9/9/2023

Abhishek Roka

10621019

Python Practical File

Full Time Diploma in Computer Engineering 5th Semester

|  |  |  |
| --- | --- | --- |
| Practical no. | Practical Name | Remarks |
| 1 | Calculate the multiplication and sum of two numbers |  |
| 2 | Print the sum of the current number and the previous number |  |
| 3 | Print characters from a string that are present at an even index number |  |
| 4 | Remove first n characters from a string |  |
| 5 | Check if the first and last number of a list is the same |  |
| 6 | Display numbers divisible by 5 from a list |  |
| 7 | Return the count of a given substring from a string |  |
| 8 | Print the following pattern  1  2 2  3 3 3  4 4 4 4  5 5 5 5 5 |  |
| 9 | Check Palindrome Number |  |
| 10 | Create a new list from a two list using the following condition   * new list should contain odd numbers from the first list and even numbers from the second list |  |
| 11 | Write a Program to extract each digit from an integer in the reverse order. |  |
| 12 | Calculate income tax for the given income by adhering to the below rules   |  |  | | --- | --- | | Taxable Income | Rate (in %) | | First $10,000 | 0 | | Next $10,000 | 10 | | The remaining | 20 | |  |
| 13 | Print multiplication table from 1 to 10 |  |
| 14 | Print downward Half-Pyramid Pattern with Star (asterisk)  \* \* \* \* \*  \* \* \* \*  \* \* \*  \* \*  \* |  |
| 15 | Write a function called exponent(base, exp) that returns an int value of base raises to the power of exp. |  |
| 16 | Write a function called exponent(base, exp) that returns an int value of base raises to the power of exp. |  |
| 17 | Write a program to create function func1() to accept a variable length of arguments and print their value.  **Note:** Create a function in such a way that we can pass any number of arguments to this function, and the function should process them and display each argument’s value. |  |
| 18 | Write a program to create function calculation() such that it can accept two variables and calculate addition and subtraction. Also, it must return both addition and subtraction in a single return call |  |
| 19 | Write a program to create a function show\_employee() using the following conditions.   * It should accept the employee’s name and salary and display both. * If the salary is missing in the function call then assign default value 9000 to salary |  |
| 20 | Create an inner function to calculate the addition in the following way   * Create an outer function that will accept two parameters, a and b * Create an inner function inside an outer function that will calculate the addition of a and b * At last, an outer function will add 5 into addition and return it |  |
| 21 | Write a program to create a recursive function to calculate the sum of numbers from 0 to 10. |  |
| 22 | Assign a different name to function and call it through the new name  Below is the function display\_student(name, age). Assign a new name show\_tudent(name, age) to it and call it using the new name. |  |
| 23 | Generate a Python list of all the even numbers between 4 to 30 |  |
| 24 | Find the largest item from a given list |  |
| 25 | Write a program that defines a function count\_lower\_upper() that accepts a string and calculates the number of uppercase and lowercase alphabets in it. It should return these values as a dictionary. Call this function for some sample strings. |  |
| 26 | Write a program that defines a function compute() that calculates the value of n + nn + nnn + nnnn, where n is digit received by the function. Test the function for digits 4 and 7. |  |
| 27 | Write a program that defines a function create\_array() to create and return a 3D array whose dimensions are passed to the function. Also initialize each element of this array to a value passed to the function. |  |
| 28 | Write a program that defines a function create\_list() to create and return a list which is an intersection of two lists passed to it. |  |
| 29 | Write a program that defines a function sanitize\_list() to remove all duplicate entries from the list that it receives |  |
| 30 | Write a program to receive three integers from keyboard and get their sum and product calculated through a user-defined function cal\_sum\_prod(). |  |
| 31 | Pangram is a sentence that uses every letter of the alphabet. Write a program that checks whether a given string is pangram or not, through a user-defined function ispangram(). |  |
| 32 | Write a Python program that accepts a hyphen-separated sequence of words as input and calls a function convert() which converts it into a hyphen-separated sequence after sorting them alphabetically.  For example, if the input string is here-come-the-dots-followed-by-dashes Then, the output must be: by-come-dashes-dots-followed-here-the |  |
| 33 | Write a python function to create and return a list containing tuples of the form (x, x2, x3) for all x between 1 and 20 (both included). |  |
| 34 | Write a program that defines a function ispalindrome() which checks whether a given string is a palindrome or not. Ignore spaces and case mismatch while checking for palindrome. |  |
| 35 | Write a program that defines a function convert() that receives a string containing a sequence of whitespace separated words and returns a string after removing all duplicate words and sorting them alphanumerically. |  |
| 36 | Write a program that defines a function convert() that receives a string containing a sequence of whitespace separated words and returns a string after removing all duplicate words and sorting them alphanumerically. |  |
| 37 | Write a program that defines a function convert() that receives a string containing a sequence of whitespace separated words and returns a string after removing all duplicate words and sorting them alphanumerically. |  |
| 38 | Write a program that defines two functions called create\_sent1() and create\_sent2().  Both receive following 3 lists:  subjects = [‘He’,’She’]  verb = [‘loves’,’hates’]  objects = [‘TV Serials’,’Netflix’]  Both functions should form sentences by picking elements from the lists and returns them. Use for loops in create\_sent1() and list comprehension in create\_sent2() |  |
| 39 | Perform the following operations on a list of names.  − Create a list of 5 names  – ‘Anil’,’Amol’,’Aditya’,’Avi’,’Alka’  − Insert a name ‘Anuj’ before ‘Aditya’  − Append a name ‘Zulu’  − Delete ‘Avi’ from the list  − Replace ‘Anil’ with ‘AnilKumar’  − Sort all the names in the list  − Print reversed sorted list |  |
| 40 | Perform the following operations on a list of numbers.  − Create a list of 5 odd numbers  − Create a list of 5 even numbers  − Combine the two lists  − Add prime numbers 11,17,29 at the beginning of the combined list  − Report how many elements are present in the list  − Replace last 3 numbers in the list with 100 , 200 , 300  − Delete all the numbers in the list  − Delete the list |  |
| 41 | Write a program to implement a Stack data structure. Stack is a Last In First Out (LIFO) list, in which addition and deletion takes place at the same end. |  |
| 42 | Write a program to implement a Queue data structure. Queue is a First In First Out (FIFO) list, in which addition takes place at the rear end of the queue and deletion takes place at the front end of the queue. |  |
| 43 | Write a program to generate and store in a list 20 random numbers in the range 10 to 100. From this list delete all those entries which have value between 20 and 50. Print the remaining list. |  |
| 44 | Write a program to add two 3x4 matrices. |  |
| 45 | Pass a tuple to divmod() function and obtain the quotient and the remainder. |  |
| 46 | Write a Python program to perform the following operations:  − Pack first 10 multiples of 10 into a tuple  − Unpack the tuple into 10 variables, each holding 1 value  − Unpack the tuple such that first values gets stored in variable x, last value in y and all values in between into disposable variables \_  − Unpack the tuple such that first values gets stored in variable i, last value in j and all values in between into disposable variables \_ |  |
| 47 | A list contains names of boys and girls as its elements. Boys’ names are stored as tuples. Write a Python program to find out number of boys and girls in the list. |  |
| 48 | A list contains tuples containing roll number, names and age of student. Write a Python Program to gather all the names from this list into another list. |  |
| 49 | Given the following tuple  (‘F’,’l’,’a’,’b’,’b’,’e’,’r’,’g’,’a’,’s’,’t’,’e’,’d’)  Write a Python program to carry out the following operations:  − Add an ! at the end of the tuple  − Convert a tuple to a string  − Extract (‘b’,’b’) from the tuple  − Find out number of occurrences of ‘e’ in the tuple  − Check whether ‘r’ exists in the tuple  − Convert the tuple to a list  − Delete characters ‘b’,’b’,’e’,’r’ from the tuple |  |
| 50 | What will be the output of the following program?  a = {10,20,30,40,50,60,70}  b = {33,44,51,10,20,50,30,33}  print(a | b)  print(a & b)  print(a – b)  print(b – a)  print(a ^ b)  print(a >=b)  print(a <= b) |  |
| 51 | What will be the output of the following program?  a = {1,2,3,4,5,6,7}  b = {1,2,3,4,5,6,7}  c = {1,2,3,4,5,6,7}  d = {1,2,3,4,5,6,7}  e= {3,4,1,0,2,5,8,9}  a |= e  print(a)  b &= e  print(b)  c -=e  print(c)  d ^= e  print(d) |  |
| 52 | Write a program to carry out the following operations on the given set  s = {10, 2,-3,4,5,88}  − number of items in set s  − maximum element in set s  − minimum element in set s  − sum of all elements in set s  − obtain a new sorted set from s, set s remaining unchanged  − report whether 100 is an element of set s  − report whether -3 is an element of set s |  |
| 53 | What will be the output of the following program?  l = [10,20,30,40,50]  t = (‘Sundeep’, 25, 79.58)  s = ‘set theory’  s1 = set(l)  s2 = set(t)  s3 = set(s)  print(s1)  print(s2)  print(s3) |  |
| 54 | What will be the output of the following program?  s= {1, 2, 3, 7, 6, 4)  s.discard(10)  s.remove(10)  print(s) |  |
| 55 | What will be the output of the following program?  s1= {10, 20, 30, 40, 50)}  s2 =(10, 20, 30, 40, 50}  print(id(s1), id(s2)) |  |
| 56 | What will be the output of the following program?  s1={10, 20, 30, 40, 50}  s2 = {10, 20, 30, 40, 50}  s3 = (\*s1, \*s2} |  |
| 57 | What will be the output of the following program?  s= set('KanLabs')  t= s[:-1]  print(t) |  |
| 58 | What will be the output of the following program?  num = {10, 20, {30, 40}, 50)  print(num) |  |
| 59 | What will be the output of the following program?  s=(Tiger', 'Lion', 'Jackal'}  del(s)  print(s) |  |
| 60 | What will be the output of the following program?  fruits = ('Kiwi, 'Jack Fruit', 'Lichi'}  fruits.clear()  print(fruits) |  |
| 61 | What will be the output of the following program?  s = {10, 25, 4, 12, 3, 8}  s = sorted(s)  print(s) |  |
| 62 | What will be the output of the following program?  s = {}  t = {1,4,5,2,3}  print(type(s), type(t)) |  |
| 63 | A set contains names which begin either with A or with B. write a program to separate out the names into two sets, one containing names beginning with A and another containing names beginning with B. |  |
| 64 | Create an empty set. Write a program that adds five new names to this set, modifies one existing name and deletes two names existing in it. |  |
| 65 | Write a program to create a set containing 10 randomly generated numbers in the range 15 to 45. Count how many of these numbers are less than 30. Delete all numbers which are greater than 35. |  |
| 66 | What will be the output of the following program?  s = {'Mango', 'Banana', 'Guava', 'Kiwi'}  s.clear( )  print(s)  del(s)  print(s) |  |

**Practical 1**

Calculate the multiplication and sum of two numbers

**Code**

num1 = int(input("Enter num1: "))

num2= int(input("Enter num2: "))

mul = num1\*num2

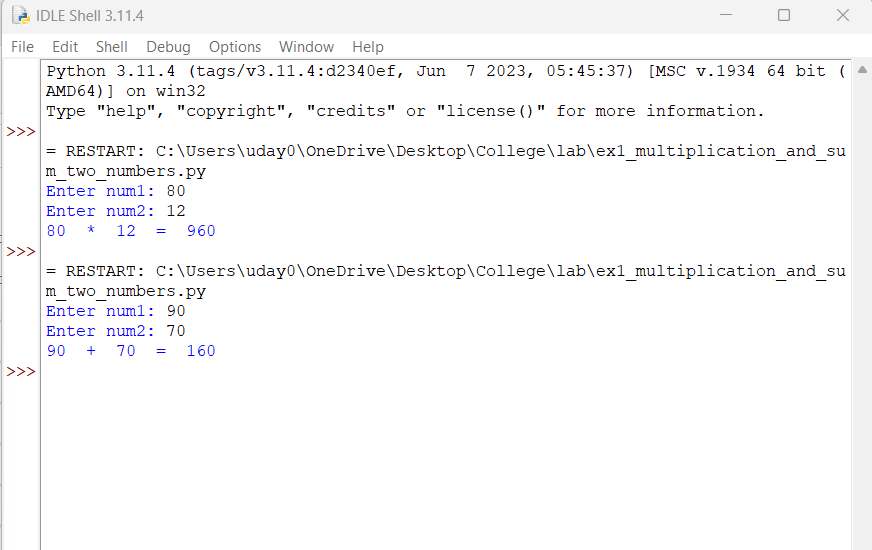
if mul <= 1000:

print(num1, " \* ", num2, " = ", mul)

else:

print(num1, " + ", num2, " = ", num1+num2)

**Output:**



**Practical 2**

Print the sum of the current number and the previous number

**Code:**

prev\_num = 0

rangelen = range(10)

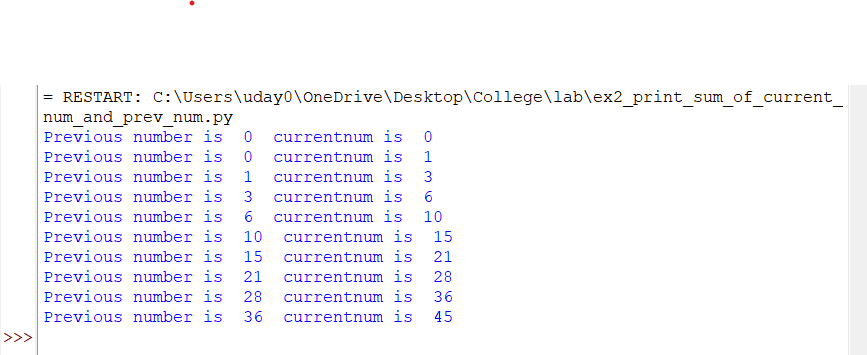
currentnum=0

for i in rangelen:

currentnum = prev\_num + i

print("Previous number is ", prev\_num, " currentnum is ", currentnum)

prev\_num = currentnum

**Output:**

**Practical 3**

Print characters from a string that are present at an even index number

**Code:**

user\_string = input("Enter string: ")

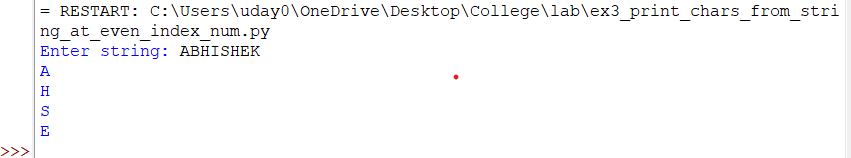
length = len(user\_string)

range\_len = range(0,length,2)

for i in range\_len:

print(user\_string[i])

**Output:**



**Practical 4**

Remove first n characters from a string

**Code:**

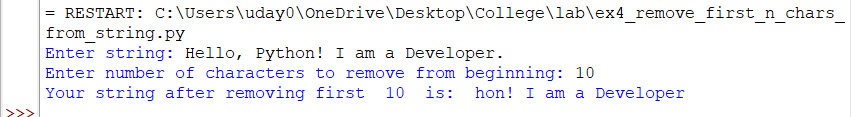
user\_string = input("Enter string: ")

num\_of\_chars = int(input("Enter number of characters to remove from beginning: "))

new\_string = user\_string[num\_of\_chars:-1]

print("Your string after removing first ", num\_of\_chars, " is: ", new\_string)

**Output:**



**Practical 5**

Check if the first and last number of a list is the same

**Code:**

num = int(input("Enter number of elements to add in list: "))

rangeLen = range(num)

user\_list = []

for i in rangeLen:

print("Enter element no.", i+1, " : ", end="")

user\_num = int(input())

user\_list.append(user\_num)

if user\_list[0] == user\_list[-1]:

print("First and last numbers are same.")

else:

print("First and last numbers are not same.")

**Output:**

### 

**Practiaal 6**

Display numbers divisible by 5 from a list

**Code:**

user\_list = []

length = int(input("Enter number of elements to be entered in list: "))

rangeLen = range(length)

for i in rangeLen:

print("Enter element no.", i, " : ", end="")

num = int(input())

user\_list.append(num)

for i in user\_list:

if i%5==0:

print(i," is divisible by 5.")

**Output:**

### 

**Practiacal 7**

Return the count of a given substring from a string

**Code:**

string\_to\_examin = "Hello, World! I am a Python Developer. I am a freelancer. Also work on web development. Hello, Development"

to\_find\_string = "Hello"

print(to\_find\_string, " occured ", string\_to\_examin.count(to\_find\_string), " times in ", string\_to\_examin)

**Output:**

### 

**Practical 8**

Print the following pattern

### 1

### 2 2

### 3 3 3

### 4 4 4 4

### 5 5 5 5 5

**Code:**

num= int(input("Enter number for triangle: "))

heightRangeLen = range(num)

for i in heightRangeLen:

lengthRangelen = range(i+1)

for j in lengthRangelen:

print(i+1, end=" ")

print()

**Output:**

### 

**Practical 9**

Check Palindrome Number

**Code:**

user\_num = input("Enter number: ")

if user\_num == user\_num[::-1]:

print(user\_num, " is a palindrome number.")

else:

print(user\_num," is not a palindrome number.")

**Output:**

### 

**Practical 10**

Create a new list from a two list using the following condition

* new list should contain odd numbers from the first list and even numbers from the second list.

**Code:**

list1= [23,44,567,67,88,987]

list2 = [24, 45, 89, 90, 33]

new\_list = []

for i in list1:

if i%2==1:

new\_list.append(i)

for i in list2:

if i%2==0:

new\_list.append(i)

print("New list is: ", new\_list)

**Output:**

### 

**Practical 11**

Write a Program to extract each digit from an integer in the reverse order.

**Code:**

num = int(input("Enter integer: "))

new\_num=0

store\_orginal\_num = num

while num>0:

new\_num = new\_num\*10 + num%10

num = num // 10

print("Digits of ", store\_orginal\_num, " extracted in reverse order is: ", new\_num)

**Output:**

### 

**Practical 12**

Calculate income tax for the given income by adhering to the below rules

|  |  |
| --- | --- |
| Taxable Income | Rate (in %) |
| First $10,000 | 0 |
| Next $10,000 | 10 |
| The remaining | 20 |

**Code:**

"""

Taxable income Rate

10,000 0

next 10,000 10

remaining 20

"""

income\_tax = 0

your\_income = float(input("Your income is $ "))

income\_division = []

if your\_income<10000:

income\_tax=0

elif your\_income < 20000:

income\_division.append(10000)

income\_division.append(your\_income-10000)

else:

income\_division.append(10000)

income\_division.append(10000)

income\_division.append(your\_income-20000)

length = len(income\_division)

rangeLen = range(length)

for i in rangeLen:

if i == 1:

income\_tax += income\_division[i]\*0.1

elif i > 1:

income\_tax += income\_division[i]\*0.2

print("Your income tax is: $", income\_tax, "/-")

**Output:**

### 

**Practical 13**

Print multiplication table form 1 to 10

**Code:**

table\_of = range(1,11)

table\_numbers = range(1,11)

for i in table\_of:

print("Table of ", i)

for j in table\_numbers:

print(i, " x ", j, " = ", i\*j)

print()

**Output:**

### 

### 

**Practical 14**

Print downward Half-Pyramid Pattern with Star (asterisk)

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**Code:**

length\_of\_triangle\_height = int(input("Enter height of triangle: "))

order\_of\_height = range(length\_of\_triangle\_height,0,-1)

for i in order\_of\_height:

elements\_range = range(i)

for j in elements\_range:

print("\*",end=" ")

print()

**Output:**

### 

**Practical 15**

Write a function called exponent(base, exp) that returns an int value of base raises to the power of exp.

**Code:**

def exponent(base, exp):

return base \*\* exp

base = float(input("base= "))

exp = float(input("exponent= "))

print(base, " ^ ", exp, " = ", exponent(base,exp))

**Output:**

### 

**Practical – 16**

Write a program to create a function that takes two arguments, name and age, and print their value.

### Code:

### def function(name, age):

### print("name is: ", name)

### print("age is: ", age)

### function("John","20")

### Output:

### 

**Practical 17**

Write a program to create function func1() to accept a variable length of arguments and print their value.

**Note**: Create a function in such a way that we can pass any number of arguments to this function, and the function should process them and display each argument’s value.

### Code:

### def func1(\*args):

### print("sum",args, " is: ", sum(args))

### 

### func1(20,30,40)

### func1(10,30,50,60,100)

### Output:

### 

**Practical-18**

Write a program to create function calculation() such that it can accept two variables and calculate addition and subtraction. Also, it must return both addition and subtraction in a single return call

### Code:

### def calculation(num1,num2):

### return num1+num2, num1-num2

### 

### x = float(input("x = "))

### y = float(input("y = "))

### result = calculation(x,y)

### print("x + y = ", result[0])

### print("x - y = ", result[1])

### Output:

### 

**Practical 19**

Write a program to create a function show\_employee() using the following conditions.

* It should accept the employee’s name and salary and display both.
* If the salary is missing in the function call then assign default value 9000 to salary

### Code:

### def show\_employee(name, salary=9000):

### print("Employee name: ", name)

### print("Employee salary: ", salary)

### name = input("Enter name: ")

### salary = int(input("Enter salary: "))

### print("\nPassing name only:-")

### show\_employee(name)

### print("\nPassing name and salary")

### show\_employee(name, salary)

### Output:

### 

**Practical – 20**

Create an inner function to calculate the addition in the following way

* Create an outer function that will accept two parameters, a and b
* Create an inner function inside an outer function that will calculate the addition of a and b
* At last, an outer function will add 5 into addition and return it

### Code:

### def calculation(a,b):

### result = 0

### def addition():

### return a+b

### result += addition()

### result += 5

### return result

### print(calculation(3,5))

### Output:

### 

**Practical – 21**

Write a program to create a recursive function to calculate the sum of numbers from 0 to 10.

### Code:

### def sumNumbers(n):

### if n==0:

### return 0

### else:

### return n+sumNumbers(n-1)

### print(sumNumbers(10))

### Output:

### 

**Practical – 22**

Assign a different name to function and call it through the new name

Below is the function display\_student(name, age). Assign a new name show\_tudent(name, age) to it and call it using the new name.

### Code:

### def display\_student(name, age):

### print("Student name is: ", name)

### print("Student age is: ", age)

### show\_student = display\_student

### show\_student("Rohan",18)

### Output:

### 

**Practical – 23**

Generate a Python list of all the even numbers between 4 to 30

### Code:

### def even\_numbers\_4\_to\_30():

### return list(range(4,30,2))

### print("List is: ", even\_numbers\_4\_to\_30())

### Output:

### 

**Practical – 24**

Find the largest item from a given list

### Code:

### myList = [4,6,5,8,80,90,10,3,2]

### print("Largest item in the list: ", max(myList))

### Output:

### 

**Practical-25**

Write a program that defines a function count\_lower\_upper() that accepts a string and calculates the number of uppercase and lowercase alphabets in it. It should return these values as a dictionary. Call this function for some sample strings.

### Code:

### def count\_lower\_upper(userstring):

### record = {"lower":0, "upper":0}

### characterRange = list(range(65,91)) + list(range(97,123))

### for i in userstring:

### if ord(i) in characterRange:

### if i.islower():

### record["lower"]+=1

### elif i.isupper():

### record["upper"]+=1

### return record

### while True:

### user = input("Enter String: ")

### print(count\_lower\_upper(user))

### Output:

### 

**Practical – 26**

Write a program that defines a function compute() that calculates the value of n + nn + nnn + nnnn, where n is digit received by the function. Test the function for digits 4 and 7.

### Code:

### def compute(n):

### return n \* (1+11+111+1111)

### print(compute(4))

### print(compute(7))

### Output:

### 

**Practical – 27**

Write a program that defines a function create\_array() to create and return a 3D array whose dimensions are passed to the function. Also initialize each element of this array to a value passed to the function.

### Code:

### def create\_array(dims,values):

### l=0

### array = []

### if len(values)==dims[0]\*dims[1]\*dims[2]:

### for i in range(dims[0]):

### subarray=[]

### for j in range(dims[1]):

### subsubarray=[]

### for k in range(dims[2]):

### subsubarray.append(values[l])

### l+=1

### subarray.append(subsubarray)

### array.append(subarray)

### return array

### aList = []

### for i in range(3\*8\*3):

### aList.append(i)

### print(create\_array((3,8,3), aList))

### Output:

### 

**Practical – 28**

Write a program that defines a function create\_list() to create and return a list which is an intersection of two lists passed to it.

### Code:

### def create\_array(list1,list2):

### list3=[]

### for i in list1:

### if i in list2:

### list3.append(i)

### return list3

### print(create\_array([3, 2, 5, 2, 8, 8, 8, 7, 4, 2],[8, 1, 7, 7, 1, 8, 5, 4, 2, 6]))

### Output:

### 

**Practical – 29**

Write a program that defines a function sanitize\_list() to remove all duplicate entries from the list that it receives.

### Code:

### def sanitize\_list(givenList):

### newList = []

### for i in set(givenList):

### newList.append(i)

### return newList

### print(sanitize\_list([8, 8, 6, 4, 1, 6, 1, 5, 8, 5]))

### Output:

### 

**Practical – 30**

Write a program to receive three integers from keyboard and get their sum and product calculated through a user-defined function cal\_sum\_prod().

### Code:

### def cal\_sum\_prod(num1,num2,num3):

### return num1+num2+num3, num1\*num2\*num3

### n1 = int(input("Enter first integer: "))

### n2 = int(input("Enter second integer: "))

### n3 = int(input("Enter third integer: "))

### result = cal\_sum\_prod(n1,n2,n3)

### print("Sum is: ", result[0])

### print("Product is: ", result[1])

### Output:

### 

**Practical – 31**

Pangram is a sentence that uses every letter of the alphabet. Write a program that checks whether a given string is pangram or not, through a user-defined function ispangram().

### Code:

### def ispangram(sentence):

### alphabets = [chr(i) for i in range(65,91)] + [chr(i) for i in range(97,123)] #list of all alphabets lowercase and uppercase

### for i in set(sentence):

### if i not in alphabets:

### return False

### return True

### statement = input("Enter statment to check for pangram: ")

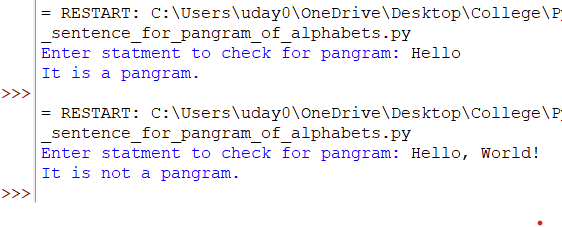
### if ispangram(statement):

### print("It is a pangram.")

### else:

### print("It is not a pangram.")

### Output:



**Practical – 32**

Write a Python program that accepts a hyphen-separated sequence of words as input and calls a function convert() which converts it into a hyphen-separated sequence after sorting them alphabetically. For example, if the input string is

here-come-the-dots-followed-by-dashes

Then, the output must be:

by-come-dashes-dots-followed-here-the

### Code:

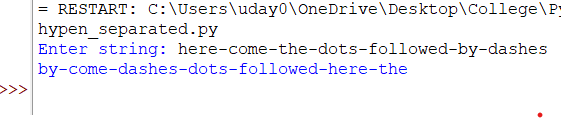
### def convert(sentence):

### return "-".join(sorted(sentence.split("-")))

### user\_input = input("Enter string: ")

### print(convert(user\_input))

### Output:



**Practical – 33**

Write a python function to create and return a list containing tuples of the form (x, x2, x3) for all x between 1 and 20 (both included).

### Code:

### def x\_square\_cube():

### returnList = []

### for x in range(1,21):

### returnList.append((x,x\*\*2,x\*\*3))

### return returnList

### print(x\_square\_cube())

### Output:

### 

**Practical – 34**

Write a program that defines a function ispalindrome() which checks whether a given string is a palindrome or not. Ignore spaces and case mismatch while checking for palindrome.

### Code:

### def ispalindrome(arg\_string):

### arg\_string = arg\_string.replace(" ","").lower()

### if arg\_string == arg\_string[::-1]:

### return True

### return False

### user\_string = input("Enter string: ")

### print("Palindrome condition: ",ispalindrome(user\_string))

### Output:

### 

**Practical – 35**

Write a program that defines a function convert() that receives a string containing a sequence of whitespace separated words and returns a string after removing all duplicate words and sorting them alphanumerically.

### Code:

### def convert(arg\_string):

### x = sorted(set(arg\_string.split(" ")))

### return " ".join(x)

### sentence = input("Enter string: ")

### print(convert(sentence))

### Output:

### 

**Practical – 36**

Write a program that defines a function count\_alphabets\_digits() that accepts a string and calculates the number of alphabets and digits in it. It should return these values as a dictionary. Call this function for some sample strings.

### Code:

### def count\_alphabets\_digits(userstring):

### record = {"alphabets":0, "digits":0}

### characterRange = list(range(65,91)) + list(range(97,123)) + list(range(48,58))

### for i in userstring:

### if ord(i) in characterRange:

### if i.isalpha():

### record["alphabets"]+=1

### elif i.isdigit():

### record["digits"]+=1

### return record

### while True:

### user = input("Enter String: ")

### print(count\_alphabets\_digits(user))

### Output:

### 

**Practical – 37**

Write a program that defines a function called frequency() which computes the frequency of words present in a string passed on it. The frequencies should be returned in sorted order by words in the string.

### Code:

### def frequency(arg\_string):

### returnDict = {}

### arg\_string = arg\_string.split(" ")

### for i in set(arg\_string):

### returnDict[i] = 0

### for i in arg\_string:

### returnDict[i] +=1

### return returnDict

### user\_string = input("Enter string: ")

### print(frequency(user\_string))

### Output:

### 

**Practical – 38**

Write a program that defines two functions called create\_sent1() and create\_sent2().

### Both receive following 3 lists:

### subjects = [‘He’,’She’]

### verb = [‘loves’,’hates’]

### objects = [‘TV Serials’,’Netflix’]

### Both functions should form sentences by picking elements from the lists and returns them. Use for loops in create\_sent1() and list comprehension in create\_sent2().

### Code:

### def create\_sent1(sub,verb,obj):

### # using for loop

### sentences = []

### for i in sub:

### for j in verb:

### for k in obj:

### sentences.append(" ".join((i,j,k)))

### return sentences

### def create\_sent2(sub,verb,obj):

### # Using list comprehension

### return [" ".join((i,j,k)) for i in sub for j in verb for k in obj]

### subject = ['He','She']

### verb = ['loves', 'hates']

### obj = ['TV Serial','Netflix']

### print("Using create\_sent1() function which uses for loop:")

### for i in create\_sent1(subject,verb,obj):

### print(i)

### 

### print("\nUsing create\_sent2() function which uses list comprehension:")

### for i in create\_sent2(subject,verb,obj):

### print(i)

### Output:

### 

### Practical – 39

Perform the following operations on a list of names.

### Create a list of 5 names – ‘Anil’,’Amol’,’Aditya’,’Avi’,’Alka’

### Insert a name ‘Anuj’ before ‘Aditya’

### Append a name ‘Zulu’

### Delete ‘Avi’ from the list

### Replace ‘Anil’ with ‘AnilKumar’

### Sort all the names in the list

### Print reversed sorted list

### Code:

### #initializing the list

### namelist=['Anil','Amol','Aditya','Avi','Alka']

### #inserting Anuj before Aditya in list.

### namelist.insert(namelist.index('Aditya'),'Anuj')

### print("Name list after inserting 'Anuj' before 'Aditya' : ", namelist)

### # Appeding name Zulu to the list

### namelist.append('Zulu')

### print("Name list after appending name 'Zulu' to list: ", namelist)

### # removing Avi from the list

### namelist.remove('Avi')

### print("Name list after removing 'Avi' from the list: ", namelist)

### # replacing Anil with AnilKumar

### namelist[namelist.index('Anil')]='AnilKumar'

### print("Name list after replacing 'Anil' with 'AnilKumar' in the list: ", namelist)

### # Sorting the list

### namelist.sort()

### print("Name list after names in the list: ", namelist)

### # reversing the list

### namelist.reverse()

### print("Name list after reversing the order of names in the list: ", namelist)

### Output:

### 

### Practical – 40

Perform the following operations on a list of numbers.

### Create a list of 5 odd numbers

### Create a list of 5 even numbers

### Combine the two lists

### Add prime numbers 11,17,29 at the beginning of the combined list

### Report how many elements are present in the list

### Replace last 3 numbers in the list with 100 , 200 , 300

### Delete all the numbers in the list

### Delete the list

### Code:

### odd\_num\_list = list(range(1,10,2)) # created list of 5 odd numbers

### even\_num\_list = list(range(2,11,2)) # created list of 5 even numbers

### combined\_list = odd\_num\_list + even\_num\_list # combining both list

### print("Combinded list: ", combined\_list)

### combined\_list = [11,17,29] + combined\_list

### print("After adding prime number to the beginning of the list: ", combined\_list)

### print("Number of elements present in th e list = ", len(combined\_list))

### combined\_list[-3:len(combined\_list)] = [100,200,300]

### print("list after replacing last 3 numbers with 100, 200, 300: ", combined\_list)

### del combined\_list[0:]

### print("list after deleting all numbers: ", combined\_list)

### del combined\_list

### print("List deleted!")

### Output:

### 

### Practical – 41

Write a program to implement a Stack data structure. Stack is a Last In First Out (LIFO) list, in which addition and deletion takes place at the same end.

### Code:

### stack = []

### def push(element):

### stack.append(element)

### def poping():

### print("Popped: ",stack.pop())

### while True:

### print("""\n\nOperations to be done on stack:

### 1. Push

### 2. Pop

### 3. Display stack

### e. Exit

### Enter: """, end=" ")

### option = input()

### if option=='1':

### element = input("Enter element to push: ")

### push(element)

### elif option=='2':

### poping()

### elif option=='3':

### print("Stack is: ", stack)

### elif option.lower() == 'e':

### print("Program Terminated...")

### break

### else:

### print("Invalid option!")

### Output:

### Push

### 

### Pop

### 

### Display (Traverse)

### 

### Exit

### 

### Practical – 42

Write a program to implement a Queue data structure. Queue is a First In First Out (FIFO) list, in which addition takes place at the rear end of the queue and deletion takes place at the front end of the queue.

### Code:

### queue = []

### def enqueue(element):

### queue.append(element)

### def dequeue():

### global queue

### print("Dequeued element: ", queue[0])

### queue=queue[1:]

### while True:

### print("""\n\nOperations to be done on queue:

### 1. Enqueue

### 2. Dequeue

### 3. Display queue

### e. Exit

### Enter: """, end=" ")

### option = input()

### if option=='1':

### element = input("Enter element to enqueue: ")

### enqueue(element)

### elif option=='2':

### dequeue()

### elif option=='3':

### print("queue is: ", queue)

### elif option.lower() == 'e':

### print("Program Terminated...")

### break

### else:

### print("Invalid option!")

### Output:

### Enqueue

### 

### Dequeue

### 

### Display (Traverse)

### 

### Exit

### 

### Practical – 43

Write a program to generate and store in a list 20 random numbers in the range 10 to 100. From this list delete all those entries which have value between 20 and 50. Print the remaining list.

### Code:

### import random

### random\_num\_list = [random.randrange(10,101) for i in range(20)]

### print("Random number list: ", random\_num\_list)

### i=0

### while i<len(random\_num\_list):

### if random\_num\_list[i]>=20 and random\_num\_list[i]<=50:

### del random\_num\_list[i]

### else:

### i+=1

### print("List after removing elements between 20 and 50: ", random\_num\_list)

### Output:

### 

**Practical – 44**

Write a program to add two 3x4 matrices.

### Code:

### mat1=[[1,2,3,4],

### [5,6,7,8],

### [9,10,11,12]]

### mat2=[[1,2,3,4],

### [5,6,7,8],

### [9,10,11,12]]

### mat3=[[0,0,0,0],

### [0,0,0,0],

### [0,0,0,0]]

### i,j=0,0

### while i<3:

### while j<4:

### mat3[i][j] = mat1[i][j] + mat1[i][j]

### j+=1

### i+=1

### j=0

### print("Sum is: ", mat3)

### Output:

### 

### Practical – 45

Pass a tuple to divmod() function and obtain the quotient and the remainder.

### Code:

### def divmod(numtuple):

### return numtuple[0]//numtuple[1], numtuple[0]%numtuple[1]

### result = divmod((10,3))

### print("Quotient is: ", result[0])

### print("Remainder is: ", result[1])

### Output:

### 

### Practical – 46

### Write a Python program to perform the following operations:

### Pack first 10 multiples of 10 into a tuple

### Unpack the tuple into 10 variables, each holding 1 value

### Unpack the tuple such that first values gets stored in variable x, last value in y and all values in between into disposable variables \_

### Unpack the tuple such that first values gets stored in variable i, last value in j and all values in between into disposable variables \_

### Code:

### multiple\_of\_ten = tuple([i for i in range(10,110,10) if i%10==0])

### v1,v2,v3,v4,b5,v6,v7,v8,v9,v10 = multiple\_of\_ten

### x,\*\_,y = multiple\_of\_ten

### i,\*\_,j = multiple\_of\_ten

### print(x,\_,y)

### print(i,\_,j)

### Output:

### 

### Practical – 47

A list contains names of boys and girls as its elements. Boys’ names are stored as tuples. Write a Python program to find out number of boys and girls in the list.

### Code:

### names\_list = ["Nidhi","Niharika","Meena",("Abhishek","Bunty","Manan","Rohit"),"Rashmi","Drishti"]

### girls = 0

### boys=0

### i=0

### while i<len(names\_list):

### if type(names\_list[i]) is tuple:

### boys=len(names\_list[i])

### else:

### girls+=1

### i+=1

### print("Number of boys=", boys, "\nNumber of girls=",girls)

### Output:

### 

**Practical – 48**

A list contains tuples containing roll number, names and age of student. Write a Python Program to gather all the names from this list into another list.

### Code:

### data = [(1,"Abhishek",19),(2,"Bunty",18),(3,"Manan",19),(4,"Rohit",19)]

### names = []

### for i in data:

### names.append(i[1])

### print("Names of student in data: ", names)

### Output:

### 

### Practical – 49

Given the following tuple

(‘F’,’l’,’a’,’b’,’b’,’e’,’r’,’g’,’a’,’s’,’t’,’e’,’d’)

Write a Python program to carry out the following operations:

### Add an ! at the end of the tuple

### Convert a tuple to a string

### Extract (‘b’,’b’) from the tuple

### Find out number of occurrences of ‘e’ in the tuple

### Check whether ‘r’ exists in the tuple

### Convert the tuple to a list

### Delete characters ‘b’,’b’,’e’,’r’ from the tuple

### Code:

### data = ('F','l','a','b','b','e','r','g','a','s','t','e','d') #create tuple

### # adding '!' in end of tuple.

### data = list(data) #unpacking tuple

### data.append('!')

### data=tuple(data)#packing tuple

### print("tuple after appending '!' in it: ", data)

### # converting tuple to string

### tupletostring = "".join(data)

### print("After converting tuple to string: ", tupletostring)

### # extracting ('b','b') from given tuple

### if data[data.index('b')+1] == 'b':

### print(data[data.index('b'):data.index('b')+2])

### # Counting occurance of 'e' in given tuple

### print("Number of occurance of 'e' is ", data.count('e'))

### # checking 'r' exists in data or not

### print("'r' exists in given tuple: ", 'r' in data)

### # converting tuple to list

### data = list(data)

### print("After converting tuple to list: ", data)

### # deleting characters 'b','b','e','r' from given tuple

### data = tuple("".join(data).replace('bber',""))

### print("tuple after deleting characters 'b','b','e','r': ", data)

### Output:

### 

### Practical-50

### What will be the output of the following program?

### a = {10,20,30,40,50,60,70}

### b = {33,44,51,10,20,50,30,33}

### print(a | b)

### print(a & b)

### print(a – b)

### print(b – a)

### print(a ^ b)

### print(a >=b)

### print(a <= b)

### Output:

### 

### Practical – 51

### What will be the output of the following program?

### a = {1,2,3,4,5,6,7}

### b = {1,2,3,4,5,6,7}

### c = {1,2,3,4,5,6,7}

### d = {1,2,3,4,5,6,7}

### e= {3,4,1,0,2,5,8,9}

### a |= e

### print(a)

### b &= e

### print(b)

### c -=e

### print(c)

### d ^= e

### print(d)

### Output:

### 

### Practical – 52

### Write a program to carry out the following operations on the given set

### s = {10, 2,-3,4,5,88}

### number of items in set s

### maximum element in set s

### minimum element in set s

### sum of all elements in set s

### obtain a new sorted set from s, set s remaining unchanged

### report whether 100 is an element of set s

### report whether -3 is an element of set s

### Code:

### s = {10,2,-3,4,5,88}

### print("Number of items in s= ",len(s))

### print("Maximum element in s= ", max(s))

### print("Minimum element in s= ", min(s))

### print("Sum of all elements in s= ", sum(s))

### k=set(sorted(list(s)))

### print("New set is: ", k)

### print("100 is element of s: ", 100 in s)

### print("-3 is element of s: ", -3 in s)

### Output:

### 

### Practical – 53

### What will be the output of the following program?

### l = [10,20,30,40,50]

### t = (‘Sundeep’, 25, 79.58)

### s = ‘set theory’

### s1 = set(l)

### s2 = set(t)

### s3 = set(s)

### print(s1)

### print(s2)

### print(s3)

### Output:

### 

### Practical – 54

### What will be the output of the following program?

### s= {1, 2, 3, 7, 6, 4)

### s.discard(10)

### s.remove(10)

### print(s)

### Output:

### Element 10 is not present in set s.discard( ) would do nothing, whereas, remove( ) would report an error.

### 

### Practical – 55

### What will be the output of the following program?

### s1= {10, 20, 30, 40, 50)}

### s2 =(10, 20, 30, 40, 50}

### print(id(s1), id(s2))

### Output:

### 

### Practical - 56

### What will be the output of the following program?

### s1={10, 20, 30, 40, 50}

### s2 = {10, 20, 30, 40, 50}

### s3 = (\*s1, \*s2}

### print(s3)

### Output:

### 

### Practical - 57

### What will be the output of the following program?

### s= set('KanLabs')

### t= s[:-1]

### print(t)

### Output:

### Error: set is not a subscriptable object. In other words [ ] cannot be used with a set.

### 

### Practical – 58

### What will be the output of the following program?

### num = {10, 20, {30, 40}, 50)

### print(num)

### Output:

### Error: Nested sets are illegal.

### 

### Practical - 59

### What will be the output of the following program?

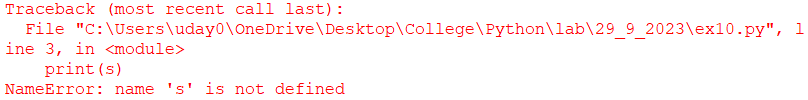
s=(Tiger', 'Lion', 'Jackal'}

del(s)

print(s)

**Output:**

Error: name 's' is not defined. This happens because del( ) deletes the set object.



### Practical - 60

### What will be the output of the following program?

### fruits = ('Kiwi, 'Jack Fruit', 'Lichi'}

### fruits.clear()

### print(fruits)

### Output:

### set( ) . After calling clear( ), fruits becomes an empty set.

### 

### Practical – 61

### What will be the output of the following program?

### s = {10, 25, 4, 12, 3, 8}

### s = sorted(s)

### print(s)

### Output:

### 

### Practical – 62

### What will be the output of the following program?

### s = {}

### t = {1,4,5,2,3}

### print(type(s), type(t))

### Output:

### 

### Practical – 63

### A set contains names which begin either with A or with B. write a program to separate out the names into two sets, one containing names beginning with A and another containing names beginning with B.

### Code:

### simple\_set = {'Abhishek','Bunty','Ashish','Bhawna','Bimal','Aman','Bhavesh'}

### A\_name\_set = set()

### B\_name\_set = set()

### for i in simple\_set:

### if i.lower().startswith('a'):

### A\_name\_set.add(i)

### else:

### B\_name\_set.add(i)

### print("Set with names begins with A: ", A\_name\_set)

### print("Set with names begins with B: ", B\_name\_set)

### Output:

### 

### Practical – 64

### Create an empty set. Write a program that adds five new names to this set, modifies one existing name and deletes two names existing in it.

### Code:

### names = set()

### names.add("Abhishek")

### names.add("Bunty")

### names.add("Manan")

### names.add("Rohit")

### names.add("Nidhi")

### print(names)

### names = set(((" ".join(names)).replace("Abhishek","AbhishekRoka")).split(" "))

### print(names)

### names.discard("Rohit")

### names.discard("Nidhi")

### print(names)

### Output:

### 

### Practical – 65

### Write a program to create a set containing 10 randomly generated numbers in the range 15 to 45. Count how many of these numbers are less than 30. Delete all numbers which are greater than 35.

### Code:

### from random import randrange

### nums = set([randrange(15,46) for i in range(10)])

### count = 0

### greaterthan35 = []

### for i in nums:

### if i<30:

### count+=1

### elif i > 35:

### greaterthan35.append(i)

### for i in greaterthan35:

### nums.discard(i)

### print("Numbers less than 30 in given set: ", count)

### print("Set after deleting all numbers greater than 35: ", nums)

### Output:

### 

### Practical – 66

### What will be the output of the following program?

### s = {'Mango', 'Banana', 'Guava', 'Kiwi'}

### s.clear( )

### print(s)

### del(s)

### print(s)

### Output:

### set( )

### NameError: name 's' is not defined

### 

### 

### Practical – 67

### Create a dictionary called students containing names and ages. Copy the dictionary into stud. Empty the students dictionary, as stud continues to hold the data.

### Code: students= {

### "Abhishek":19,

### "Bunty":18,

### "Manan":20,

### "Mayank":18

### }

### stud = students.copy()

### print("Initially, students=",students, "\nand, stud=" ,stud)

### students.clear()

### print("After clearing students dict")

### print("students=",students, "\nand, stud=" ,stud)

### Output:

### 

### Practical – 68

### Create a list of cricketers. Use this list to create a dictionary in which the list values become keys of the dictionary. Set the values of all keys to 50 in the dictionary created.

### Code:

### cricketers = ["Virat","Rohit","Yujvendra","Brett","Mitchell","Chris"]

### cricketers\_dict = dict()

### for i in cricketers:

### cricketers\_dict[i] = 50

### print("Created cricketers dict from list is: ", cricketers\_dict)

### Output:

### 

### Practical – 69

### Write a program to sort a dictionary in ascending/descending order by key and ascending/descending order by value.

Code:

a\_dict = {

1:2,

2:3,

3:1,

4:0

}

while True:

print("Options:")

print("1. Sort in ascending order by key")

print("2. Sort in descending order by key")

print("3. Sort in ascending order by value")

print("4. Sort in descending order by value")

print("5. Exit")

choice = input("Enter your choice: ")

if choice == "1":

sorted\_dict = dict(sorted(a\_dict.items()))

print("Ascending order by key:", sorted\_dict)

elif choice == "2":

sorted\_dict = dict(sorted(a\_dict.items(), reverse=True))

print("Descending order by key:", sorted\_dict)

elif choice == "3":

sorted\_dict = dict(sorted(a\_dict.items(), key=lambda item: item[1]))

print("Ascending order by value:", sorted\_dict)

elif choice == "4":

sorted\_dict = dict(sorted(a\_dict.items(), key=lambda item: item[1], reverse=True))

print("Descending order by value:", sorted\_dict)

elif choice == "5":

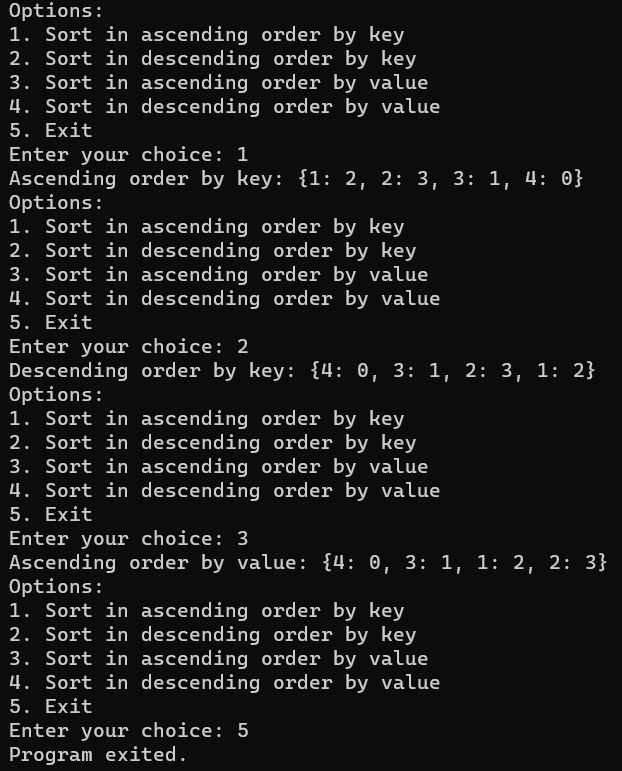
break

else:

print("Invalid choice. Please enter a valid option.")

print("Program exited.")

Output:



### Practical – 70

### Write a program to create three dictionaries and concatenate them to create fourth dictionary.

Code:

students= {

"Abhishek":19,

"Bunty":18,

"Manan":20,

"Mayank":18

}

cricketers = {'Virat': 50, 'Rohit': 50, 'Yujvendra': 50, 'Brett': 50, 'Mitchell': 50, 'Chris': 50}

new\_dict = {

1:2,

2:3,

3:1,

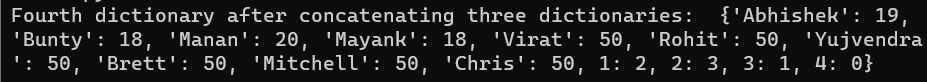
4:0

}

forth\_dict = {\*\*students,\*\*cricketers,\*\*new\_dict}

print("Fourth dictionary after concatenating three dictionaries: ", forth\_dict)

Output:



### Practical – 71

### Write a program to check whether a dictionary is empty or not.

Code:

def isEmptyDict(a\_dict):

if len(a\_dict) <= 0:

return True

else:

return False

new\_dict = {

1:2,

2:3,

3:1,

4:0

}

empty\_dict = dict()

print("is new\_dict empty? ", isEmptyDict(new\_dict))

print("is empty\_dict empty? ", isEmptyDict(empty\_dict))

Output:



### Practical – 72

### Suppose there are two dictionaries called boys and girls containing names as keys and ages as values. Write a program to merge the two dictionaries into a third dictionary.

### Code:

### Practical – 73

### For the following dictionary, write a program to report the maximum and minimum salary.

Code:

employee = {

"Abhishek":1000000,

"Bunty":80000,

"Manan":90000,

"Rohit":95000

}

print("Maximum salary: ",max(employee.values()),"\nMinimum salary: ",min(employee.values()))

Output:



### Practical – 74

### Suppose a dictionary contains roll numbers and names of students. Write a program to receive the roll number, extract the name corresponding to the roll number and display a message congratulating the by his name. If the roll number doesn’t exist in the dictionary, the message should be ‘Congratulations Student!’.

Code:

students= {

"10621019":"Abhishek",

"10621121":"Bunty",

"10621001":"Manan",

"10621991":"Mayank"

}

rollnum = input("Enter roll number: ")

if rollnum in students.keys():

print("Congratulations ",students[rollnum],"!",sep="")

else:

print("Congratulations Students!")

Output:



### Practical – 75

### Write a program that reads a string from the keyboard and creates dictionary containing frequency of each character occurring in the string. Also print these occurrences in the form of a histogram.

### Code:

### import matplotlib.pyplot as plt

### def letter\_frequency(user\_string):

### letters\_dict = {letter:user\_string.count(letter) for letter in user\_string if letter.isalpha()}

### # plotting the frequency table

### plt.bar(letters\_dict.keys(), letters\_dict.values())

### plt.xlabel('letters')

### plt.ylabel('occurance')

### plt.title('letters frequency given string')

### plt.show()

### letter\_frequency(input("Enter text: "))

### Output:

### 

### Practical – 76

### Create a dictionary containing names of students and marks obtained by them in three subjects. Write a program to replace the marks in three subjects with the total in three subjects, and average marks. Also report the topper of the class.

### Code:

### students = {

### "abhishek":[97,98,99],

### "amit":[78,90,67],

### "abhinav":[67,45,90]

### }

### max\_marks = 0

### name = ""

### for i in students.keys():

### marks = sum(students[i])

### if marks > max\_marks:

### max\_marks = marks

### name = i

### students[i] = [marks, marks/3]

### print("Students performance:\n",students)

### print(f"Topper is: {name} with marks: {max\_marks}")

### Output:

### 

### Practical – 77

### Given the following dictionary:

### portfolio = { 'accounts' : [ 'SBI', 'IOB'],

### 'shares' : ['HDFC', 'ICICI', 'TM', 'TCS'],

### 'ornaments' : ['10 gm gold', '1 kg silver']}

### Write a program to perform the following operations:

### Add a key to portfolio called 'MF' with values 'Reliance' and 'ABSL'.

### Set the value of 'accounts' to a list containing 'Axis' and 'BOB'.

### Sort the items in the list stored under the 'shares' key.

### Delete the list stored under 'ornaments' key.

Code:

portfolio={

'accounts':['SBI','IOB'],

'shares':['HDFC','ICICI','TM','TCS'],

'ornaments':['10 gm gold','1 kg silver']

}

portfolio['MF']=['Reliance','ABSL']

portfolio['accounts']=['Axis','BOB']

portfolio['shares'] = sorted(portfolio['shares'])

del portfolio['ornaments']

print(portfolio)

Output:



### Practical – 78

### Create two dictionaries-one containing grocery items and their prices and another containing grocery items and quantity purchased. By using the values from these two dictionaries compute the total bill.

Code:

grocery\_prices = {

"onions":78.0,

"peas":20,

"palak":15,

"carrot":30

}

grocery\_qty = {

"onions":3,

"peas":2,

"palak":5,

"carrot":3

}

bill=0

for i in grocery\_prices.keys():

bill += grocery\_prices[i] \* grocery\_qty[i]

print("Total bill: Rs.", bill)

Output:



### Practical – 79

### Create a dictionary of 10 usernames and passwords. Receive the username and password from keyboard and search for them in the dictionary. Print appropriate message on the screen based on whether a match is found or not.

Code:

users={

"user1":"helloWorld",

"user2":"privieat",

"user3":"isDollarGood",

"user4":"password",

"user5":"12345678",

"user6":"qwerty",

"user7":"09102003",

"user8":"where\_is\_mongo",

"user9":"hello",

"user10":"yes\_old+LEds1"

}

user = input("Enter username: ")

password = input("Enter password: ")

good = False

if user in users.keys():

if users[user] == password:

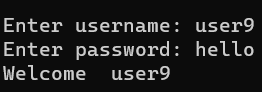
print("Welcome ", user)

good = not good

if not good:

print("Bad Credentials. Try again!")

Ouput:



### Practical – 80

### Given the following dictionary

### marks = { 'Subu' : { 'Maths' : 88, 'Eng' : 60, 'SSt' : 95 },

### 'Amol' : { 'Maths' : 78, 'Eng' : 68, 'SSt' : 89 },

### 'Raka' : { 'Maths' : 56, 'Eng' : 66, 'SSt' : 77 }}

### Write a program to perform the following operations:

### Print marks obtained by Amol in English.

### Set marks obtained by Raka in Maths to 77.

### Sort the dictionary by name.

Code:

marks = {

'Subu' : { 'Maths' : 88, 'Eng' : 60, 'SSt' : 95 },

'Amol' : { 'Maths' : 78, 'Eng' : 68, 'SSt' : 89 },

'Raka' : { 'Maths' : 56, 'Eng' : 66, 'SSt' : 77 }

}

print('Marks obtained by Amol in english:', marks['Amol']['Eng'])

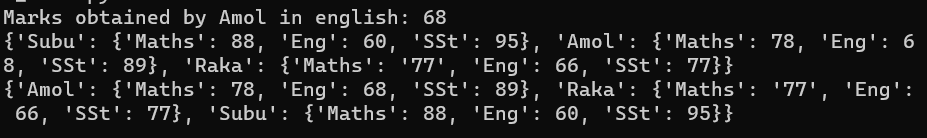
marks['Raka']['Maths'] = '77'

print(marks)

marks = dict(sorted(marks.items( )))

print(marks)

Output:



### Practical – 81

### Create a dictionary which stores the following data:

|  |  |  |
| --- | --- | --- |
| Interface | IP Address | Status |
| eth0 | 1.1.1.1 | up |
| eth1 | 2.2.2.2 | up |
| wlan0 | 3.3.3.3 | down |
| wlan1 | 4.4.4.4 | up |

### Write a program to perform the following operations:

### Find the status of a given interface.

### Find interface and IP of all interfaces which are up.

### Find the total number of interfaces.

### Add two new entries to the dictionary.

Code:

ifs = {

'eth0':{'IP' : '1.1.1.1', 'Status' : 'up'},

'eth1':{'IP' : '2.2.2.2', 'Status' : 'up'},

'wlan0':{'IP' : '3.3.3.3', 'Status' : 'down'},

'wlan1':{'IP' : '4.4.4.4', 'Status' : 'up'}

}

test = input('Enter interface: ')

print(ifs[test]['Status'])

for k, v in ifs.items( ) :

if v['Status'] == 'up' :

print(k, v['IP'])

print('Total interfaces = ', len(ifs))

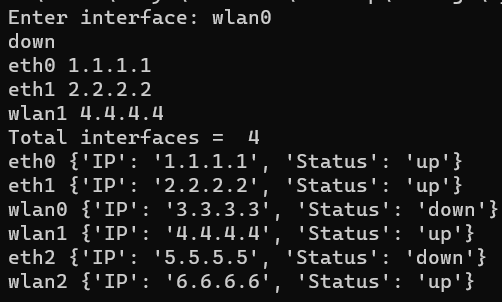
ifs['eth2'] = {'IP' : '5.5.5.5', 'Status' :'down'}

ifs['wlan2'] = {'IP' : '6.6.6.6', 'Status' : 'up'}

for k, v in ifs.items( ) :

print(k, v)

Output:



### Practical – 82

### Suppose a dictionary contains 5 key-value pairs of name and marks. Write a program to print them from last pair to first pair. Keep deleting every pair printed, such that the end of printing the dictionary falls empty.

Code:

marks = { 'Subu' : 88, 'Amol' : 78, 'Raka' : 56, 'Dinesh' : 68, 'Ranjit' : 88}

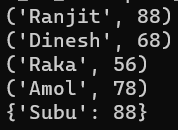
l = len(marks)-1

for i in range(l) :

print(marks.popitem( ))

print(marks)

output:



### Practical – 83

### Given the following dictionary:

### d = { 'd1': {'Fruitname' : 'Mango', 'Season' : 'Summer'}, 'd2': {'Fruitname' : 'Orange', 'Season' : 'Winter'}}

### How will you access and print Mango and Winter?

Code:

d = { 'd1': {'Fruitname' : 'Mango', 'Season' : 'Summer'}, 'd2': {'Fruitname' :

'Orange', 'Season' : 'Winter'}}

print(d['d1']['Fruitname'])

print(d['d2']['Season'])

output:



### Practical – 84

### Write a program where dataset is given as python dictionary, and program interacts with user and print output as per user need on basis of UserID.

### Code: relation = {

### "U123": {

### "rollnum": "10621019",

### "name": "Abhishek Roka",

### "city": "New Delhi"

### },

### "U124": {

### "rollnum": "10621009",

### "name": "Bunty",

### "city": "Patna"

### },

### "U125": {

### "rollnum": "10621897",

### "name": "Manan",

### "city": "NSP"

### }

### }

### while True:

### print("Options:")

### print("1. Print data for a specific user")

### print("2. Exit")

### choice = input("Enter your choice: ")

### if choice == "1":

### user\_id = input("Enter user ID (e.g., U123, U124, U125): ")

### if user\_id in relation:

### user\_data = relation[user\_id]

### print(f"User ID: {user\_id}")

### print(f"Roll Number: {user\_data['rollnum']}")

### print(f"Name: {user\_data['name']}")

### print(f"City: {user\_data['city']}")

### else:

### print("User ID not found in the dataset.")

### elif choice == "2":

### break

### else:

### print("Invalid choice. Please enter a valid option.")

### print("Program exited.")

### Output:

### 