Functions in Python

Q) Write a program for adding two numbers using a function.

```
def sum ( x ,  y):
   print("SUM = ", x + y)
print("ENTER FIRST VALUE")
a = int(input())
print("ENTER SECOND VALUE")
b = int(input())
sum (a,b)
```

Q) Write a program to perform the following operations on any two given integers using functions.

1. Add
2.Subtract
3.Multiply
4.Divide

```
def add ( x ,  y):
   return(x + y)

def sub ( x ,  y):
   return(x - y)

def mul ( x ,  y):
   return(x * y)

def div ( x ,  y):
```

```
    return(x / y)


while (True):
    print("1.ADD")
    print("2.SUB")
    print("3.MUL")
    print("4.DIV")
    print("5.EXIT")
    print("ENTER OPTION")
    option = int(input())
    print("ENTER FIRST VALUE")
    a = int(input())
    print("ENTER SECOND VALUE")
    b = int(input())
if option ==1 :
        c = add (a,b)
        print("SUM = ", c)
    elif option == 2:
        c = sub (a,b)
        print("DIFFERENCE = ", c)
    elif option == 3:
       c = mul (a,b)
       print("PRODUCT= ", c)
    elif option == 4:
        c = div (a,b)
        print("DIVISION= ", c)
```

```
    else:
        print("TERMINATING")
        break;
```

Q) Write a program to swap two numbers

```
def swap(a,b):
    a,b = b,a
    print("AFTER SWAPPING\n")
    print("\na =",a )
    print("\nb =",b )
print("ENTER FIRST NUMBER\n")
a = input()
print("ENTER SECOND NUMBER\n")
b = input()
print("\na =",a )
print("\nb =",b )
swap(a,b)
```


Variable Length Arguments

Example:

```
def info(s,*names):
    print("Information about : ",s)
    for i in names:
        print(i)
```

```
info("food" , "Italian","mexican","turkish")
print("========================")
info("cars","Mercedes","BMW","Audi","Jaguar","Ford")
print("========================")
info("cities","Mumbai","London","Paris","New York","Barcelona","Moscow")
def info(s,*names):
   print("Information about : ",s)
   for i in names:
      print(i)

info("food" , "Italian","mexican","turkish")
print("========================")
info("cars","Mercedes","BMW","Audi","Jaguar","Ford")
print("========================")
info("cities","Mumbai","London","Paris",
"New York","Barcelona","Moscow")
```

Recursion in Python

Q) A program to calculate factorial using Recursion

```
def facto(n):
   if (n == 0 or n ==1):
      return 1
   else:
      return(n*facto(n-1))
print("ENTER  NUMBER\n")
n = int(input())
ans = facto(n);
print("\nANSWER =",ans )
```

Q) Write a program to display the calendar of any given year.

```
import calendar   #importing a module called calendar
print("ENTER YEAR\n")
year = int(input())
print ("The calender of year %d  year   is : " %year)
print (calendar.calendar(year,2,1,6))
```

Q) Write a program to display the Current Date and time.

```
from datetime import datetime  #  imported datetime class from the datetime
                                                    module
now = datetime.now()  # here datetime is an object
print ("Now: ", now)
print ("Today's date: ", now.strftime('%Y-%m-%d'))
print ("Today's date: ", now.strftime('%d-%m-%Y'))
print ("year:", now.year)
```

```
print ("month:", now.month)

print ("day:", now.day)

print ("hour:", now.hour)

print ("minute:", now.minute)

print ("second:", now.second)
```

Q) Write a program to demonstrate conversion of time across various timezones.

```
from datetime import datetime

from pytz import timezone

# Current time

print (datetime.now())

#Current time to UTC

now_utc = datetime.now(timezone('UTC'))

print (now_utc.strftime("%Y-%m-%d %H:%M:%S %Z%z"))

#Convert to US/Eastern time zone

now_useastern = now_utc.astimezone(timezone('US/Eastern'))

print (now_useastern.strftime("%Y-%m-%d %H:%M:%S %Z%z"))

#Convert to Asia/Kolkata time zone

now_india = now_useastern.astimezone(timezone('Asia/Kolkata'))

print (now_india.strftime("%Y-%m-%d %H:%M:%S %Z%z"))
```

Strings in Python

Example : Program to demonstrate string functions

```
print("ENTER YOUR NAME")
name =  input()
print("My name is %s \n" %name)
print(name[4])
print(name[4:12]) #this is called slicing
print (name[4:12:2]) # this is slicing with striding
print(name.strip('J'))
print(max(name))
print(name.swapcase())
print(list(enumerate(name)))
```

Q) Write a Python program to count the number of characters (character frequency) in a string.

```
def char_frequency(str1):
   dict = {}
   for n in str1:
      keys = dict.keys()
      if n in keys:
         dict[n] += 1
      else:
         dict[n] = 1
   return dict
print("ENTER STRING\n")
```

```
s = raw_input()
print(char_frequency(s))
```

Q) Write a Python program to count the occurrences of each word in a given sentence.

```
def word_count(str):
   counts = dict()
   words = str.split()
   for word in words:
      if word in counts:
         counts[word] += 1
      else:
         counts[word] = 1
   return counts
print("ENTER STRING\n")
s = input()
print(word_count(s))
```

Q) Write a Python program to create a Caesar encryption

Note : In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on. The method is named after Julius Caesar, who used it in his private correspondence.

```
def caesar_encrypt(realText, step):
        outText = []
        cryptText = []

        uppercase = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
```

```
        lowercase = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']


for eachLetter in realText:
                if eachLetter in uppercase:
                        index = uppercase.index(eachLetter)
                        crypting = (index + step) % 26
                        cryptText.append(crypting)
                        newLetter = uppercase[crypting]
                        outText.append(newLetter)
                elif eachLetter in lowercase:
                        index = lowercase.index(eachLetter)
                        crypting = (index + step) % 26
                        cryptText.append(crypting)
                        newLetter = lowercase[crypting]
                        outText.append(newLetter)
        return outText
print("ENTER STRING\n")
s = input()
code = caesar_encrypt(s, 2)
print(code)
```