“””WRITE A PYTHON PROGRAM TO DEMONSTRATE THE WORKING OF PRINT TO DISPLAY VARIOUS MESSAGES”””

print('HELLO WORLD') print("HELLO WORLD") print('HELLO WORLD','WELCOME')

print("it's not a C language")

print('welcome to ece','welcome to cse','welcome to eee','welcome to it') print('welcome to ece\n','welcome to cse\n','welcome to eee\n','welcome to it') print('welcome to ece','welcome to cse','welcome to eee','welcome to it',sep='\n') print('welcome to ece','welcome to cse','welcome to eee','welcome to it',sep=',',end='\*') print('\nwelcome to \

aits \ rajampet') a=5 print('a=',a)

print('a=%d'%a) print('a={}'.format(a))

myfile=open('ece.txt','w')

print('welcome to ece branch',file=myfile) myfile.close()

print('welcome') print() print('hello')

""" OUTPUT:

HELLO WORLD HELLO WORLD

HELLO WORLD WELCOME

it's not a C language

welcome to ece welcome to cse welcome to eee welcome to it welcome to ece

welcome to cse welcome to eee welcome to it welcome to ece welcome to cse welcome to eee welcome to it

welcome to ece,welcome to cse,welcome to eee,welcome to it\* welcome to aits rajampet

a= 5 a=5 a=5

welcome

hello """

#write a python program to demonstrate literals,varaibles, identifiers and data types a=0B1010 #binary integer literal

b=0b0101 #binary integer literal c=123 #decimal integer literal d=0o12 #octal integer literal e=0XA #hexa decimal integer literal f=2.5638978965 #float literal comp=2+3.5j #complex literal str1='hello welcome' #string literal str2="hello welcome" #string literal str3='''welcome to

aits

rajampet''' #string literal ch='E' #character literal x=True+5 # boolean literal y=False+6 # boolean literal

print(a,b,c,d,e,str1,str2,str3,ch,format(f,'.2f'),comp,comp.real,comp.imag,sep='\n') print(x,y)

l=[1,'ECE',2.5] #list literal print(l)

t=(1,'ece',2.5) #tuple literal print(t)

d={1:'ece',2:'cse',3:'eee'} #dictionary literal print(d)

s={1,2,3} #set literal print(s)

num=25 a=b=c=50 x,y,z=1,'ece',2.5 name='aits' my\_tax=0.08

print(num,a,b,c,x,y,z,name,my\_tax) print(num)

print(id(num)) k=20

print(k) print(id(k)) print(type(num)) print(type(name))

""" OUTPUT:

10

5

123

10

10

hello welcome hello welcome welcome to aits

rajampet E

2.56 (2+3.5j)

2.0

3.5

6 6

[1, 'ECE', 2.5]

(1, 'ece', 2.5)

{1: 'ece', 2: 'cse', 3: 'eee'}

{1, 2, 3}

25 50 50 50 1 ece 2.5 aits 0.08

25

1450043696

20

1450043616

<class 'int'>

<class 'str'> """

#WRITE A PYTON PROGRAM TO DEMONSTRATE THE WORKING OF OPERATORS

#arithmetic operators a=5

b=2

print('<====arithemtic operators====>',end='\n') print('a+b:',a+b)

print('a-b:',a-b)

print('a/b:',a/b)

print('a\*b:',a\*b)

print('a%b:',a%b)

print('a//b:',a//b)

print('a\*\*b:',a\*\*b)

#logical operators a=(1>5)

b=(2<5)

print('<====logical operators====>',end='\n') print('a and b:',a and b)

print('a or b:',a or b)

print('not a:',not a)

#Bitwise operators a=5

b=2

print('<====Bitwise operators====>',end='\n') print('a&b:',a&b)

print('a|b:',a|b)

print('~b:',~b)

print('a^b:',a^b)

print('a<<b:',a<<b)

print('a>>b:',a>>b)

#Comparision operators a=5

b=2

print('<====Comparision operators====>',end='\n') print('a==b:',a==b)

print('a!=b:',a!=b)

print('a<b:',a<b)

print('a>b:',a>b)

print('a>=b:',a>=b)

print('a<=b:',a<=b)

#Assignment Operators

print('<====Assignment operators====>',end='\n') a=5

print('a=:',a) a&=3 print('a&=3:',a) a|=3 print('a|=3:',a) a^=3 print('a^=3:',a) a<<=3

print('a<<=3:',a)

a>>=3

print('a>>=3:',a) a=6

a+=3

print('a+=3:',a) a-=3

print('a-=3:',a) a\*=3 print('a\*=3:',a) a/=3 print('a/=3:',a) a//=3 print('a//=3:',a) a\*\*=3 print('a\*\*=3:',a)

#Identity operators l1=['ece','cse']

l2=['ece','cse'] l3=l1

print('<====Identity operators====>',end='\n') print('l1 is l3:',l1 is l3)

print('l1 is l2:',l1 is l2) print('l1 is not l2:',l1 is not l2) print('l1 == l2:',l1 == l2) #Membership operators l1=['ece','cse']

l2=['ece','cse'] l3=l1

print('<====Membership operators====>',end='\n') print('ece in l3:','ece' in l3)

print('eee in l2:','cse' in l2) print('eee not in l2','eee' not in l2) """

OUTPUT:

<====arithemtic operators====> a+b: 7

a-b: 3

a/b: 2.5

a\*b: 10

a%b: 1

a//b: 2

a\*\*b: 25

<====logical operators====> a and b: False

a or b: True not a: True

<====Bitwise operators====> a&b: 0

a|b: 7

~b: -3

a^b: 7

a<<b: 20

a>>b: 1

<====Comparision operators====> a==b: False

a!=b: True a<b: False a>b: True a>=b: True a<=b: False

<====Assignment operators====> a=: 5

a&=3: 1

a|=3: 3

a^=3: 0

a<<=3: 0

a>>=3: 0

a+=3: 9

a-=3: 6

a\*=3: 18

a/=3: 6.0

a//=3: 2.0

a\*\*=3: 8.0

<====Identity operators====> l1 is l3: True

l1 is l2: False

l1 is not l2: True l1 == l2: True

<====Membership operators====> ece in l3: True

eee in l2: True eee not in l2 True """

#write a python program to demonstrate the working of for loop. #initailization

esum=osum=0 numbers=[1,2,3,4,5,6,7,8,9,10]

for i in numbers: if i%2==0:

esum+=i else:

osum+=i

print('Summation of even numbers:',esum) print('Summation of add numbers:',osum) """

OUTPUT:

Summation of even numbers: 30 Summation of add numbers: 25 """

#WRITE A PYTHON PROGRAM TO DEMONSTRATE THE WORKING OF WHILE LOOP

import random while True:

input("Press enter to roll the dice") num = random.randint(1,6) print("You got",num)

option = input("Roll again?(y/n) ") if option == 'n':

break

print('Rolling Over.Thank You') """

OUTPUT:

Press enter to roll the dice You got 1

Roll again?(y/n) y

Press enter to roll the dice You got 1

Roll again?(y/n) y

Press enter to roll the dice You got 5

Roll again?(y/n) y

Press enter to roll the dice You got 3

Roll again?(y/n) y

Press enter to roll the dice You got 3

Roll again?(y/n) y

Press enter to roll the dice You got 6

Roll again?(y/n) y

Press enter to roll the dice You got 5

Roll again?(y/n) n Rolling Over.Thank You

#Write a C program to input amount from user and print minimum number of notes required for the amount.

n500=n100=n50=n20=n10=n5=n2=n1=0

amount=int(input('Enter amount:')) if(amount>=500):

n500=amount//500 amount-=n500\*500

if(amount>=100): n100=amount//100 amount-=n100\*100

if(amount>=50): n50=amount//50 amount-=n50\*50

if(amount>=20): n20=amount//20 amount-=n20\*20

if(amount>=10): n10=amount//10 amount-=n10\*10

if(amount>=5): n5=amount//5 amount-=n5\*5

if(amount>=2): n2=amount//2 amount-=n2\*2

if(amount>=1): n1=amount

print('<===Total No.of notes:====>') print('500=',n500) print('100=',n100)

print('50=',n50)

print('20=',n20)

print('10=',n10)

print('5=',n5)

print('2=',n2)

print('1=',n1) """

OUTPUT:

Enter amount:1226

<===Total No.of notes:====> 500= 2

100= 2

50= 0

20= 1

10= 0

5= 1

2= 0

1= 1 """

#write a python program to display sum of squares for the first n natural numbers num=int(input('Enter Your number:'))

sum=0

for i in range(num+1): sum+=i\*\*2

print('Sum of Squares for the first',num,'numbers is:',sum) """

OUTPUT:

Enter Your number:5

Sum of Squares for the first 5 numbers is: 55 """

#write a python program to display reverse of a number num=int(input('Enter Your Number:'))

original=num reverse=0

while original!=0:

rem=original%10; reverse=reverse\*10+rem original//=10

print('Reverse of a',num,'is:',reverse) """

OUTPUT:

Enter Your Number:123456 Reverse of a 123456 is: 654321 """

#write a python program to display N no.of Fibonacci numbers n1,n2=0,1

count=0

number=int(input('Enter No.of fibonacci numbers to be display:')) if number<0:

print('please enter positive numbers only ')

elif number==1:

print('The fibonacci numbers upto',number) print(n1)

else:

print('The first',number,'Fibonacci numbers') while count<number:

print(n1) n3=n1+n2 n1=n2 n2=n3 count+=1

""" OUTPUT:

Enter No.of fibonacci numbers to be display:8 The first 8 fibonacci numbers

0

1

1

2

3

5

8

13 """

#WRITE A PYTHON PROGRAM TO FIND PRIME NUMBERS BETWEEN THE RANGE

start=int(input('Enter starting range:')) end=int(input('Enter Ending range:'))

print('The Prime Numbers between',start,'and',end,'are') for num in range(start,end+1):

if num>1:

for i in range(2,num): if(num%i==0):

break

"""

else:

print(num)

OUTPUT:

Enter starting range:1 Enter Ending range:20

The Prime Numbers between 1 and 20 are 2

3

5

7

11

13

17

19

#WRITE A PYTHON PROGRAM TO DEMONSTRATE WORKING OF LIST IN PYTHON AND IT'S OPERATIONS

mylist=[1,2,3,4] new=[10,11]

newlist=[15,12,13,12] #length

print('length of mylist is:',len(mylist)) #accessing

print('The first element is:',mylist[0]) print('The second element is:',mylist[1]) print('The third element is:',mylist[2]) print('The fourth element is:',mylist[3]) #slicing

print('mylist[0:3]',mylist[0:3])

print('mylist[:2]',mylist[:2])

print('mylist[1:]',mylist[1:])

print('mylist[-3:-1]',mylist[-3:-1])

print('mylist[::-1]',mylist[::-1]) #append

mylist.append(5)

print('After append,the list is:',mylist) #insert

mylist.insert(1,6)

print('After insert,the list is:',mylist) mylist.append(new)

print('After append,the list is:',mylist) #extend

mylist.extend(newlist)

print('After extend,the list is:',mylist) #accessing

print('The sixth element is:',mylist[6]) print('The seventh element is:',mylist[7]) print('The eight element is:',mylist[8])

print('The sixth of zero index element is:',mylist[6][0]) #remove

newlist.remove(12)

print('after remove ,the list is:',newlist) #del

del newlist[2]

print('after delete ,the list is:',newlist) #pop

print('before pop() ,the list is:',mylist) mylist.pop()

print('after pop() ,the list is:',mylist) print('before pop(6) ,the list is:',mylist) mylist.pop(6)

print('after pop(6) ,the list is:',mylist) #clear

print('before clear() ,the list is:',new) new.clear()

print('after clear() ,the list is:',new) #count

l1=[1,2,2,3,3,4,2]

print('count(2) is:',l1.count(2)) #index

print('index(2) is:',l1.index(2)) #sort

mylist.sort() print('Sorting is:',mylist) mylist.sort(reverse=True) print('Sorting is:',mylist) #reverse mylist.reverse()

print('Reverse list is:',mylist) #copy and assigning l1=[1,2,3]

l2=l1 print(l1,l2) l1[0]=4

print(l1,l2) l2=l1.copy() print(l1,l2) l1[0]=5

print(l1,l2) l2=list(l1) print(l1,l2) l1[0]=6

print(l1,l2) #max,min & sum

print('Max of mylist is:',max(mylist)) print('Min of mylist is:',min(mylist)) print('sum of mylist is:',sum(mylist)) #all,any

val1=[1,2,3,4,0]

print('all:',all(val1))

print('any:',any(val1))

val2=[1,2,3,4,5]

print('all:',all(val2))

print('any:',any(val2)) val3=[None,None,None,False] print('all:',all(val3))

print('any:',any(val3)) #enumerate branch=['ece','eee','cse'] obj=enumerate(branch) print('Return Type:',type(obj)) for (i,j) in enumerate(branch): print('The',i,'element is:',j) #ord,chr

name='AITS' n=[]

m=[]

for i in name: n.append(ord(i)) for j in n:

m.append(chr(j)) print(n,m)

#comparison l1=[1,2,4,5]

l2=[1,2,5,8]

l3=[1,2,5,8,10]

l4=[1,2,4,5]

print(l2>l1) print(l2<l3) print(l1==l4) '''

OUTPUT

length of mylist is: 4 The first element is: 1 The second element is: 2 The third element is: 3 The fourth element is: 4 mylist[0:3] [1, 2, 3]

mylist[:2] [1, 2]

mylist[1:] [2, 3, 4]

mylist[-3:-1] [2, 3]

mylist[::-1] [4, 3, 2, 1]

After append,the list is: [1, 2, 3, 4, 5]

After insert,the list is: [1, 6, 2, 3, 4, 5]

After append,the list is: [1, 6, 2, 3, 4, 5, [10, 11]]

After extend,the list is: [1, 6, 2, 3, 4, 5, [10, 11], 15, 12, 13, 12]

The sixth element is: [10, 11] The seventh element is: 15 The eight element is: 12

The sixth of zero index element is: 10 after remove ,the list is: [15, 13, 12] after delete ,the list is: [15, 13]

before pop() ,the list is: [1, 6, 2, 3, 4, 5, [10, 11], 15, 12, 13, 12]

after pop() ,the list is: [1, 6, 2, 3, 4, 5, [10, 11], 15, 12, 13]

before pop(6) ,the list is: [1, 6, 2, 3, 4, 5, [10, 11], 15, 12, 13]

after pop(6) ,the list is: [1, 6, 2, 3, 4, 5, 15, 12, 13] before clear() ,the list is: [10, 11]

after clear() ,the list is: [] count(2) is: 3

index(2) is: 1

Sorting is: [1, 2, 3, 4, 5, 6, 12, 13, 15]

Sorting is: [15, 13, 12, 6, 5, 4, 3, 2, 1]

Reverse list is: [1, 2, 3, 4, 5, 6, 12, 13, 15]

[1, 2, 3] [1, 2, 3]

[4, 2, 3] [4, 2, 3]

[4, 2, 3] [4, 2, 3]

[5, 2, 3] [4, 2, 3]

[5, 2, 3] [5, 2, 3]

[6, 2, 3] [5, 2, 3]

Max of mylist is: 15Min of mylist is: 1 sum of mylist is: 61

all: False any: True all: True any: True all: False any: False

Return Type: <class 'enumerate'> The 0 element is: ece

The 1 element is: eee The 2 element is: cse

[65, 73, 84, 83] ['A', 'I', 'T', 'S']

True True True

#write a python program to convert temperature from fahrenheit to celsius and vice versa print('Enter (F) to convert temperature from fahrenheit to celsius')

print('Enter (C) to convert temperature from celsius to fahrenheit ') print('Enter (X) to Exit')

while True:

input\_temp=input('Enter your value:') if(input\_temp=='F' or input\_temp=='f'):

fahren\_temp=float(input('Enter fahrenheit temperature to convert:')) celsius\_temp=(fahren\_temp-32)\*5/9

print(fahren\_temp,'degress fahrenheit equals',format(celsius\_temp,'.2f'),'degree celsuis') elif(input\_temp=='C' or input\_temp=='c'):

celsuis\_temp=float(input('Enter celsuis temperature to convert:')) fahren\_temp=(9/5\*celsuis\_temp)+32

print(celsuis\_temp,'degree celsuis equals',format(fahren\_temp,'.2f'),'degree fahrenheit') else:

print('Thank You ')

exit()

''' OUTPUT:

Enter (F) to convert temperature from fahrenheit to celsius Enter (C) to convert temperature from celsius to fahrenheit Enter (X) to Exit

Enter your value:F

Enter fahrenheit temperature to convert:100

100.0 degress fahrenheit equals 37.78 degree celsuis Enter your value:C

Enter celsuis temperature to convert:25.28

25.28 degree celsuis equals 77.50 degree fahrenheit Enter your value:X

Thank You.....

'''#write a python program to read elements into a list and display them and also findmaximum element in a list

mylist=[]

size=int(input('Enter list size:')) for x in range(size):

ele=int(input('Enter your element:')) mylist.append(ele)

print('The list elements are:') for e in range(len(mylist)): print(mylist[e],end=' ') print()

max=min=mylist[0] sum=0

for e in range(len(mylist)): if(max<=mylist[e]):

max=mylist[e] if(min>=mylist[e]):

min=mylist[e] sum+=mylist[e]

print('The maximum element is:',max) print('The minimum element is:',min) print('The summation of all elements is:',sum) '''

OUTPUT

Enter list size:5

Enter your element:20 Enter your element:10 Enter your element:60 Enter your element:50 Enter your element:40 The list elements are:

20 10 60 50 40

The maximum element is: 60 The minimum element is: 10

The summation of all elements is: 180

'''#write a python program to demonstrate the working of tuple in python t=(1,2,3)

print(t) print(t[0])

print(t[-1]) del t

t=()

print(t) t1=(1)

print(type(t1)) t1=(1,)

print(type(t1)) t2=('ece','eee','cse') print(t2) t3=(1,2,3,4,6,5)

print(t3) t4=t2+t3 print(t4) print(t4[2:4])

print(t4[: :-1]) print(len(t4)) t5=(t2,t3) print(t5) print(t5[0]) print(t5[0][1]) print(any(t3)) print(all(t3))

print(sorted(t3,reverse=True)) print(sum(t3))

print(max(t3)) print(min(t2)) tuple1=tuple('aits') print(tuple1) print(tuple1[2])

for x in enumerate(t2): print(x)

''' OUTPUT:

(1, 2, 3)

1

3 ()

<class 'int'>

<class 'tuple'> ('ece', 'eee', 'cse') (1, 2, 3, 4, 6, 5)

('ece', 'eee', 'cse', 1, 2, 3, 4, 6, 5)

('cse', 1)

(5, 6, 4, 3, 2, 1, 'cse', 'eee', 'ece')

9

(('ece', 'eee', 'cse'), (1, 2, 3, 4, 6, 5))

('ece', 'eee', 'cse')eee True

True

[6, 5, 4, 3, 2, 1]

21

6

cse

('a', 'i', 't', 's')

t

(0, 'ece')

(1, 'eee')

(2, 'cse')

'''PROGRAMS ON DICTIONARY & SET TYPE:

#WRITE A PYTHON PROGRAM TO DEMONTARTE THE WORKING OF VARIOUS OPERATIONS ON DICTIONARY

mydict={ 'college':'AITS',

'branch':'ECE', 'count':65

}

print('My Dictionary is:',mydict) #length

print('Length of dictionary is:',len(mydict)) #accessing

print('First Element:',mydict['college']) print('Second Element:',mydict['branch']) print('Third Element:',mydict['count'])

print('First Element using get():',mydict.get('college')) #adding

mydict['count']=60 mydict['course']='B.Tech' print('After adding:',mydict)

mydict.update({'branch':'cse','count':63}) print(mydict) mydict.update({'name':'nag','num':1241}) print(mydict)

mylist=[['a',1],['b',2]] mydict.update(dict(mylist)) print(mydict)

#removing

del mydict['count'] print(mydict) val=mydict.pop('course') print(val,'was deleted.',mydict) val=mydict.popitem() print(val,'was deleted.',mydict) #mydict.clear()

print(mydict) #setdefault()

x=mydict.setdefault('college','aitsr') print(x)

print(mydict) #fromkeys() x=(1,2,3)

y=('ece','cse','eee') z=dict.fromkeys(x,y) print(z)

#keys()

for x in mydict.keys():

print(x) #values()

for x in mydict.values():

print(x)#items()

for x in mydict.items():

print(x)

#copy() new=mydict.copy() print(new)

''' OUTPUT:

**My Dictionary is: {'college': 'AITS', 'branch': 'ECE', 'count': 65} Length of dictionary is: 3**

**First Element: AITS Second Element: ECE**

**Third Element: 65**

**First Element using get(): AITS**

**After adding: {'college': 'AITS', 'branch': 'ECE', 'count': 60, 'course': 'B.Tech'}**

**{'college': 'AITS', 'branch': 'cse', 'count': 63, 'course': 'B.Tech'}**

**{'college': 'AITS', 'branch': 'cse', 'count': 63, 'course': 'B.Tech', 'name': 'nag', 'num': 1241}**

**{'college': 'AITS', 'branch': 'cse', 'count': 63, 'course': 'B.Tech', 'name': 'nag', 'num': 1241, 'a': 1, 'b': 2}**

**{'college': 'AITS', 'branch': 'cse', 'course': 'B.Tech', 'name': 'nag', 'num': 1241, 'a': 1, 'b': 2}**

**B.Tech was deleted. {'college': 'AITS', 'branch': 'cse', 'name': 'nag', 'num': 1241, 'a': 1, 'b': 2}**

**('b', 2) was deleted. {'college': 'AITS', 'branch': 'cse', 'name': 'nag', 'num': 1241, 'a': 1}**

**{'college': 'AITS', 'branch': 'cse', 'name': 'nag', 'num': 1241, 'a': 1} AITS**

**{'college': 'AITS', 'branch': 'cse', 'name': 'nag', 'num': 1241, 'a': 1}**

**{1: ('ece', 'cse', 'eee'), 2: ('ece', 'cse', 'eee'), 3: ('ece', 'cse', 'eee')} college**

**branch name num**

**a AITS**

**cse nag 1241**

**1**

**('college', 'AITS')**

**('branch', 'cse')**

**('name', 'nag')**

**('num', 1241)**

**('a', 1)**

**{'college': 'AITS', 'branch': 'cse', 'name': 'nag', 'num': 1241, 'a': 1}**

'''

'''#write a python program to demonstrate the working of mutable arguments in python functions

def findsum(num):

for x in range(len(num)):

if num[x]<0:

num[x]=0

return sum(num) mylist=[]

size=int(input('Enter size of list:')) for k in range(size):

mylist.append(int(input('Enter value:'))) print('My list is:',mylist) result=findsum(mylist)

print('My list is:',mylist) print('Result is:',result) '''

OUTPUT:

Enter size of list:5 Enter value:-1 Enter value:2 Enter value:3 Enter value:-7 Enter value:5

My list is: [-1, 2, 3, -7, 5]

My list is: [0, 2, 3, 0, 5]

Result is: 10 '''

'''#WRITE A PYTHON PROGRAM TO DEMONTARTE THE WORKING OF VARIOUS OPERATIONS ON SET

s={}

print(type(s)) s=set() print(type(s))

s1={1,2,3,4,5,6,4,3,2}

print('myset is:',s1) #len()

print('My set length is:',len(s1)) #copy()

s3=s1.copy() print('After copy:',s3) #add()

s1.add(8) print('After add:',s1) #update()

s1.update(['ece','cse']) print('After Update:',s1) #remove() s1.remove(2) print('After remove:',s1) #discard() s1.discard(8)

print('After discard:',s1) #pop()

print('pop element is:',s1.pop()) print('After pop:',s1)

#clear() s1.clear()

print('After clear:',s1) s1.update({1,2,3,'ece','cse',2.5}) print('My New Set:',s1) s2={1,2,4,5,6,'cse'}

print('set2 is:',s2) #set operations #union()

print('After union:',s1|s2) print('After union:',s1.union(s2)) #insection()

print('After intersection:',s1&s2)

print('After intersection:',s1.intersection(s2)) #difference()

print('After differnece:',s1-s2) print('After difference:',s1.difference(s2)) print('After differnece:',s2-s1) print('After difference:',s2.difference(s1)) #symmetric\_difference()

print('After symmetric\_differnece:',s1^s2)

print('After symmetric\_difference:',s1.symmetric\_difference(s2)) '''

OUTPUT:

<class 'dict'>

<class 'set'>

myset is: {1, 2, 3, 4, 5, 6} My set length is: 6

After copy: {1, 2, 3, 4, 5, 6}

After add: {1, 2, 3, 4, 5, 6, 8}

After Update: {1, 2, 3, 4, 5, 6, 'cse', 8, 'ece'}

After remove: {1, 3, 4, 5, 6, 'cse', 8, 'ece'}

After discard: {1, 3, 4, 5, 6, 'cse', 'ece'} pop element is: 1

After pop: {3, 4, 5, 6, 'cse', 'ece'} After clear: set()

My New Set: {1, 2, 3, 2.5, 'ece', 'cse'}

set2 is: {1, 2, 4, 5, 6, 'cse'}

After union: {1, 2, 3, 2.5, 4, 'cse', 5, 6, 'ece'}

After union: {1, 2, 3, 2.5, 4, 'cse', 5, 6, 'ece'}

After intersection: {1, 2, 'cse'}

After intersection: {1, 2, 'cse'}

After differnece: {'ece', 2.5, 3}

After difference: {'ece', 2.5, 3}

After differnece: {4, 5, 6}

After difference: {4, 5, 6}

After symmetric\_differnece: {'ece', 2.5, 3, 4, 5, 6}

After symmetric\_difference: {'ece', 2.5, 3, 4, 5, 6}

'''PROGRAMS ON FUNCTIONS:

#WRITE A PYTHON PROGRAM TO PERFORM ADDITION BETWEEN TWO NUMBERS USING FUNCTION

def add(a,b):

"""this function display addition of two numbers""" return(a+b)

a=int(input('Enter a value:')) b=int(input('Enter b value:')) result=add(a,b)

print('The addition of %d and %d is %d'%(a,b,result)) '''

OUTPUT:

Enter a value:10 Enter b value:20

The addition of 10 and 20 is 30

#write a python program to find average of three numbers using function def avg(n1,n2,n3):

return (n1+n2+n3)/3.0 print(avg(10,20,30)) print(avg(10,20,30)+10) if avg(10,23,-40)<0:

print('invalid') print(avg(avg(1,2,3),5,7))

print(avg(1,2,3)\*avg(1,2,3)) '''

20.0

30.0

invalid 4.666666666666667

4.0

#write a python program to find factorial of a number using recursive function

def recfact(n): if(n!=0):

return n\*recfact(n-1) else:

return 1 n=int(input('Enter n value:'))

print('The factorial is:',recfact(n)) '''

OUTPUT:

Enter n value:5

The factorial is: 120 '''

#write a python program to find sum of natural numbers using recursive function def recsum(n):

if(n!=0):

return n+recsum(n-1) else:

return 0 n=int(input('Enter n value:')) print('The sum is:',recsum(n)) '''

OUTPUT:

Enter n value:10 The sum is: 55 '''

#WRITE A PYTHON PROGRAM TO DISPLAY CALENDER MONTH USING MODULE DESIGN

#global scope days={1:31,2:28,3:31,4:30,5:31,6:30,7:31,8:31,9:30,10:31,11:30,12:31}

mnames={1:'JAN',2:'FEB',3:'MAR',4:'APR',5:'MAY',6:'JUN',7:'JUL',8:'AUG',9:'SEP',10:'OCT',11:'N OV',12:'DEC'}

mcodes={1:0,2:3,3:3,4:6,5:1,6:4,7:6,8:2,9:5,10:0,11:3,12:5}

weeks={0:'SUN',1:'MON',2:'TUE',3:'WED',4:'THR',5:'FRI',6:'SAT'}

#getting month def getmonth():

month=int(input('Enter Your month[1-12]:')) if month>=1 and month<=12:

return month else:

print('Invalid month number') return -1

#getting year def getyear():

year=int(input('Enter Your year[1800-2099]:')) if year>=1800 and year<=2099:

return year else:

print('Invalid year and enter year(1800-2099):') return -1

#processing calendar month def calmonth(year,month):

leap=leapyear(year) ndays=numdays(month,leap) dweek=daymonth(year,month,leap) return dweek

#checking for leap year def leapyear(year):

if ((year%4==0) and (not(year%100==0)) or (year%400==0)): return True

else:

return False

#finding number days in a month def numdays(month,leap):

if leap:

days[2]=29 else:

days[2]=28 return days[month]

#calculating starting day of the month def daymonth(year,month,leap):

centurydigit=year//100 yeardigit=year%100

if month in (1,2) and leap: yearcode=((yeardigit+(yeardigit//4))%7)-1

else:

yearcode=(yeardigit+(yeardigit//4))%7 monthcode=mcodes[month]

if centurydigit==18: centurycode=2

elif centurydigit==20: centurycode=6

else:

centurycode=0 dweek=(yearcode+monthcode+centurycode+1)%7 return dweek

#displaying calendar month def display(dweek,month,year):

print('\n',mnames[month],year) print(sep='',end=' ')

for wname in range(7): print(format(weeks[wname],'3'),end=' ')

print()

if dweek==6: start=6

else:

start=dweek current=1 width=4 blank=' '

blankcol=format(blank,str(width)) while current<=start:

print(blankcol,end='') current=current+1

currentday=1

while currentday<=days[month]: if currentday<10:

print(format(blank,'3')+str(currentday),end='') else:

print(format(blank,'2')+str(currentday),end='') if current<=6:

current=current+1 else:

current=1 print()

currentday=currentday+1 print('\n')

#main program terminate=False while not terminate:

month=getmonth() year=getyear()

if year==-1 or month==-1: terminate=True

else:

dweek=calmonth(year,month)

#print('Day of the Month is:',weeks[dweek]) display(dweek,month,year)

OUTPUT:

Enter Your month[1-12]:3

Enter Your year[1800-2099]:2020 MAR 2020

SUN MON TUE WED THR FRI SAT 1 2 3 4 5 6 7

8 9 10 11 12 13 14

|  |  |  |
| --- | --- | --- |
| **15** | **16** | **17 18 19 20 21** |
| **22** | **23** | **24 25 26 27 28** |
| **29** | **30** | **31** |

'''#WRITE A PYTHON PROGRAM TO DEMONTARTE THE WORKING OF CLASS AND OBJECT

class Parrot:

# class attribute name = ""

age = 0

# create parrot1 object parrot1 = Parrot() parrot1.name = "Blue" parrot1.age = 10

# create another object parrot2 parrot2 = Parrot() parrot2.name = "red" parrot2.age = 15

# access attributes

print(f"{parrot1.name} is {parrot1.age} years old") print(f"{parrot2.name} is {parrot2.age} years old")

OUTPUT:

Blue is 10 years old Red is 15 years old

'''#WRITE A PYTHON PROGRAM TO DEMONTARTE THE WORKING OF INHERITANCE CONCEPT

# base class

class Animal:

def eat(self):

print( "I can eat!") def sleep(self):

print("I can sleep!")

# derived class

class Dog(Animal):

def bark(self):

print("I can bark! Woof woof!!") **# Create object of the Dog class** dog1 = Dog()

# Calling members of the base class

dog1.eat() dog1.sleep()

# Calling member of the derived class

dog1.bark();

OUTPUT:

I can eat!

I can sleep!

I can bark! Woof woof!!

'''#WRITE A PYTHON PROGRAM TO DEMONTARTE THE WORKING OF ENCAPSULATION CONCEPT

class Computer: def \_\_init\_\_(self):

self.\_\_maxprice = 900 def sell(self):

print("Selling Price: {}".format(self.\_\_maxprice)) def setMaxPrice(self, price):

self.\_\_maxprice = price c = Computer()

c.sell()

# change the price **c.\_\_maxprice = 1000 c.sell()**

# using setter function **c.setMaxPrice(1000) c.sell()**

# OUTPUT

:

Selling Price: 900

Selling Price: 900

Selling Price: 1000

'''#WRITE A PYTHON PROGRAM TO DEMONTARTE THE WORKING OF MULTIPLE INHERITANCE

class Mammal:

def mammal\_info(self):

print("Mammals can give direct birth.")

class WingedAnimal:

def winged\_animal\_info(self): print("Winged animals can flap.")

class Bat(Mammal, WingedAnimal): pass

# # create an object of Bat class

b1 = Bat()

b1.mammal\_info() b1.winged\_animal\_info()

OUTPUT:

Mammals can give direct birth. Winged animals can flap.

'''#WRITE A PYTHON PROGRAM TO DEMONTARTE THE WORKING OF POLYMORPHISM

# # Parent class

class Shape:

def draw(self):

return "Drawing a shape"

# # Subclasses

class Circle(Shape): def draw(self):

return "Drawing a circle"

class Square(Shape): def draw(self):

return "Drawing a square"

# # Function using polymorphism

def show\_shape(shape): print(shape.draw())

# # Create objects of subclasses

circle = Circle() square = Square()

**# Use polymorphism** show\_shape(circle) show\_shape(square)

# OUTPUT:

Drawing a circle Drawing a square

'''#WRITE A PYTHON PROGRAM TO DEMONTARTE THE WORKING OF ALL INCORPORATING OOPS CONCEPTS

class Animal:

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

self.\_\_name = name def speak(self):

return "Some generic animal sound" def get\_name(self):

return self.\_\_name

# # Subclass 1

class Dog(Animal):

def speak(self):

**return "Woof!" # Subclass 2 class Cat(Animal):**

def speak(self):

return "Meow!"

# # Function using polymorphism

def make\_animal\_speak(animal): print(f"{animal.get\_name()} says {animal.speak()}")

# # Create objects of subclasses

dog = Dog("Buddy") cat = Cat("Whiskers")

**# Use polymorphism make\_animal\_speak(dog) make\_animal\_speak(cat)**

# OUTPUT:

Buddy says Woof! Whiskers says Meow!