## 1_Read_txt.py

```python
# Program - 1
f = open("p1.txt", "r")
line = " "
while line :
    line = f.readline()
    for word in line.split() :
        print(word, end = "#")
    print()
f.close()
```

**2_Seperate_char_txt.py**

```python
# Program - 2
ch = " "
v_num = 0
c_num = 0
l_num = 0
u_num = 0
v_list = ["a", "A","e", "E","i", "I","o", "O","u", "U"]

with open("Text_Files\p1.txt", "r") as f :
    while ch:
        ch = f.read(1)
        if ch.islower():
            l_num += 1
        if ch.isupper():
            u_num +=1
        if ch in v_list:
            v_num +=1
        elif ch not in v_list and ch.isalpha():
            c_num +=1

print(f"The number of vowels : {v_num}")
print(f"The number of consonants : {c_num}")
print(f"The number of lowercase letters : {l_num}")
print(f"The number of uppercase letters : {u_num}")

```

### 3_Writelines_to_another_file.py

```python
# Program - 3
f = open("Text_Files\p1.txt", "r")
line = f.readline()
l1 = []
l2 = []
for i in line :
    if "a" in i :
        l1.append(i)
    else:
        l2.append(i)
f.close()

f =  open(r"Text_Files\p2.txt", "w")
f.writelines(l1)
f.close()

f =  open(r"Text_Files\ p2-3.txt", "w")
f.writelines(l2)
f.close()
```

# 4_Binary_file_operation.py

```python
# Program - 4
import pickle

f = open("Text_Files\Students.dat", "wb+")
n = int(input("No. of records : "))
d = {}
for i in range(1, n+1):
    print(f"({i})")
    name = input("Enter the name of the student :")
    rollno = int(input("Enter the roll No. :"))
    d[rollno]=name
pickle.dump(d,f)
f.close()
f = open("Text_Files\Students.dat", "rb+")
a = int(input("Enter the roll no. to search by : "))
b = pickle.load(f)
if a in b.keys():
    print(f"The name of the student with the given roll\
        no. {a} is {b[a]}")
else:
    print("No record matches with the given roll no.")
f.close()
```

**5_Dice.py**

```
# Program - 5
import random, time
print("Random Number Generator")

def generate_num():
    time.sleep(2)
    a=random.randint (1,6)
    print (a)

C= True
while C:
    print ("Generating...")
    generate_num()
    b=input ("Do you want to roll the dice once more (y/n)?")
    if b == 'y':
        continue
    else:
        C=False

```

### 6_Stack.py

```python
# Program - 6
a=[]
while True:
    print ("\nPush -> 1")
    print ("Pop -> 2")
    print ("Display-Stack -> 3")
    print ("Exit -> 4\n")
    b= int(input("Enter your choice: "))

    if b==1:
        c=input("Enter any element:")
        if "," in c :
            c = c.split(",")
            if " "  in c :
                c.remove(" ")
            print(c)
        a.append (c)

    elif b==2:
        if a== []:
            print (" Underflow! Stack is empty...")
        else:
            print("Popped element is", a.pop())

    elif b==3:
        if a== []:
            print ("Stack is empty...")
        else:
            d= len(a)
            for i in range (d-1,-1,-1):
                print (f"\n{a[i]}")

    elif b==4:
        print ("End")
        break
    else:
        print("Invalid choice!")

```

**7_CSV.py**

```python
# Program - 7
import csv
product_data= [
                ["PID", "PNAME", "COST", "QUANTITY"],
                ["P1", "BRUSH", 50, 200],
                ["P2" "TOOTHPASTE", 120, 150],
                ["P3" "SOAP", 40,300],
                ["P4", "SHEETS", 100, 500],
                ["P5", "PEN", 10, 250]
            ]

def write ():
    a= open("PRODUCT. csv", "w", newline="")
    c= csv.writer (a)
    c. writerows (product_data)

def read ():
    a= open("PRODUCT· CSV", "r")
    c= csv.reader (a)
    for i in c:
        print (i)
write()
read ()
```

# 8_Exeption_handling_txt.py

```
# Program - 8
error_name = ""
try :
    file_loc = input("Enter your file location & name of the file : ")
    f1 = open(file_loc, "r")
    content = f1.read()
    print(content)

except FileNotFoundError as err :
    error_name = "FileNotFoundError"
    print("File Not Found")

except IOError as err:
    error_name = "IOError"
    print("Error occured while reading the file.")

except Exception as e :
    print(f"Caught {e}")

finally :
    if error_name != "FileNotFoundError":
        f1.close()
        print("File Closed")

```

# 9_Remaining_Days_calc.py

```python
# Program - 9
import datetime
a=0
b= datetime.datetime. now()
print("Today's date:", b.day)
if b. month == 2:
    a=28
elif b. month in (1,3,5,7,8, 10, 12):
    a=31
else:
    a=30
print("Total remaining days in the current month are:", a-b.day, "days")
```

**10_CSV_Search.py**

```python
# Program - 10
import csv
with open("user-info.csv", "w") as obj:
    fileobj = csv.writer (obj)
    fileobj.writerow(["User_ID", "Password"])
    while True:
        user_id=input("Enter ID: ")
        password= input("Enter password:")
        record=[user_id, password]
        fileobj. writerow (record)
        x=input ("Press Y/y to continue or N/n to terminate the program:\n")
        if x in "Nn":
            break
        elif x in "Yy":
            continue
with open("user-info.csv", "r") as obj1:
    fileobj1 = csv. reader (obj1)
    given=input ("Enter the user-id to be Searched\n")
    for i in fileobj1:
        next(fileobj1)
        if i[0] == given:
            print ("Password is", i[1])
            break
        else:
            print ("No record matches with thegiven user-id")
```

## 11_Len_of_txt_file.py

```python
# Program - 11
myfile= open("Text_Files\p1.txt", "r")
s= myfile.readlines ()
linecount = len(s)
size=0
for i in s:
    a= len(i)
    size+=a
print (" Size of the file is", size)
print (" No. of lines in the file is", linecount)
```

### 12_Armstrong.py

```python
# Program - 12
def arm ():
    n=int(input("Enter the number:"))
    s=n
    b= len(str (n))
    Sum=0
    while n!=0:
        Sum=Sum+(8** b)
        n = n //10
        if s==Sum:
            print (" The given number", s, "isarmstrong number")
        else:
            print (" The given number", s, " is not armstrong number")
arm()
```

### 13_Factorial.py

```python
# Program - 13
def factorial (n):
    if n<0:
        fact=-1
        while n<-1:
            fact *=n
            n-=-1
        return fact
    elif n ==0 or n == 1:
        return 1
    else:
        fact = 1
        while n>1:
            fact *= n
            n-=1
        return fact
num = int (input("Enter the number to find factorial!" ))

print(" Factorial of ", num, "is", factorial (num))
```

## 14_Date_Conversion.py

```python
# Program - 14
import string # importing string module

month_dict = {
                '01': 'January', '02': 'February', '03': 'March',
                '04': 'April', '05': 'May', '06': 'June',
                '07': 'July', '08': 'August', '09': 'September',
                '10': 'October', '11': 'November', '12': 'December'
            }  # initializing dictionary which contains the months
def validation ():
    try:
        while True :
            global date

            date = str(input("Enter the date in the format of <MMDDYYYY> : "))

            if len(date)!= 8:
                print("!! Invalid Format !!")
                continue

            for i in string.ascii_letters + string.whitespace + string.punctuation :
                if i in date :
                    print("!! Characters Not Allowed !!")
                    break

            else :
                month = date[0:2]
                day = date[2 : 4]
                if month not in month_dict:
                    print("!! Invalid Month Entry !!")
                    continue

                if int(day) > 31 :
                    print("!! Date greater than 31 !!")
                    continue

                if month == "02" and int(day) > 29 :
                    print("!! February only has 29 days !!")
                    continue

                else:
                    month =  month_dict[month]
                    break
    except Exception as e:
        print(f'Caught {type(e)}: e')

    finally :
        return  month
```

```python

def standardize(_month):
    standard_form =f"{ _month}, {date[2 : 4]}, {date [4 : ]}"
    return standard_form

print(f"The Standard Form : {standardize(validation())}")
```

**15_IsFibonacci.py**

```python
# Program - 15
import math

def is_perfect_square(x):
    s = int(math.sqrt(x))
    return s*s == x

def is_fibonacci (n):
    return is_perfect_square(5*n*n+4) or is_perfect_square(5*n*n-4)

num = float(input("Enter a number : "))
if is_fibonacci(num) == True :
    print(f"{num} is a fibonacci number.")
else :
    print(f"{num} is not a fibonacci number.")
```