



[\(https://www.darshan.ac.in/\)](https://www.darshan.ac.in/)

## **Python Programming - 2101CS405**

### **Lab - 6**

**Name : Viral Chauhan**

**Enrollment : 22010101027**

**Roll No. : 184    Batch : A4**

**Tuples, dictionary, set**

**A**

## 01) WAP to sort python dictionary by key or value.

```
In [ ]: myDict = {}
key = ""
value = 0

while(True):
    key = input("Enter a key for dictionary or 'q' to exit : ")
    if(key.lower() == "q"):
        break
    value = input("Enter a value for the key : ")
    myDict[key] = value

myKeys = list(myDict.keys())
print(f"my keys : {myKeys}")
myKeys.sort()
print(f"my Dict : {myDict}")
sortedDict = {i: myDict[i] for i in myKeys}
print(f"sorted Dict by keys : {sortedDict}")

myKeys = list(myDict.keys())
value = list(myDict.values())
helper = list(myDict.values())
helper.sort()

sortedDict2 = {}
for i in helper:
    myindex = value.index(i)

    sortedDict2[myKeys[myindex]] = i
print(f"sorted by value : {sortedDict2}")
```

```
my keys : ['x', 'a', 'z', 'b', 'y']
my Dict : {'x': 3, 'a': 5, 'z': 1, 'b': 4, 'y': 2}
sorted Dict by keys : {'a': 5, 'b': 4, 'x': 3, 'y': 2, 'z': 1}
helper = [1, 2, 3, 4, 5] : value = [3, 5, 1, 4, 2]
sorted by value : {'z': 1, 'y': 2, 'x': 3, 'b': 4, 'a': 5}
```

## 02) WAP to merge two dictionaries given by user.

```
In [ ]: myDict = {}
myDict2 = {}
while(True):
    key = input("Enter a key for dictionary or 'q' to exit : ")
    if(key.lower() == "q"):
        break
    value = input("Enter a value for the key : ")
    myDict[key] = value

print(":::::: second dictionary :::::")
while(True):
    key = input("Enter a key for dictionary or 'q' to exit : ")
    if(key.lower() == "q"):
        break
    value = input("Enter a value for the key : ")
    myDict2[key] = value

print(f"myDict = {myDict} , myDict2 = {myDict2}")
myDict.update(myDict2)
print(f"updated dict = {myDict}")
```

Enter a key for dictionary or 'q' to exit : 1  
Enter a value for the key : a  
Enter a key for dictionary or 'q' to exit : 2  
Enter a value for the key : b  
Enter a key for dictionary or 'q' to exit : 3  
Enter a value for the key : c  
Enter a key for dictionary or 'q' to exit : q  
:::::: second dictionary :::::  
Enter a key for dictionary or 'q' to exit : 4  
Enter a value for the key : d  
Enter a key for dictionary or 'q' to exit : 5  
Enter a value for the key : e  
Enter a key for dictionary or 'q' to exit : 6  
Enter a value for the key : f  
Enter a key for dictionary or 'q' to exit : q  
myDict = {'1': 'a', '2': 'b', '3': 'c'} , myDict2 = {'4': 'd', '5': 'e', '6': 'f'}  
updated dict = {'1': 'a', '2': 'b', '3': 'c', '4': 'd', '5': 'e', '6': 'f'}

## 03) WAP to find tuples that have all elements divisible by K from a list of tuples.

```
In [ ]: mylist = [(1, 2, 3), (4, 5, 6), (7, 8, 9), (3, 6), (9, 12)]
k = int(input("Enter a number"))
res = list(filter(lambda sub:all(ele%k ==0 for ele in sub),mylist))
print(f"res = {res}")
```

Enter a number3  
res = [(3, 6), (9, 12)]

#### 04) WAP to find Tuples with positive elements in List of tuples.

```
In [ ]: mylist = [(1, 2, 3), (4, 5, 6), (7, 8, 9), (0, -1, -2), (-4, -5, -6), (4, -5, 6)]

res = list(filter(lambda sub:all(ele >=0 for ele in sub),mylist))
print(f"res = {res}")

res = [(1, 2, 3), (4, 5, 6), (7, 8, 9)]
```

#### 05) WAP which perform union of two sets.

```
In [ ]: set1 = {1, 2, 3, 4, 5}
set2 = {3, 4, 5, 6, 7}

set1 = set1.union(set2)
print(set1)

{1, 2, 3, 4, 5, 6, 7}
```

## B

#### 01) WAP to convert binary tuple into integer.

```
In [ ]: import functools as ft
mytuple = (1, 0, 1, 0, 1)
mypower = 0
mylist = []
for i in mytuple:
    mylist.append((2**mypower)*i)
    mypower +=1
res = ft.reduce(lambda a,b:a+b,mylist)
print(res)
```

21

## 02) WAP to count frequency in list by dictionary.

```
In [ ]: mylist = [1,2,3,4,5,6,7,8,9,0,0,9,8,7,6,5,4,3,2,1]
freq={}
for i in mylist:
    if(i in freq):
        freq[i] +=1
    else:
        freq[i]=1
print(freq)
```

```
{1: 2, 2: 2, 3: 2, 4: 2, 5: 2, 6: 2, 7: 2, 8: 2, 9: 2, 0: 2}
```

## 03) WAP to remove all the duplicate words from the list using dictionary.

```
In [ ]: mylist = ['red', 'blue', 'yellow', 'violet', 'red', 'blue', 'yellow', 'violet', 'red',
mylist2=[]
unique={}
for i in mylist:
    if(i in unique):
        unique[i] +=1
    else:
        unique[i]=1
print(unique)
mylist2.extend(unique.keys())
print(mylist2)
```

```
{'red': 3, 'blue': 3, 'yellow': 3, 'violet': 3}
['red', 'blue', 'yellow', 'violet']
```

```
In [ ]:
```



[\(https://www.darshan.ac.in/\)](https://www.darshan.ac.in/)

## **Python Programming - 2101CS405**

### **Lab - 7**

**Name : Viral chauhan**

**Enrollment : 22010101027**

**Roll No. : 184    Batch : A4**

## **Functions**

### 01) WAP to count simple interest using function.

```
In [1]: def simpleInterest(amount,rateOfInterest,time):  
        return ((amount*rateOfInterest*time)/100)  
  
p = float(input("Enter principle amount : "))  
r = float(input("Enter rate of interest : "))  
n = float(input("Enter time in year : "))  
print("Your simple interest is ",simpleInterest(p,r,n))
```

```
Enter principle amount : 100  
Enter rate of interest : 2  
Enter time in year : 1  
Your simple interest is  2.0
```

### 02) WAP that defines a function to add first n numbers.

```
In [2]: def sumOfFirstNNumbers(n):  
        return ((n/2)*(n+1))  
n = int(input("Enter a number : "))  
print(f"sum of first {n} number is {sumOfFirstNNumbers(n)}")
```

```
Enter a number : 5  
sum of first 5 number is 15.0
```

### 03) WAP to find maximum number from given two numbers using function.

```
In [4]: def maximum(n1,n2):  
        return n1 if n1>n2 else n2  
n1 = int(input("Enter a first number : "))  
n2 = int(input("Enter a second number : "))  
  
print(f"maximum of two numbers {n1} and {n2} is {maximum(n1,n2)}")
```

```
Enter a first number : 5  
Enter a second number : 4  
maximum of two numbers  5 and 4 is 5
```

#### 04) WAP that defines a function which returns 1 if the number is prime otherwise return 0.

```
In [13]: def isPrime(n):
        for i in range(2,int(n**0.5 + 1 )):
            if(n%i == 0):
                return 0

        return 1
n = int(input("Enter a number : "))
print(isPrime(n))
```

Enter a number : 4  
0

#### 05) Write a function called primes that takes an integer value as an argument and returns a list of all prime numbers up to that number.

```
In [14]: n = int(input("Enter a number : "))
primeList = []
for i in range(1,n):
    if(isPrime(i)):
        primeList.append(i)
print(primeList)
```

Enter a number : 100  
[1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

#### 06) WAP to generate Fibonacci series of N given number using function name fibbo. (e.g. 0 1 1 2 3 5 8...)

```
In [19]: def fibbo(a1,a2,n):
        fibboSeries = []
        fibboSeries.append(a1)
        fibboSeries.append(a2)
        for i in range(a1, n-2 if a1==0 else n-1):
            fibboSeries.append(a1+a2)
            a2 = a1 + a2
            a1 = a2 - a1
        return fibboSeries
n = int(input("Enter a number : "))
print(fibbo(0,1,n))
```

Enter a number : 5  
[0, 1, 1, 2, 3]



## 07) WAP to find the factorial of a given number using recursion.

```
In [21]: def factorial(n):
          if(n<=1):
              return 1
          return (n* factorial(n-1))
n = int(input("Enter a number : "))
print(factorial(n))
```

Enter a number : 5  
120

## 08) WAP to implement simple calculator using lamda function.

```
In [27]: def myCal(n1,n2,ch):
          match ch :
              case "+":
                  return (lambda n1,n2: n1+n2)(n1,n2)
              case "-":
                  return (lambda n1,n2: n1-n2)(n1,n2)
              case "*":
                  return (lambda n1,n2: n1*n2)(n1,n2)
              case "/":
                  return (lambda n1,n2: n1/n2)(n1,n2)
              case "%":
                  return (lambda n1,n2: n1%n2)(n1,n2)
          return "invalid operator"
n1 = int(input("Enter a first number : "))
n2 = int(input("Enter a second number : "))

ch = input("Enter operation you want to perform (+,-,*,/,% for modulo)")
print(myCal(n1,n2,ch))
```

Enter a first number : 5  
Enter a second number : 4  
Enter operation you want to perform (+,-,\*,/,% for modulo)\*  
20

## 09)Write a Python program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically

Sample Items : green-red-yellow-black-white  
Expected Result : black-green-red-white-yellow

```
In [29]: s = "green-red-yellow-black-white"
mystr = s.split("-")
mystr.sort()
s = ("-").join(mystr)
print(s)
```

black-green-red-white-yellow

## 10) Write a python program to implement all function arguments type

Positional arguments

Default argument

Keyword arguments (named arguments)

Arbitrary arguments (variable-length arguments args and kwargs)

```
In [31]: def myFun(n1,n2=1):
        print(n1*n2)
def myFun1(*n):
    print(n)
```

```
In [33]: myFun(n2=5,n1=4) # keyword
myFun(4) # default
myFun(11,2) # positional
myFun1(4,5,6,7,8,9,0) # arbitrary
```

20  
4  
22  
(4, 5, 6, 7, 8, 9, 0)

## 01) WAP to calculate power of a number using recursion.

```
In [34]: def myRecPorwer(base,power):
        if(power <=0):
            return 1
        else:
            return base * myRecPorwer(base,power-1)

b = int(input("Enter a base number : "))
p = int(input("Enter a power number : "))
print(myRecPorwer(b,p))
```

Enter a base number : 2  
Enter a power number : 3  
8

**02) WAP to count digits of a number using recursion.**

In [ ]:

**03) WAP to reverse an integer number using recursion.**

In [ ]:

**04) WAP to convert decimal number into binary using recursion.**

In [ ]:



[\(https://www.darshan.ac.in/\)](https://www.darshan.ac.in/)

## **Python Programming - 2101CS405**

### **Lab - 9**

**Name : Viral Chauhan**

**Enrollment : 22010101027**

**Roll No. : 184 Batch : A4**

## **Exception Handling**

```
In [1]: try:
        a=5
        b="3"
        c=a+b
    except TypeError:
        print("Invalid Datatype")
    else:
        print("else Block Excuted")
    finally:
        print("finally Block Excuted")
```

Invalid Datatype  
finally Block Excuted

## A

### 01) WAP to handle divide by zero exception.

```
In [1]: try:
        a=int(input("Enter a number : "))
        b=0
        c=a/b
    except ZeroDivisionError:
        print("divide by zero not possible")
```

Enter a number : 4  
divide by zero not possible

### 02) Write a Python program that inputs a number and generates an error message if it is not a number.

```
In [4]: try:
        a=int(input("Enter a number : "))

    except ValueError:
        print("value error")
```

Enter a number : d  
value error

### 03) WAP to handle file not found Exception

```
In [8]: try:
        f = open("kishan.txt", 'r')
    except FileNotFoundError:
        print("file not found")
```

file not found

### 04) WAP to handle type Exception.

```
In [10]: try:
        a="parth"
        b= 1
        c = b+ a
    except TypeError:
        print("type error")
```

type error

### 05) WAP to demonstrate valueError and indexError with example.

```
In [23]: def errorFun(a,b='as'):
        try:
            a=int(a)
            b= b[10]
        except ValueError:
            print("Value error")
        except IndexError:
            print("Index error")
```

```
errorFun('a')
errorFun('10', 'sdfg')
```

Value error

Index error

## 06) WAP to demonstrate else and finally block.

```
In [26]: def errorFun(a):
          try:
              b= 5/a
          except ZeroDivisionError:
              print("Zero Division Error",a)
          else:
              print("this is else, only when exception is not occured",a)
          finally:
              print("i am finally, every time ",a)
          errorFun(5)
          errorFun(0)
```

```
this is else, only when exception is not occured 5
i am finally, every time 5
Zero Division Error 0
i am finally, every time 0
```

## 07) Create a short program that prompts the user for a list of grades separated by commas. Split the string into individual grades and use a list comprehension to convert each string to an integer. You should use a try statement to inform the user when the values they entered cannot be converted.

```
In [28]: grades = input("enter comma separated marks : ")
          grades = grades.split(",")
          myList = []
          try :
              myList = [int(i) for i in grades]
          except ValueError:
              print("value can't converted")
```

```
enter comma separated marks : 5,6,p
value can't converted
```

# B

## 01) WAP to Raising User Generated Exception.

```
In [29]: class MyError(Exception):
          def __init__(self, message):
              self.message = message

          a = int(input("Enter a positive number"))
          if(a<0):
              raise MyError("number can not be negative")
          else:
              print("positive number",a)
```

Enter a positive number-1

```
-----
MyError                                Traceback (most recent call last)
Cell In[29], line 8
      6 a = int(input("Enter a positive number"))
      7 if(a<0):
----> 8     raise MyError("number can not be negative")
      9 else:
     10     print("positive number",a)

MyError: number can not be negative
```

## 02) WAP to raise your custom Exception.

```
In [37]: class MyError(Exception):
          def __init__(self):
              self.message = "number cannot be negative"

          a = int(input("Enter a odd number"))
          try:
              if(a%2 == 0):
                  raise MyError
              else:
                  print("positive number",a)
          except MyError as e:
              print(e.message)
```

Enter a odd number4  
number cannot be negative



In [ ]:



[\(https://www.darshan.ac.in/\)](https://www.darshan.ac.in/)

## **Python Programming - 2101CS405**

### **Lab - 10**

**Name : Viral chauhan**

**Enrollment : 22010101027**

**Roll No. : 184    Batch : A4**

## **Modules**

**A**

## 01) WAP to create Calculator module which defines functions like add, sub,mul and div. create another file that uses the Calculator module.

```
In [4]: import Calculator as cl
def mycal(a,b,opt):
    return cl.calculate(a,b,opt)
a=int(input("Enter first number "))
b=int(input("Enter second number "))
opt=input("Enter operation you want to perform ")
mycal(a,b,opt)
```

```
Enter first number 6
Enter second number 3
Enter operation you want to perform /
2.0
```

## 02) WAP to Pick a random character from a given String.

```
In [9]: import random as rand

mystr=input("Enter a string ")
x= rand.randint(len(mystr)*-1,len(mystr)-1)
print("Charcter:",mystr[x]," index: ",x)
print("using choice method: ",rand.choice(mystr))
```

```
Enter a string helloworld
Charcter: w index: 5
using choice method: o
```

### 03) WAP to Pick a random element from a given list.

```
In [10]: import random as rand
n = int(input("Enter a length of the list"))
mylist=[input("Enter a number ") for i in range(0,n)]
x= rand.randint(0,len(mylist)-1)
print("element:",mylist[x]," index: ",x)
print("using choice method: ",rand.choice(mylist))
```

```
Enter a length of the list10
Enter a number 10
Enter a number 9
Enter a number 8
Enter a number 7
Enter a number 6
Enter a number 5
Enter a number 4
Enter a number 3
Enter a number 2
Enter a number 1
element: 1 index: 9
using choice method: 4
```

#### 04) WAP to demonstrate the use of the math module.

```
In [11]: import math as mt
print(mt.pi)
print(mt.e)
print(mt.sqrt(10))
print(mt.sin(30))
print(mt.cos(45))
print(mt.tan(60))
print(mt.floor(2.5))
print(mt.ceil(3.2))
print(mt.factorial(5))
print(mt.fabs(-5))
print(mt.pow(2,3))
print(mt.log2(10))
print(mt.log(10))
print(mt.sinh(1))
print(mt.cosh(1))
print(mt.tanh(1))
print(mt.atanh(0))
print(mt.asinh(1))
print(mt.acosh(1))
```

```
3.141592653589793
2.718281828459045
3.1622776601683795
-0.9880316240928618
0.5253219888177297
0.320040389379563
2
4
120
5.0
8.0
3.321928094887362
2.302585092994046
1.1752011936438014
1.5430806348152437
0.7615941559557649
0.0
0.881373587019543
0.0
```

## 05) WAP to demonstrate the use of date time module.

```
In [14]: import datetime as dt
x = dt.datetime.now()
print(x)
print(x.date())
print(x.time())
print(x.year)
print(x.month)
print(x.day)
print(x.hour)
print(x.minute)
print(x.second)
print(x.microsecond)
print(dt.timezone.utc)
print(x.timestamp())
print(x.utcnow())
```

```
2024-02-20 08:45:37.305186
2024-02-20
08:45:37.305186
2024
2
20
8
45
37
305186
UTC
1708398937.305186
2024-02-20 03:15:37.306446
```

## B

### 01) WAP to Roll dice in such a way that every time you get the same number.

```
In [23]: import random as rand
rand.seed(6)
print("Dice roll: ",rand.randint(1,6))
print("Dice roll: ",rand.randint(1,6))
print("Dice roll: ",rand.randint(1,6))
```

```
Dice roll: 5
Dice roll: 1
Dice roll: 4
```

## 02) WAP to generate 3 random integers between 100 and 999 which is divisible by 5.

```
In [20]: import random as rand
l1 = [i for i in range(100,1000) if i%5==0]
print(rand.choice(l1))
print(rand.choice(l1))
print(rand.choice(l1))
```

```
200
720
430
```

## 03) WAP to generate 100 random lottery tickets and pick two lucky tickets from it as a winner.

```
In [25]: import random as rand
print("Winning tickets : [",rand.randint(100000,999999),", ",rand.randint(100000,999999),"]")

Winning tickets : [ 374330 , 138611 ]
```

## 04) WAP to print current date and time in Python.

```
In [26]: import datetime as dt
print(dt.datetime.now())
```

```
2024-02-20 09:03:24.975614
```

## 05) Subtract a week (7 days) from a given date in Python.

```
In [32]: from datetime import datetime, timedelta
x = datetime.now()
y = x - timedelta(days=7)
print(x)
print(y)
```

```
2024-02-20 09:10:19.305249
2024-02-13 09:10:19.305249
```

## 06) WAP to Calculate number of days between two given dates.

```
In [36]: from datetime import datetime, timedelta
x = datetime.now()
y = x + timedelta(days=7)

print("difference is: ",y-x)
```

difference is: 7 days, 0:00:00

## 07) WAP to Find the day of the week of a given date.

```
In [44]: from datetime import datetime

x = datetime.now()
print("on ",x.date()," it was ",x.strftime('%a'))
```

on 2024-02-20 it was Tue

## Extra Programs

```
In [27]: mystr = input("Enter a string")
mystr2=""
for i in range(len(mystr)):
    tempch = mystr[i]
    if(ord(tempch)>= 65 and ord(tempch)<=91):
        mystr2 = mystr2 + chr(97+(ord(tempch)-65))
    elif(ord(tempch)>= 97 and ord(tempch)<=122):
        mystr2 = mystr2+ chr(65+(ord(tempch)-97))
print(mystr2)
```

Enter a stringPaRtH  
pArTh

```
In [2]: n=5

for i in range(0,5):
    temp = str(11**i)
    for j in range(1,n-i+1):
        print(" ",end="")
    for l in temp:
        print(" ",l,end="")
    print()
```

```

      1
    1 1
  1 2 1
1 3 3 1
1 4 6 4 1
```



In [ ]: