In [1]:
```
#1)Display "Hello World" in your output screen.
print("hello world")
```
```
hello world
```

In [2]:
```
#2)Get the input from the user and perform addition of two numbers
a=int(input("enter a number:"))
b=int(input("enter a number:"))
sum=a+b
print("the sum is:",sum)
```
```
enter a number:3
enter a number:3
the sum is: 6
```

In [3]:
```
#3)swap two variables without temp variable
x=4
y=8
x,y=y,x
print("a=",x,"b=",y)
```
```
a= 8 b= 4
```

In [4]:
```
#4)convert the entered kilometres ( Convertion  Factor= 0.621371)
km=int(input("enter value in kilometers:"))
cf=0.621371
m=km*cf
print("km to miles",(km,m))
```
```
enter value in kilometers:4
km to miles (4, 2.485484)
```

In [2]:
```
#5)check whether the given number is positive, negative or 0
a=int(input("enter a number:"))
if a>0:
    print("positive number")
elif a==0:
    print("zero")
else:
    print("negative number")
```
```
enter a number:4
positive number
```

In [3]:
```
#6)verify that the given year is a leap year
a=int(input("enter the year"))
if(a%4==0)and(a%100!=0)or(a%400==0):
    print("leap year")
else:
    print("not leap year")
```
```
enter the year1947
not leap year
```

In [4]:
```
#7)display the prime numbers within the given interval
a=int(input("enter the lower range:"))
b=int(input("enter the higher range:"))
for n in range(a,b+1):
    if n>1:
        for i in range(2,n):
            if(n%i)==0:
                break
        else:
                print(n)
```
```
enter the lower range:5
enter the higher range:15
5
7
11
13
```

In [6]:
```
#8) display the Fibonacci sequence up to n-th term
a=int(input("enter the range:"))
n1,n2=0,1
c=0
if a<=0:
    print("enter the positive number")
elif a==1:
    print("fibonacci series upto",a,":")
    print(n1)
else:
    print("fibonacci sequence:")
while c<a:
    print(n1)
    n=n1+n2
    n1=n2
    n2=n
    c+=1
```
```
enter the range:5
fibonacci sequence:
0
1
1
2
3
```

In [10]:
```
#9) check if the number is an Armstrong number or not
a=int(input("enter the number:"))
sum=0
temp=a
while(temp>0):
    digit=temp%10
    sum+=digit**3
    temp//=10
if a==sum:
    print(a,"is an armstrong number")
else:
    print(a,"is not an armstrong number")
```
```
enter the number:4
4 is not an armstrong number
```

In [13]:
```
#10) Find the Sum of natural numbers up to n-th term
num = int(input("enter the number:"))
if num < 0:
   print("Enter a positive number")
else:
   sum = 0
   # use while loop to iterate until zero
   while(num > 0):
       sum += num
       num -= 1
   print("The sum is", sum)
```
```
enter the number:45
The sum is 1035
```

In [6]:
```
#11) Write a function called show_stars(rows). If rows are 5,
#it should print the following:
rows = int(input("Enter number of rows: "))

for i in range(rows):
    for j in range(i+1):
        print("* ", end="")
    print("\n")
```
```
Enter number of rows: 5
* 

* * 

* * * 

* * * * 

* * * * * 
```

In [1]:
```
#12) Write a program to remove characters from a string starting from zero up to
#n and return a new string.
input_string = "Adcictcya"
char_to_remove = "c"
newStr = ""
for character in input_string:
    if character != char_to_remove:
        newStr += character

print("The input string is:", input_string)
print("The character to delete is:", char_to_remove)
print("The output string is:", newStr)
```
```
The input string is: Adcictcya
The character to delete is: c
The output string is: Aditya
```

In [12]:
```
#13) Iterate the given list of numbers and print only those numbers which are
#divisible by 5
list_1 = [13, 14, 87, 44, 70, 9]
result = list (filter (lambda x: (x % 5 == 0), list_1))
print ("Numbers that are divisible by 5 are:",result)
```
```
Numbers that are divisible by 5 are: [70]
```

In [8]:
```
#14) Write a program to find how many times substring "Hi" appears in the given
#string.
s = 'hihihi'
sb = 'hi'
results = 0
sub_len = len(sb)
for i in range(len(s)):
    if s[i:i+sub_len] == sb:
        results += 1
print(results)
```
```
3
```

In [9]:
```
#15) Print the following pattern
rows = 6
for i in range(rows):
    for j in range(i):
        print(i, end=' ')
    print('')
```
```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

In [18]:
```
#16) Write a program to check if the given number is a palindrome number.
n=int(input("Enter number:"))
temp=n
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
if(temp==rev):
    print("The number is a palindrome!")
else:
    print("The number isn't a palindrome!")
```
```
Enter number:545
The number is a palindrome!
```

In [19]:
```
#List Exercise
#17) Python program to interchange first and last elements in a list
def swapList(newList):

    newList[0], newList[-1] = newList[-1], newList[0]

    return newList
newList = [12, 35, 9, 56, 24]
print(swapList(newList))
```
```
[24, 35, 9, 56, 12]
```

In [20]:
```
#18)Python program to swap two elements in a list
def swapPositions(list, pos1, pos2):

    list[pos1], list[pos2] = list[pos2], list[pos1]
    return list
List = [23, 65, 19, 90]
pos1, pos2  = 1, 3

print(swapPositions(List, pos1-1, pos2-1))
```
```
[19, 65, 23, 90]
```

In [21]:
```
#19)Python | Ways to find length of list

li = [10, 20, 30]
n = len(li)
print("The length of list is: ", n)
```
```
The length of list is:  3
```

In [28]:
```
#20)Maximum of two numbers in Python
list1 = [3, 2, 8, 5, 10, 12]
list1.sort()
print(list1)
print("maximum of two numbers:",list1[-1],list1[-2])
```
```
[2, 3, 5, 8, 10, 12]
maximum of two numbers: 12 10
```

In [30]:
```
#21)Minimum of two numbers in Python
list1 = [3, 2, 8, 5, 10, 12]
list1.sort()
print(list1)
print("minimum of two numbers:",list1[0],list1[1])
```
```
[2, 3, 5, 8, 10, 12]
miniimum of two numbers: 2 3
```

In [10]:
```
#String Exercises
#22)Python program to check whether the string is Symmetrical or Palindrome
def palindrome(a):
    mid=(len(a)-1)//2
    start=0
    last=len(a)-1
    flag=0
    while(start<mid):
        if(a[start]==a[last]):
            start+=1
            last-=1
        else:
            flag=1
            break;
    if flag==0:
        print("the entered string is palindrome")
    else:
        print("The entered string is not palindrome")

def symmetry(a):
    n=len(a)
    flag=0
    if n%2:
        mid=n//2+1
    else:
        mid=n//2
    start1=0
    start2=mid
    while(start1<mid and start2<n):
        if(a[start1]==a[start2]):
            start1=start1+1
            start2=start2+1
        else:
            flag=1
            break
    if flag==0:
        print("The entered string is symmetrical")
    else:
        print("The entered string is not symmetrical")



string='amaama'
palindrome(string)
symmetry(string)
```
```
the entered string is palindrome
The entered string is symmetrical
```

In [36]:
```
#23)Reverse words in a given String in Python
string = "Welcome to the karkalan magic show"
s = string.split()[::-1]
l = []
for i in s:
    l.append(i)
print(" ".join(l))
```
```
show magic karkalan the to Welcome
```

In [39]:
```
#24)Ways to remove i'th character from string in Python
string1 = "subashinicarounagarane"
string2 = ""

for i in range(len(string1)):
    if i != 2:
        string2 = string2+ string1[i]
print ("The string after removal of i'th character : " +string2)
```
```
The string after removal of i'th character : suashinicarounagarane
```

In [46]:
```
#25)Find length of a string in python
string="subashini"
a=len(string)
print(a)
```
```
9
```

In [49]:
```
#26)Python program to print even length words in a string
n="Never give up"
s=n.split(" ")
for i in s:
    if len(i)%2==0:
        print(i)
```
```
give
up
```

In [52]:
```
#Tuple Exercises
#27)Python program to Find the size of a Tuple
size = ("A", 1, "B", 2, "C", 3)

print("Size of Tuple1: " + str(size.__sizeof__()) + "bytes")
```
```
Size of Tuple1: 72bytes
```

In [25]:
```
#28)Python – Maximum and Minimum K elements in Tuple
t=(1,2,3,4,5)
print("maximum value=",max(t))
print("minimum value=",min(t))
```
```
maximum value= 5
minimum value= 1
```

In [22]:
```
#29)Python – Sum of tuple elements
def summation(test_tup):
    test = list(test_tup)
    count = 0
    for i in test:
        count += i
    return count
test_tup = (5, 20, 3, 7, 6, 8)
print(summation(test_tup))
```
```
49
```

In [27]:
```
#30)Python – Row-wise element Addition in Tuple Matrix
tmat=((1,2,3),(4,5,6),(7,8,9))
for row in tmat:
    s=sum(row)
    print("row sum:",s)
```
```
row sum: 6
row sum: 15
row sum: 24
```

In [2]:
```
#31)Create a list of tuples from given list having number and
#its cube in each tuple
def cubeoflist(li):
    result=[(num,num**3) for num in li]
    return result
li=[6,4,5,9]
print(cubeoflist(li))
```
```
[(6, 216), (4, 64), (5, 125), (9, 729)]
```

In [8]:
```
#Dictionary Exercises
#32)Python | Sort Python Dictionaries by Key or Value
myDict={'ravi':11,'ranjish':10,'sanjeev':1,'yash':2,'suraj':32}
myKeys=list(myDict.keys())
myKeys.sort()
sorted_dict={i:myDict[i] for i in myKeys}
print(sorted_dict)
```
```
{'ran': 10, 'rav': 11, 'sanjeev': 1, 'suraj': 32, 'yash': 2}
```

In [9]:
```
#33)Python dictionary with keys having multiple inputs
dict={}
a,b,c,d=5,6,7,8
p,q,r,s=1,2,3,4
dict["w-x+y*z"]=[a-b+c*d,p-q+r*s]
print(dict)
```
```
{'w-x+y*z': [55, 11]}
```

In [12]:
```
#34)Python program to find the sum of all items in a dictionary
dict={'a':121,'b':232,'c':343,'d':454}
print("Dictionary:",dict)
print("sum:",sum(dict.values()))
```
```
Dictionary: {'a': 121, 'b': 232, 'c': 343, 'd': 454}
sum: 1150
```

In [15]:
```
#35)Python program to find the size of a Dictionary
import sys
dict1={"a":1,"b":2,"c":3}
dict2={"Geek1":"Raju","Geek2":"Nikhil","Geek3":"Deepanshu"}
dict3={1:"Lion",2:"Tiger",3:"Fox",4:"wolf"}
print("Size of dict1:"+str(sys.getsizeof(dict1))+ " bytes")
print("Size of dict2:"+str(sys.getsizeof(dict2))+ " bytes")
print("Size of dict3:"+str(sys.getsizeof(dict3))+ " bytes")
```
```
Size of dict1:232 bytes
Size of dict2:232 bytes
Size of dict3:232 bytes
```

In [18]:
```
#Set Exercises
#36)Find the size of a Set in Python
import sys
set1={"a",1,"b",2,"c",3}
set2={"Geek1","Raju","Geek2","Nikhil","Geek3","Deepanshu"}
set3={(1,"Lion"),(2,"Tiger"),(3,"Fox"),(4,"Wolf")}
print("Size of set1:"+str(sys.getsizeof(set1))+ " bytes")
print("Size of set2:"+str(sys.getsizeof(set2))+ " bytes")
print("Size of set3:"+str(sys.getsizeof(set3))+ " bytes")
```
```
Size of set1:472 bytes
Size of set2:472 bytes
Size of set3:216 bytes
```

In [26]:
```
#37)Iterate over a set in Python
test_set=set("geEks")
for val in test_set:
    print(val)
```
```
s
g
E
k
e
```

In [25]:
```
#38)Python – Maximum and Minimum in a Set
def MAX(sets):
    return (max(sets))
sets=set([23,8,19,28,25,78,98,345,678,999])
print("The maximum element in the set is:",MAX(sets))

def MIN(sets):
    return (min(sets))
print("The minimum element in the set is:",MIN(sets))
```
```
The maximum element in the set is: 999
The minimum element in the set is: 8
```

In [27]:
```
#39)Python – Remove items from Set
languages={'Python','Java','English','C','C++','Tamil','Hindi'}
languages.remove('C')
print(languages)
```
```
{'Hindi', 'C++', 'English', 'Java', 'Python', 'Tamil'}
```

In [32]:
```
#40)Python – Check if two lists have atleast one element common
def common_data(list1,list2):
    result=False
    for x in list1:
        for y in list2:
            if x==y:
                result=True
                return result
    return result
a=[1,2,3,4,5]
b=[5,6,7,8,9]
print(common_data(a,b))

a=[1,2,3,4,5]
b=[6,7,8,9]
print(common_data(a,b))
```
```
True
False
```

In [40]:
```
#Matrix Exercises
#41)Python – Assigning Subsequent Rows to Matrix first row elements
test_list = [[5, 8, 9], [2, 0, 9], [5, 4, 2], [2, 3, 9]]


print("The original list: " + str(test_list))


res = {test_list[0][ele] : test_list[ele+1] for ele in range(len(test_list)-1)}


print("The Assigned Matrix : "+str(res))
```
```
The original list: [[5, 8, 9], [2, 0, 9], [5, 4, 2], [2, 3, 9]]
The Assigned Matrix : {5: [2, 0, 9], 8: [5, 4, 2], 9: [2, 3, 9]}
```

In [20]:
```
#42)Adding and Subtracting Matrices in Python
import numpy as np
A = np.array([[1, 2], [3, 4]])
B = np.array([[4, 5], [6, 7]])
print("Printing elements of first matrix")
print(A)
print("Printing elements of second matrix")
print(B)
print("Addition of two matrix")
print(np.add(A, B))
print("subtaction of two matrix")
print(np.subtract(A, B))
```
```
Printing elements of first matrix
[[1 2]
 [3 4]]
Printing elements of second matrix
[[4 5]
 [6 7]]
Addition of two matrix
[[ 5  7]
 [ 9 11]]
subtaction of two matrix
[[-3 -3]
 [-3 -3]]
```

In [21]:
```
#43)Python – Group similar elements into Matrix
my_list = [[14, 62], [51, 23], [12, 62], [78, 87], [41, 14]]
print("The list is :")
print(my_list)
check_list = [14, 12, 41, 62]
print("The list is :")
print(check_list)
my_result = []
while my_list:
    sub_list_1 = my_list.pop()
    sub_list_2 = [element for element in check_list if element not in sub_list_1]
    try:
        my_list.remove(sub_list_2)
        my_result.append([sub_list_1, sub_list_2])
    except ValueError:
        my_result.append(sub_list_1)
print("The result is :")
print(my_result)
```
```
The list is :
[[14, 62], [51, 23], [12, 62], [78, 87], [41, 14]]
The list is :
[14, 12, 41, 62]
The result is :
[[[41, 14], [12, 62]], [78, 87], [51, 23], [14, 62]]
```

In [22]:
```
#44)Python – Row-wise element Addition in Tuple Matrix
test_list = [[('Gfg', 3), ('is', 3)], [('best', 1)], [('for', 5), ('geeks', 1)]]
print("The original list is : " + str(test_list))
cus_eles = [6, 7, 8]
res = [[sub + (cus_eles[idx], ) for sub in val] for idx, val in enumerate(test_list)]
print("The matrix after row elements addition : " + str(res))
```
```
The original list is : [[('Gfg', 3), ('is', 3)], [('best', 1)], [('for', 5), ('geeks', 1)]]
The matrix after row elements addition : [[('Gfg', 3, 6), ('is', 3, 6)], [('best', 1, 7)], [('for', 5, 8), ('geeks', 1, 8)]]
```

In [4]:
```
#45)Create an n x n square matrix, where all the sub-matrix have the
#sum of opposite corner elements as even
import itertools
def sub_mat_even(n):
    temp = itertools.count(1)
    l = [[next(temp)for i in range(n)]for i in range(n)]
    if n%2 == 0:
        for i in range(0,len(l)):
            if i%2 == 1:
                l[i][:] = l[i][::-1]
    for i in range(n):
        for j in range(n):
            print(l[i][j],end=" ")
        print()
n = 4
sub_mat_even(n)
```
```
1 2 3 4 
8 7 6 5 
9 10 11 12 
16 15 14 13 
```

In [5]:
```
#Functions Exercises
#46)How to get list of parameters name from a function in Python?
def fun(a, b):
    return a**b
import inspect
print(inspect.signature(fun))
```
```
(a, b)
```

In [6]:
```
#47)How to Print Multiple Arguments in Python?
def GFG(name, num):
    print("Hello from ", name + ', ' + num)
GFG("geeks for geeks", "25")
```
```
Hello from  geeks for geeks, 25
```

In [7]:
```
#48)Python program to find the power of a number using recursion
def power(N, P):
    if P == 0:
        return 1
    return (N*power(N, P-1))
if __name__ == '__main__':
    N = 5
    P = 2
    print(power(N, P))
```
```
25
```

In [8]:
```
#49)Sorting objects of user defined class in Python
print(sorted([1,26,3,9]))
print(sorted("Geeks foR gEEks".split(), key=str.lower))
```
```
[1, 3, 9, 26]
['foR', 'Geeks', 'gEEks']
```

In [9]:
```
#50)Functions that accept variable length key value pair as arguments
def printKwargs(**kwargs):
    print(kwargs)
if __name__ == "__main__":
    printKwargs(Argument_1='gfg', Argument_2='GFG')
```
```
{'Argument_1': 'gfg', 'Argument_2': 'GFG'}
```

In [ ]: