elements less than 5.

Code:

$q = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]$

for i in q:
 if i < 5:
 print(i)

25/02/23

Practical 4

WAP that takes two lists and returns True if they have atleast one common member.

Code:

```
def common(list1, list2):
    for x in list1:
        for y in list2:
            if x == y:
                return True
    return False
```

~~list1 = [4, 5, 6, 7, 32, 89, 39, 44, 99, 10, 11, 2]~~

~~list2 = [98, 35, 6, 2, 8, 4, 213, 68, 40, 45]~~

~~list3 = [45, 23, 89, 33, 12, 89]~~

~~list4 = [1, 5, 8, 9, 3]~~

print("List1-", list1)

print("List2-", list2)

print("Result for List1 and List2:",
 common(list1, list2))

print("\nList3-", list3)

print("List4-", list4)

print("Result for List3 and List4:",
 common(list3, list4))

Q) WAP to print specified list after removing 0th, 2nd, 4th and 5th elements.

Code:

```
def removeElement(list):
    specificElement = [0, 2, 4, 5]
    filtered_list = [value for index,
                    value in enumerate(list) if index
                    not in specificElement]
    return filtered_list
list = [2, 5, 8, 45, 67, 89, 443, 1223, 34]
print(list)
print("Filtered List-", removeElement
      (list))
```

Q) WAP to clone or copy a list.

Code:

```
def clone(list):
    clone = list.copy()
    return clone
myList = [3, 6, 4, 2, 7, 8, 5, 3, 56, 34, 78, 34]
print(myList)
print("Clone of myList - ")
clone_list = clone(myList)
print(clone_list)
```

~~AB
25/07/23~~

Practical 15

Write a Python script to sort a dictionary by value. (ascending and descending).

Code:

```
import operator
d = {'python': 90, 'CPP': 100, 'java': 80,
      'php': 60}
print("Original dictionary = ", d)
asc = dict(sorted(d.items(),
                  key=operator.itemgetter(1)))
print("Ascending order = ", asc)
dsc = dict(sorted(d.items(),
                  key=operator.itemgetter(1),
                  reverse=True))
print("Descending order = ", dsc)
```

Write a Python script to concatenate following dictionaries to create a new one

Code:

```
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}
print("Dict1 - ", dict1)
print("Dict2 - ", dict2)
concatenated_dict = {**dict1, **dict2}
print("concatenated Dict - ",
      concatenated_dict)
```

WAP to sum all items in a dictionary.

Code:

```
dict1 = {'a': 10, 'b': 20, 'c': 30}
total = sum(dict1.values())
print("Dict1 - ", dict1)
print("sum of all values: ", total)
```

Practical 6

A) WAP to read an entire text file.

Code:

```
file = open('sample', 'r')
text = file.read()
print(text)
file.close()
```

B) WAP to append text to a file and display the text.

Code:

```
file = open('syit1.txt', 'a')
text = file.read()
print(text)
file.close()
```

C) WAP to read last n lines of a file.

Code:

```
f = open('sample', 'r')
t = f.readlines()
print(t[-1])
f.close()
```

Practical 7

- 4) Design a class that store information of student and display the same.

Code:

```
class student:
    def info(self, name, city, number):
        print("Name:", name, "\nCity:", city,
              "\nNumber:", number)
obj = student()
obj.info("Heramb", "Mumbai", 9845121609)
```

- 3) Implement concept of inheritance using python.

Code:

```
class parent1:
    def f1(self):
        print("Parent1 Class")
class parent2:
    def f2(self):
        print("Parent2 Class")
class child(parent1, parent2):
    def f3(self):
        print("Child Class")
```

t = child()

t.f1()

t.f2()

t.f3()

~~AS
29.08.2022~~

Practical 8

WAP to implement exception handling in Python.

Code -

```
try:  
    n=int(input("Enter a number:"))  
    re=100/n  
except (ValueError,ZeroDivisionError):  
    print("Wrong input!")  
else:  
    print("Result:",re)
```