**Exercise-1a**

**AIM :**Write a python script to read two integer numbers and perform arithmetic operations.

**SOURCE CODE :**

a=int(input("enter the value of a"))

b=int(input("enter the value of b"))

print("addition is:",a+b)

print("subtraction is:",a-b)

print("multiplication is:",a\*b)

print("division is:",a/b)

print("modulus is:",a%b)

print("floor division is:",a//b)

print("expponentiationis:",a\*\*b)

""" INPUT :

enter the value of a 4

enter the value of b 2

**OUTPUT:**

addition is: 6

subtraction is: 2

multiplication is: 8

division is: 2.0

modulus is: 0

floor division is: 2

expponentiation is: 16

**Exercis-1b**

**AIM** :Write a python script to evaluate following expressions by considering necessary inputs.

i) ax2 + bx + c ii) ax5 + bx3 + c iii) (ax + b) /(ax – b) iv) x – a / b + c """

**SOURCE CODE:**

a=int(input("enter the value of a"))

b=int(input("enter the value of b"))

c=int(input("enter the value of c"))

x=int(input("enter the value of x"))

p=a\*x\*\*2+b\*x+c

q=a\*x\*\*5+b\*x\*\*3+c

r=(a\*x+b)/(a\*x-b)

s=(x-a)/(b+c)

print("the value of p:",p)

print("the value of q:",q)

print("the value of r:",r)

print("the value of s:",s)

""" INPUT :

enter the value of a 1

enter the value of b 2

enter the value of c 2

enter the value of x 3

OUTPUT :

the value of p: 17

the value of q: 299

the value of r: 5.0

the value of s: 0.5

**Exercise-2a**

**Aim:**Write a python script to convert a given decimal number into octal, hexadecimal and binary.

**Sourcecode:**

a=int(input("enter the decimal number"))

print("the decimal number in octal is :",oct(a))

print("the decimal number in hexa decimal is :",hex(a))

print("the decimal number in binary is :",bin(a))

""" INTPUT :

enter the decimal number 344

**OUTPUT** :

the decimal number in octal is : 0o530

the decimal number in hexa decimal is : 0x158

the decimal number in binary is : 0b101011000

**Exercise-2b**

**Aim:** Write a python script to read four integer values separated with commas and display the sum of those four numbers

**SOURCE CODE :**

a=list(input("enter the four integers").split(","))

sum=0

for x in range(0,4):

sum+=int(a[x])

print("the sum of four integers is :",sum)

"""INPUT :

enter the four integers 1,2,3,4

**OUTPUT** :

the sum of four integers is : 10

**Exercise-2c**

**Aim:**Write a python script to print ―SVEC‖ with prefix of ten spaces by using format().

**SourceCode:**

txt='{:>10}SVEC'.format('')

print(txt)

**OUTPUT** :

SVEC

**Exercise 3a**

**Aim**: Write a python script to calculate electricity bill based on following slab

rates.

Consumption units Rate (in Rupees/Unit)

0-100 4

101-150 4.6

151-200 5.2

201-300 6.3

Above 300 8

(Hint: To get Consumption units take current meter reading and old meter

reading from the user as input """

**source code**:

cunit=int(input("enter the currrent meter reading units "))

ounit=int(input("enter the old meter reading units "))

cspunits=cunit-ounit

print("the consumption units are :",cspunits)

if cspunits>=0 and cspunits<=100:

electricitybill=cspunits\*4

elifcspunits>=101 and cspunits<=150:

electricitybill=cspunits\*4.6

elifcspunits>=151 and cspunits<=200:

electricitybill=cspunits\*5.2

elifcspunits>=201 and cspunits<=300:

electricitybill=cspunits\*6.3

else:

electricitybill=csunits\*8;

print("The electricity bill is:",electricitybill)

"""INTPUT :

enter the currrent meter reading units : 500

enter the old meter reading units : 350

**OUTPUT** :

the consumption units are : 150

The electricity bill is: 690.0

**Exercise-3b**

**Aim:** Print the following pattern using python script.

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

1 2 3 4 5 4 3 2 1 """

**SOURCE CODE**:

n=5

for i in range(n):

p=1

for j in range(i,n):

print(' ',end=' ')

for j in range(i):

print(p,end=' ')

p+=1

for j in range(i+1):

print(p,end=' ')

p-=1

print()

**OUTPUT** :

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

1 2 3 4 5 4 3 2 1

**Exercise-4a**

**Aim:**Write a python script to check whether the given number (N) can be expressed as Power of Two (2) or not. For example, 8 can be expressed as 2^3.

**Sourcecode:**

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

while(n!=1):

if n%2!=0:

print("cannot be expresses as power of two")

break

n=n//2

else:

print("can be expressed as powerOfTwo")

**Output:**

enter n 45

45

cannot be expresses as power of two

**Exercise 4b**

**Aim:**b) Writeapython scripttoperformfollowingsetsoperations. i)update () ii)discard() iii)issuperset() iv)isdisjoint() v) symmetric\_difference ()

**Sourcecode:**

A = {'a', 'b'}

B = {1, 2, 3}

A.update(B)

print(A)

fruits = {"apple", "banana", "cherry"}

fruits.discard("banana")

print(fruits)

x = {"f", "e", "d", "c", "b", "a"}

y = {"a", "b", "c"}

z = x.issuperset(y)

print(z)

s1 = {1, 2, 3}

s2 = {4, 5, 6}

print(s1.isdisjoint(s2))

p = {"apple", "banana", "cherry"}

q = {"google", "microsoft", "apple"}

r = p.symmetric\_difference(q)

print(r)

**OUTPUT :**

{1, 2, 3, 'b', 'a'}

{'apple', 'cherry'}

True

True

{'microsoft', 'cherry', 'banana', 'google'}

Open with

Exercise

**Exercise 5A**

**Aim:**

Given an integer tuple, for each element in the tuple check whether there exists a smaller element on the next immediate position of the tuple. If it exists print the smaller element. If there is no smaller element on the immediate next to the element then print -1. Example: Input: 4 2 1 5 3 Output: 2 1 -1 3 -1

**Source code:**

n=input("enter spaced integers:")

a=tuple(int(num) for num in n.split())

b=list(a)

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

if(b[i]>b[i+1]):

b[i]=b[i+1]

else:

b[i]=-1

b[len(a)-1]=-1

a=tuple(b)

print(a)

**Output:**

enter spaced integers:5 7 1 7

(-1, 1, -1, -1)

**Exercise-5b**

**Aim:**

Write a Python program to print the number of times the string ―SVEC‖ can be formed for the given input string. Example: Input: acdfksekfevskdjfs Output: 1

**Sourcecode:**

s='svecshgdgjhhkjll'

y="svec"

yc=[]

for i in y:

yc.append(s.count(i))

print(min(yc))

**output:**

1

**Exercise 6a**

**Aim:**Write a python script to read details of N students – name, roll number, branch and age. Sort the student details based on their names and display.

**Sourcecode:**

n=int(input("enter how many student details you need to store: "))

print(n)

list=[]

for i in range(0,n):

list.append([])

name=input("enter name: ")

list[i].append(name)

roll=input("enter id: ")

list[i].append(roll)

branch=input("enter branch: ")

list[i].append(branch)

age=input("enter age: ")

list[i].append(age)

list.sort()

for i in range(0,n):

for j in range(4):

print(list[i][j],end=" ")

print()

**Output:**

enter how many student details you need to store: 2

2

enter name: bela

enter id: 23

enter branch: cse

enter age: 19

enter name: bani

enter id: 25

enter branch: cse

enter age: 20

bani 25 cse 20

sathvika 23 cse 19

**Exercise-6b**

**Aim:**Write a python script to delete duplicate strings from a list of strings. (Insertion order should be maintained after deleting duplicate string)

**Sourcecode:**

lst=[x for x in input("Enter strings separated by, \n").split(',')]

l=len(lst)

for i in lst:

cnt=0

for j in lst:

if i==j:

cnt+=1

if(cnt>1):

lst.remove(i)

lst.sort()

print("after removing : \n",lst)

**output:**

Enter strings separated by,

good,bad ,good

after removing :

['bad ', 'good']

**Exercise-6c**

**Aim:**Write a python script to read details of N students into nested list and convert it into a nested dictionary.

**Sourcecode:**

list=[]

n=int(input("enter n: "))

for i in range(0,n):

list.append([])

name=input("enter name: ")

list[i].append(name)

roll=input("enter roll")

list[i].append(roll)

branch=input("enter branch ")

list[i].append(branch)

print(list)

d={}

for i in range(0,n):

j=0

d[i+1]={'name':list[i-1][j],'roll':list[i-1][j+1],'branch':list[i-1][j+2]}

print(d)

**output:**

enter n: 3

enter name: bela

enter roll 23

enter branch cse

enter name: bani

enter roll 34

enter branch cse

enter name: pratha

enter roll 23

enter branch cse

[['bela', '23', 'cse'], ['bani', '34', 'cse'], ['swetha', '23', 'cse']]

{1: {'name': 'pratha', 'roll': '23', 'branch': 'cse'}, 2: {'name': 'bela', 'roll': '23', 'branch': 'cse'}, 3: {'name': 'bani', 'roll': '34', 'branch': 'cse'}}

**Exercise-7a**

**Aim:**Design a function that can perform sum of two or three or four numbers.

**Sourcecode:**

def add(a=0,b=0,c=0,d=0):

print(a+b+c+d)

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

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

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

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

add(a,b,c)

add(a,b,c,d)

**Output:**

Enter a:2

Enter b:3

Enter c:4

Enter d:4

9

13

**Exercise-7b**

**Aim:**

Write a python script to implement Towers of Hanoi problem

**Sourcecode:**

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

if(disks == 1):

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

return

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: '))

tower\_of\_hanoi(disks, 'A', 'B', 'C')

**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.

**Exercise-7c**

**Aim:**Write a pythonscriptto print the numbers that do not appear in the Fibonacci series of n numbers where n is given by the user. (Hint: If n is 8 then up to 8 Fibonacci numbers has to be printed Ex: 1 1 2 3 5 8 13 21 and in this series missing numbers should be traced and printed, Ex: missing numbers are: 4 6 7 9 10 11 12 14 15 16 17 18 19.

**Source code:**

n=int(input("enter limit: "))

a=0

b=1

c=0

list=[0,1]

for i in range(0,n-2):

c=a+b

list.append(c)

a=b

b=c

print(list)

m=list[n-1]

print("missing numbers")

for i in range(0,m):

if i not in list:

print(i,end=' ')

**output:**

enter limit: 10

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

missing numbers

4 6 7 9 10 11 12 14 15 16 17 18 19 20 22 23 24 25 26 27 28 29 30 31 32 33

**Exercise-8a**

**Aim:**Write a python script to create a regular expression to extract all words with 5 characters length from a given string.

**Source code:**

import re

text = input("enter string: ")

print(re.findall(r"\b\w{5}\b", text))

**output:**

enter string: geeks for geeks

['geeks', 'geeks']

**EXERCISE-8B**

**Aim:**Write a python script to create a regular expression to extract the phone number from a given text.

**Sourc code:**

a=input('enter a sentence:')

s=a.split()

for x in s:

if len(x)==10:

if x.isnumeric():

print(x,end=' ')

**or**

import re

str = input("enter any string containing phonenumber:\n")

x = re.findall('[\d]{10}', str)

print(x)

**output:**

Bindu 878787777 from Andhrapradesh

['878787777']

**Exercise-9a**

**Aim:** Write a python script to copy the content of one file into another file

**Source code:**

# open both files

with open('first.txt','r') as firstfile, open('second.txt','a') as secondfile:

for line in firstfile:

secondfile.write(line)

**or**

sfile=input("Enter Source File:")

try:

sf=open(sfile,"rb")

tfile = input("Enter Target File:")

tf=open(tfile,"wb")

tf.write(sf.read())

sf.close()

tf.close()

print("File Copied...")

except FileNotFoundError as e:

print(e)

**output:**

enter Source File:first.txt

Enter Target File:second.txt

File Copied...

**Exercise-9b**

**Aim:**Write a python script to read all the strings from the text file and display them.

**Sourcecode**:

with open('First.txt','r') as file:

for line in file:

for word in line.split():

print(word)

**or**

a=input('enter file name:')

try:

f1=open(a,'r')

print(f1.read())

except FileNotFoundError:

print('file not found')

**output:**

bindu

cse-ds

**Exercise-10a**

**Aim:**Write a Python program for the following problem: Create a base class Basic\_Info with data members name, rollno, gender and two member functions getdata() and display(). Derive a class Physical\_Fit from Basic\_Info which has data members height and weight and member functions getdata() and display(). Display all the information using object of derived class.

**Sourcecode:**

class Basic\_Info:

def getdata(self):

self.name=input("enter name: ")

self.rno=int(input("enter roll num: "))

self.gender=input("enter gender: ")

def putdata(self):

print("name: ",self.name)

print("roll: ",self.rno)

print("gender: ",self.gender)

class Physical\_Fit:

def getdata(self):

Basic\_Info.getdata(self)

self.height=int(input("enter height: "))

self.weight=int(input("enter weight: "))

def putdata(self):

Basic\_Info.putdata(self)

print("height: ",self.height)

print("weight: ",self.weight)

p1=Physical\_Fit()

p1.getdata()

p1.putdata()

**'''output:**

enter name: bani

enter roll num: 23

enter gender: Female

enter height: 5

enter weight: 56

name: bani

roll: 23

gender: Female

height: 5

weight: 56'''

**Exercise-10b**

**Aim:**

Define a class REPORT with the following specification: Private members: Adno: 4-digit admission number SVEC20 – B.TECH – COMPUTER SCIENCE AND ENGINEERING(DATA SCIENCE) 99 Name:20 characters Marks: A list of 5 floating point values Average: average marks obtained GETAVG () a function to compute the average obtained in five subjects. Public members: READINFO () function to accept values for adno, name, marks. Invoke the function GETAVG (). DISPLAYINFO () function to display all data members of report on the screen. You should give function definitions. Write driver code to demonstrate all the functions.

**Sourcecode:**

class REPORT:

def BASICINFO(self):

while True:

self.\_\_name = input('Enter Name: ')

if len(self.\_\_name) > 20:

print("Invalid!!! permitts only 20 characters")

continue

else:

break

self.\_\_admno=int(input("enter 4 digit number:"))

assert(self.\_admno>=1000 and self.\_admno<10000),"admission number is of 4 digits "

self.\_\_marks=[]

for i in range(0,5):

self.\_\_marks.append(int(input("enter marks:")))

print(self.\_\_marks)

def \_\_GETAVG(self):

self.sum=0;self.\_\_avg=0

for i in self.\_\_marks:

self.sum += i

self.\_\_avg=self.sum/5

def DISPLAYINFO(self):

s.\_\_GETAVG()

print("\*\*Displaying the details\*\*")

print("Name:",self.\_name,"Adm no.:",self.admno,"Marks:",self.marks, "Average:",self.\_avg)

s=REPORT()

s.BASICINFO()

s.\_\_GETAVG()

s.DISPLAYINFO()

**output:**

Enter Name: bela

enter 4 digit number:2313

enter marks:20

enter marks:20

enter marks:20

enter marks:20

enter marks:20

[20, 20, 20, 20, 20]

Traceback (most recent call last):

File "<string>", line 27, in <module>

AttributeError: 'REPORT' object has no attribute '\_\_GETAVG'

>dir(s)

s.REPORT\_GETAVG()

>s.DISPLAYINFO()

\*\*Displaying the details\*\*

Name: belaAdm no.: 2313 Marks: [20, 20, 20, 20, 20] Average: 20.0