GURU NANAK COLLEGE (AUTONOMOUS)-SHIFT-II

CHENNAI-600 042



DEPARTMENT OF INFORMATION TECHNOLOGY

B.Sc. IT

SEMESTER-V

PYTHON PROGRAMMING LAB



INDEX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | DATE | TOPIC | PAGE No. | SIGNATURE |
| 1. |  | ARMSTRONG NUMBER |  |  |
| 2. |  | FIBONACCI SEQUENCE |  |  |
| 3. |  | FACTORIAL USING RECURSIVE FUNCTION |  |  |
| 4. |  | POSITIVE NUMBERS USING RANGE |  |  |
| 5. |  | PRIME NUMBER |  |  |
| 6. |  | PALINDROME |  |  |
| 7. |  | VOWELS |  |  |
| 8. |  | CREATING A DICTIONARY FROM THE KEYWORD |  |  |
| 9. |  | BASIC OPERATIONS IN TUPLE |  |  |
| 10. |  | TOWER OF HANOI |  |  |
| 11. |  | CONVERTING LIST TO DICTIONARY |  |  |
| 12. |  | BUBBLESORT |  |  |
| 13 |  | LINEAR SEARCH |  |  |
| 14 |  | CONSTRUCTOR |  |  |
| 15. |  | DATA ABSTRACTION |  |  |
| 16. |  | EXCEPTION HANDLING |  |  |
| 17. |  | INHERITANCE |  |  |
| 18. |  | LAMBDA USING DICTIONDARY |  |  |
| 19. |  | PYTHON CODE TO DEMONSTRATE STRING LENGTH |  |  |
| 20. |  | CREATING A DICTIONARY WITH PLAYERS NAME AND SCORES |  |  |
| 21. |  | TUPLE IN LOOP |  |  |
| 22. |  | TUPLE FUNCTIONS:   1. Sum Functions 2. min and max function 3. Converting a list and a string into a tuple |  |  |

# ARMSTRONG NUMBER

Program:

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")

Input:

Enter the number: 153

Output:

153 is an Armstrong number

FIBONACCI SEQUENCE

Program:

n=int(input("enter the number"))

a = 0

b = 1

if n < 0:

print("Incorrect input")

elif n == 0:

print(a)

elif n == 1:

print(b)

else:

for i in range(0,n):

c = a + b

a = b

b = c

print(c)

Input: 6

Output: 0 1

1

2

3

5

FACTORIAL USING RECURSIVE FUNCTION

Program:

def factorial (n):

if n == 1:

return 1

else:

return n \* factorial (n-1)

n = int(input(“Enter the number to find factorial:”))

print(“Factorial:”, factorial(n))

Input:

Enter the number to find factorial: 5

Output:

Factorial: 120

POSITIVE NUMBERS USING RANGE

Program:

a=int(input("Enter the number:"))

if a > 0:

print("It is a positive number")

elif a==0:

print("It is a zero")

else:

print("It is a negative number")

Input:

Enter the number: -3

Output:

It is a negative number

# PRIME NUMBER

Program:

num=int(input("Enter the number"))

if num > 1:

for i in range(2, num//3):

if (num % i) == 0:

print(num, "is not a prime number") break

else:

print(num, "is a prime number")

Input:

Enter the number: 11

Output:

11 is a prime number

# PALINDROME

Program:

def isPalindrome(str):

t=int((len(str)/2))

for i in xrange(0,t):

if str[i] != str[len(str)-i-1]:

return False return True

s = "malayalam"

ans = isPalindrome(s)

if (ans):

print("It is a Palindrome")

else:

print("No")

Output:

Yes, It is a Palindrome

# VOWELS

Program:

def Check\_Vow(string, vowels):

final = [each for each in string if each in vowels] print(len(final))

print(final)

# Driver Code

string = "Guru Nanak College" vowels = "AeEeIiOoUu"

Check\_Vow(string, vowels);

Output:

7

[‘u’, ‘u’, ‘a’, ‘a’, ‘o’, ‘e’, ‘e’]

# CREATING A DICTIONARY FROM THE KEYWORD

Program:

x={}

print('how many element?',end =' ')

n = int(input())

for i in range (n):

print('enter the key' ,end =' ')

k=input()

print('enter its value:',end =' ')

v=int(input())

x.update({k:v})

print('the dictiondary is =',x)

Output:

How many elements?:2

Enter key:mark 1

Enter its value:35

The dictionary is:{‘mark 1’:35}

Enter key:mark 2

Enter its value:45

The dictionary is:{‘mark 1’:35,’mark2’:45}

# BASIC OPERATIONS IN TUPLE

Program:

tup1 = ('physics', 'chemistry', 1997, 2000)

tup2 = (1, 2, 3, 4, 5 )

tup3 = ( "a", "b", "c", "d" )

print (tup1)

print (tup2)

print (tup3)

# Positive Indexing

print ("tup1[0]: ", tup1[0])

# Slicing

print ("tup2[1:5]: ", tup2[1:5])

# Negative Indexing

print ("tup1[-1]:", tup1[-1])

# Concatenating tuples

tup3 = tup1 + tup2

print (tup3)

# Deleting a tuple

del tup1

print ("After deleting tup1: ",tup1)

# Existence of an element

print(1 in tup2)

print(1997 not in tup2)

# Length of a tuple

print(len(tup1))

Output:

(‘physics’,’chemistry’,1997,2000)

(1,2,3,4,5)

(‘a’,’b’,’c’,’d’)

Tup1[0]: physics

Tup2[1:5]: (2,3,4,5)

Tup1[-1]: 2000

(‘physics’,’chemistry’,1997,2000,1 ,2 ,3 ,4 ,5)

After deleting tup1:

True

True

5

# TOWER OF HANOI

Program:

# Creating a recursive function

def tower\_of\_hanoi(disks, source, auxiliary, target):

if(disks == 1):

print('Move disk 1 from rod {} to rod {}.'.format(source, target))

return

# function call itself

tower\_of\_hanoi(disks - 1, source, target, auxiliary)

print('Move disk {} from rod {} to rod {}.'.format(disks, source, target))

tower\_of\_hanoi(disks - 1, auxiliary, source, target)

disks = int(input('Enter the number of disks: '))

# We are referring source as A, auxiliary as B, and target as C

tower\_of\_hanoi(disks, 'A', 'B', 'C') # Calling the function

Output:

Enter the number of disks: 3

Move disk 1 from rod A to rod C.

Move disk 2 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 3 from rod A to rod C.

Move disk 1 from rod B to rod A.

Move disk 2 from rod B to rod C.

Move disk 1 from rod A to rod C.

CONVERTING LIST TO DICTIONARY

Program:

countries = ["USA","INDIA","GERMANY","FRANCE"]

cities = ['WASHINGTON','NEWDELHI','BERLIN','PARIS']

z = zip(counties,cities)

d = dict(z)

print('{:15s}— {:15s}'.format('country','captial'))

for k in d

print('{:15s}--{:15s}'.format(k,d[k]))

Output:

Country -- Captial

USA -- WASHINGTON

INDIA -- NEWDELHI

GERMANY -- BERLIN

FRANCE -- PARIS

# BUBBLE SORT

Program:

def bubblesort(arr):

n = len(arr) arr=[64,34,25,72,22,11,90]

for i in range(len(arr)):

for j in range(0,len(arr)-i-1):

if arr[j]>arr[j+1]:

arr[j],arr[j+1]=arr[j+1],arr[j]

bubblesort(arr)

print("sorted array is :")

for i in range(len(arr)):

print("%d"%arr[i])

Output:

sorted array is : 11

22

25

34

64

72

90

LINEAR SEARCH

Program:

def search(list,n):

for i in range(len(list)):

if list[i] == n:

return True

return False

# list which contains both string and numbers.

list = [1, 2, 'sachin', 4,'Guru', 6]

# Driver Code

n = 'Guru'

if search(list, n):

print("Found")

else:

print("Not Found")

Output: Found

CONSTRUCTOR

Program:

class Employee:

def \_\_init\_\_(self,name,ids):

self.ids = ids self.name = name

def display (self):

print(‘ID: {ids}\n Name: {name}’.format(ids = self.ids,name = self.name))

emp1 = employee(“John”,101)

emp2 = employee(“David”,102)

emp1.display()

emp2.display()

Output: ID: 101

NAME: JOHN

ID: 102

NAME: DAVID

# DATA ABSTRACTION

Program:

from abc import ABC, abstractmethod

class Car(ABC):

def mileage(self):

pass

class Tesla(Car):

def mileage(self):

print("The mileage is 30kmph")

class Suzuki(Car):

def mileage(self):

print("The mileage is 25kmph ")

class Duster(Car):

def mileage(self):

print("The mileage is 24kmph ")

class Renault(Car):

def mileage(self):

print("The mileage is 27kmph ")

# Driver code

t= Tesla ()

t.mileage()

r = Renault()

r.mileage()

s = Suzuki()

s.mileage()

d = Duster()

d.mileage()

Output:

The mileage is 30kmph

The mileage is 27kmph

The mileage is 25kmph

The mileage is 24kmph

EXCEPTION HANDLING

Program:

# import module sys to get the type of exception

import sys

randomList = ['a', 0, 2]

for entry in randomList:

try:

print("The entry is", entry)

r = 1/int(entry)

break

except:

print("Oops!",sys.exc\_info()[0],"occured.")

print("Next entry.")

print()

print("The reciprocal of",entry,"is",r)

Output:

The entry is a

OOPS ! <class’value error’>occurred.

Next entry.

The entry is 0

OOPS! <class’zerodivisionerror’>occurred.

Next entry

The entry is 2

The reciprocal of 2 is 0.5

# INHERITANCE

Program:

class Animal:

def speak(self):

print("Animal Speaking")

class Dog(Animal):

def bark(self):

print("dog barking")

d = Dog()

d.bark()

d.speak()

Output:

Dog barking

Animal speaking

# LAMBDA USING DICTIONARY

Program:

colors = {"red":10,"green":35,"blue":15,"white":25}

c1 = sorted(colors.items(),key = lambda t:t[0])

print(c1)

c2 = sorted(colors.items(),key = lambda t:t[1])

print(c2)

Output:

[('blue', 15), ('green', 35), ('red', 10), ('white', 25)]

[('red', 10), ('blue', 15), ('white', 25), ('green', 35)]

PYTHON CODE TO DEMONSTRATE STRING LENGTH

Program:

# using for loop

# Returns length of string def findLen(str):

counter = 0

for i in str:

counter += 1 return counter

str = "gurunanakcollege"

print(findLen(str))

Output:

16

CREATING A DICTIONARY WITH

PLAYERS NAME AND SCORES

Program:

print("how many players ? ",end = ' ')

n = int(input())

for i in range(n):

print("enter player name",end = '') k = input()

print("enter runs:",end = '') v = int(input())

x.update({k:v})

print("\n players in this match:")

for name in x.keys():

print(name)

print("enter the players name:",end = '')

name = input()

runs= x.get(name, -1)

if(runs == -1):

print("players not found")

else:

print("{}made runs {}.".format(name,runs))

Output:

how many players ? 3

enter player name arun

enter runs:50

enter player name deepak enter runs:60

enter player name ram

enter runs:70

players in this match:

arun deepak ram

enter the players name: ram ram made runs 70.

TUPLE IN LOOP

Program:

tup = ('geek',)

n = 5

#Number of time loop runs for i in range(int(n)):

tup = (tup,)

print(tup)

# Code to create a tuple with repetition

tuple3 = ('python',)\*3

print(tuple3)

# Code for creating nested tuples

tuple1 = (0, 1, 2, 3)

tuple2 = ('python', 'geek') tuple3 = (tuple1, tuple2) print(tuple3)

Output:

(‘geek’,)

((‘geek’,),)

(((‘geek’,),),)

((((‘geek’,),),),)

(((((‘geek’,),),),),)

(‘python’,’python’,’python’)

((0,1,2,3),(‘python’,’geek’))

TUPLE FUNCTIONS

Program:

#sum functions

x = (10, 20, 30, 40, 50, 60, 70, 80, 90)

print("Tuple : ", x)

total = sum(x)

print("The Sum of Integer Tuple is : ", total)

numbers = (9, -5, 7, 0, 24, -1, 2, 10)

print("\nNumbers Tuple : ", numbers)

print("The Sum of a Numbers Tuple is : ", sum(numbers))

# Python tuple min and max function

fruits = ('Banana', 'Orange', 'Blackberry', 'Apple', 'Kiwi',

'Grape')

print("Tuple : ", fruits)

minimum = min(fruits)

print("The Minimum Value in Fruits Tuple is : ", minimum)

numbers = (9, -5, 7, 0, 24, -1, 2, 10)

print("\nNumbers Tuple : ", numbers)

print("The Minimum Value Numbers Tuple is : ", min(numbers))

maximum = max(fruits)

print("The Maximum Value in Fruits Tuple is : ", maximum)

numbers = (9, -5, 7, 0, 24, -1, 2, 10)

print("\nNumbers Tuple : ", numbers)

print("The Maximum Value Numbers Tuple is : ",

max(numbers))

# Code for converting a list and a string into a tuple list1 = [0, 1, 2]

print(tuple(list1))

print(tuple('python')) # string 'python'

Output:

Tuple : (10, 20,30,40,50,60,70,80,90)

The sum of integer tuple is : 450

Numbers tuple : (9,-5,7,0,24,-1,2,10)

The sum of a number tuple is : 46

Tuple : ('Banana', 'Orange', 'Blackberry', 'Apple',

'Kiwi',’Grape’)

The minimum value in fruits tuple is : Apple

Numbers tuple : (9,-5,7,0,24,-1,2,10)

The minimum value number tuple is : -5

The maximum value in fruits tuple is : orange

Numbers tuple : (9,-5,7,0,24,-1,2,10)

The maximum value number tuple is : 24

(0,1,2)

(‘p’,’y’,’t’,’h’,’o’,’n’)