

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

hello world

```
In [56]: #Get the input from the user and perform addition of two numbers
a=int(input("enter the first number:"))
b=int(input("enter the second number:"))
c=a+b
print("the sum is:",c)
```

enter the first number:7
enter the second number:4
the sum is: 11

```
In [4]: #convert the entered kilometres
a=int(input("enter the number of kilometres:"))
a=a*0.621371
print("the number of metres is",a)
```

enter the number of kilometres:3
the number of metres is 1.8641130000000001

```
In [6]: #check whether the given number is positive, negative or 0
a=int(input("enter the number :"))
if a > 0:
    print("positive number")
elif a < 0:
    print("negative number")
else:
    print("zero")
```

enter the number :4
positive number

```
In [7]: #verify that the given year is a leap year
year=int(input("enter the year:"))
if year/4 :
    print("it is a leap year")
else:
    print("it is not a leap year")
```

enter the year:2018
it is a leap year

```
In [11]: #display the prime numbers within the given interval
a=int(input("enter the min number:"))
b=int(input("enter the max number:"))
print("prime number between",a,"to",b,"are")
for num in range(a,b+1):
    for i in range(2,num):
        if(num%i==0) :
            break
        else:
            print(num)
```

enter the min number:2
enter the max number:7
prime number between 2 to 7 are
3
5
5
5
7
7
7
7
7

```
In [14]: #Fibonacci nmubers
a=0
b=1
n=int(input("Enter the range: "))
print("The fibonacci numbers are: ")
for x in range(1,n-1,1):
    sum=a+b
    print(sum)
    a=b
    b=sum
```

Enter the range: 10
The fibonacci numbers are:
1
2
3
5
8
13
21
34

```
In [2]: #check if the number is an Armstrong number or not
y=int(input("Enter your number:"))
sum=0
temp=y
d=temp%10
e=(temp//10)%10
f=int(temp/100)
sum=(d**3)+(e**3)+(f**3)
if sum==y:
    print("It is an armstrong number")
else:
    print("It is not an armstrong number")
```

Enter your number:4
It is not an armstrong number

```
In [58]: #Find the Sum of natural numbers up to n-th term
y=int(input("enter the sum for n th term: "))
sum=0
for x in range(1,y+1,1):
    sum+=x
print("sum of n terms",sum)
```

enter the sum for n th term: 5
sum of n terms 15

```
In [57]: #Write a function called show_stars(rows). If rows are 5, it should print the followi
def show_stars(rows):
    for i in range(1,rows+1):
        print("*"*i)
show_stars(int(input("Enter your number: ")))
```

Enter your number: 5
*
**


```
In [15]: #Write a program to remove characters from a string starting from zero up to n and
def remove_chars(str, n):
    return str[n:]
my_string = input("Enter your string:")
i=int(input("Enter the index number where u want to remove: "))
new_string = remove_chars(my_string, i)
print(new_string)
```

Enter your string:ilaks
Enter the index number where u want to remove: 3
ks

```
In [20]: #Iterate the given list of numbers and print only those numbers which are divisible
n=int(input("enter the range : "))
list=[]
for i in range (0,n):
    c=int(input("enter the elements : "))
    list.append(c)
print("the numbers divisibl by 5 are : ")
for i in list:
    if i%5==0:
        print(i)
```

enter the range : 5
enter the elements : 34
enter the elements : 78
enter the elements : 65
enter the elements : 45
enter the elements : 42
the numbers divisibl by 5 are :
65
45

```
In [21]: #Write a program to find how many times substring "Hi" appears in the given string.
str=("Hi,This is my python assignment ,Hi")
substr="Hi"
```

```
count=str.count(substr)
print("The count of the substring is : ",count)
```

The count of the substring is : 2

```
In [25]: #Print the number pattern
n = 6
for number in range(n):
    for i in range(number):
        print(number, end=" ")
    print(" ")
```

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

```
In [27]: #Write a program to check if the given number is a palindrome number
num = input("Enter a number:")
if num == num[::-1]:
    print("Yes its a palindrome")
else:
    print("No, its not a palindrome")
```

Enter a number:282
Yes its a palindrome

```
In [28]: #Python program to interchange first and last elements in a list
my_list = [15,86,95,76,73,64]
print("Initial list: ")
print(my_list)
my_list[0], my_list[-1] = my_list[-1], my_list[0]
print("Updated list after swapping:")
print(my_list)
```

Initial list:
[15, 86, 95, 76, 73, 64]
Updated list after swapping:
[64, 86, 95, 76, 73, 15]

```
In [29]: #Swapping of two numbers in a list
my_list = [58,75,69,37,25,589]
print("The initial list is:")
print(my_list)
i1 =int(input("Enter i1:"))
i2 =int(input("Enter i2:"))
temp = my_list[i1]
my_list[i1] = my_list[i2]
my_list[i2] = temp
print("The Updated list is:")
print(my_list)
```

The initial list is:
[58, 75, 69, 37, 25, 589]
Enter i1:4
Enter i2:2
The Updated list is:
[58, 75, 25, 37, 69, 589]

```
In [30]: #Python Ways to find length of list
my_list = [100,200,300,400,500]
print("My list elements: ")
print(my_list)
length = len(my_list)
print("The total length of my list is: ")
print(length)
```

My list elements:
[100, 200, 300, 400, 500]
The total length of my list is:
5

```
In [31]: #Maximum of two numbers in Python
a=int(input("Enter the value of a:"))
b=int(input("Enter the value of b:"))
if(a>b):
    print ("a is greater")
else:
    print("b is greater")
```

Enter the value of a:4
Enter the value of b:8
b is greater

```
In [32]: #Minimum of two numbers in Python
a=int(input("Enter the value of a:"))
b=int(input("Enter the value of b:"))
if(a<b):
    print ("a is smaller")
```

```
else:
    print("b is smaller")
```

Enter the value of a:4
Enter the value of b:6
a is smaller

In [34]: *#Python program to check whether the string is Symmetrical or Palindrome*

```
my_string = input("Enter the string:")
symmetrical = my_string == my_string[::-1]
palindrome = my_string == "".join(reversed(my_string))
if symmetrical:
    print("The string is symmetrical")
else:
    print("The string is not symmetrical")
if palindrome:
    print("The string is a palindrome")
else:
    print("The string is not a palindrome")
```

Enter the string:malayalam
The string is symmetrical
The string is a palindrome

In [35]: *#Reverse words in a given String in Python*

```
my_string = "hey ilaks"
print("My initial string is:")
print(my_string)
words = my_string.split()
words.reverse()
new_string = " ".join(words)
print("My reversed string is:")
print(new_string)
```

My initial string is:
hey ilaks
My reversed string is:
ilaks hey

In [36]: *#Ways to remove i'th character from string in Python*

```
my_string = "Hello!"
index_to_remove = int(input("Enter the index number to be removed:"))
new_string = my_string[:index_to_remove] + my_string[index_to_remove+1:]
print(new_string)
```

Enter the index number to be removed:4
Hell!

In [37]: *#Find length of a string in python*

```
my_string = "hey ilaks"
string_length = len(my_string)
print("Length of my string is:")
print(string_length)
```

Length of my string is:
9

In [45]: *#Python program to print even length words in a string*

```
print('enter a string')
n=input()
s=n.split(" ")
print("The even indexed strings are:")
for i in s:
    #checking the length of words
    if len(i)%2==0:
        print(i)
```

enter a string
python program
The even indexed strings are:
python

In [48]: *#Python program to Find the size of a Tuple*

```
import sys
# Define a tuple
my_tuple = ('ilaks',2003)
# Get the size of the tuple in bytes
size = sys.getsizeof(my_tuple)
# Print the size in bytes
print(f"The size of the tuple is {size} bytes")
```

The size of the tuple is 56 bytes

In [51]: *#Python – Maximum and Minimum K elements in Tuple*

```
import heapq
def find_k_largest_smallest_elements(k, my_tuple):
    # Find the k largest elements using the nlargest function
    largest_elements = heapq.nlargest(k, my_tuple)

    # Find the k smallest elements using the nsmallest function
    smallest_elements = heapq.nsmallest(k, my_tuple)
```

```

    return largest_elements, smallest_elements
my_tuple = (10,20,30,40,50,60,70,80,90,100)
k=int(input("Enter no. of elements needed:"))
largest, smallest = find_k_largest_smallest_elements(k, my_tuple)
print(f"The {k} largest elements in the tuple are: {largest}")
print(f"The {k} smallest elements in the tuple are: {smallest}")

```

Enter no. of elements needed:3
The 3 largest elements in the tuple are: [100, 90, 80]
The 3 smallest elements in the tuple are: [10, 20, 30]

```

In [52]: #Python - Sum of tuple elements
my_tuple=(20,40,50,60,80)
print("Tuple=",my_tuple)
sum_of_tuple = sum(my_tuple)
print("The sum of my tuple elements is:", sum_of_tuple)

```

Tuple= (20, 40, 50, 60, 80)
The sum of my tuple elements is: 250

```

In [54]: #Python - Row-wise element Addition in Tuple Matrix
matrix = ((1,2,3),(4,5,6),(7,8,9))
print("My row matrix:",matrix)
print("The sum of each row matrix is:")
for row in matrix:
    row_sum = sum(row)
    print(row_sum)

```

My row matrix: ((1, 2, 3), (4, 5, 6), (7, 8, 9))
The sum of each row matrix is:
24

```

In [59]: #swap two variables without temp variable
a=int(input("enter the first number:"))
b=int(input("enter the second number:"))
print("before swapping",a,b)
a=a+b
b=a-b
a=a-b
print("after swapping",a,b)

```

enter the first number:5
enter the second number:4
before swapping 5 4
after swapping 4 5

```

In [1]: #Create a list of tuples from given list having number and its cube in each tuple
list=[1,2,3,4]
res=[(val,pow(val,3)) for val in list]
print(res)

[(1, 1), (2, 8), (3, 27), (4, 64)]

```

```

In [5]: #Python Sort Python Dictionaries by Key or value
my_dict = {('John', 25): 'USA', ('Alice', 30): 'Canada', ('Bob', 35): 'Australia'}
print(my_dict[('John',25)])
print(my_dict[('Alice',30)])
print(my_dict[('Bob',35)])

```

USA
Canada
Australia

```

In [11]: #Python program to find the sum of all items in a dictionary
my_dict = {'a':10, 'b':20, 'c':30, 'd':40}
sum_of_values = sum(my_dict.values())
print("Sum of values:",sum_of_values)

```

Sum of values: 100

```

In [14]: #Python program to find the size of a Dictionary
my_dict = {'a':10,'b':20,'c':30,'d':40}
dict_size = len(my_dict)

print("Size of the dictionary:",dict_size)

```

Size of the dictionary: 4

```

In [15]: #Find the size of a Set in Python
my_set = {1,2,3,4,5}

set_size = len(my_set)

print("Size of the set:",set_size)

```

Size of the set: 5

```

In [18]: #Iterate over a set in Python

```

```
my_set = {1,2,3,4,5}
```

```
for item in my_set:  
    print(item)
```

```
1  
2  
3  
4  
5
```

In [19]: *#Python – Maximum and Minimum in a Set*

```
my_set = {5,2,8,1,9}
```

```
maximum_value = max(my_set)
```

```
minimum_value = min(my_set)
```

```
print("Maximum value:",maximum_value)  
print("Minimum value:",minimum_value)
```

```
Maximum value: 9  
Minimum value: 1
```

In [20]: *#Python – Remove items from Set*

```
my_set = {1,2,3,4,5}
```

```
my_set.remove(3)  
print(my_set)
```

```
my_set.discard(5)  
print(my_set)
```

```
my_set.discard(10)  
print(my_set)
```

```
{1, 2, 4, 5}  
{1, 2, 4}  
{1, 2, 4}
```

In [21]: *#Python – Check if two lists have atleast one element common*

```
list1 = [1,2,3,4,5]  
list2 = [5,6,7,8,9]
```

```
set1 = set(list1)  
set2 = set(list2)
```

```
common_elements = set1.intersection(set2)
```

```
if common_elements:  
    print("The lists have at least one common element.")  
else:  
    print("The lists do not have any common elements.")
```

```
The lists have at least one common element.
```

In [1]: *#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 [7]: *#Adding and Subtracting Matrices in Python*

```
# importing numpy as np  
import numpy as np
```

```
#creating first matrix  
A = np.array([[1,2],[3,4]])
```

```
# creating second matrix  
B = np.array([[4,5],[6,7]])
```

```
print("Printing elements of first matrix")  
print(A)  
print("Printing elements of second matrix")  
print(B)
```

```
# adding two matrix  
print("Addition of two matrix")  
print(np.add(A,B))  
#subtracting two matrix
```

```
print("subtraction 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]]
subtraction of two matrix
[[-3 -3]
 [-3 -3]]
```

```
In [8]: #Python – Group similar elements into Matrix
from itertools import groupby
test_list=[1,3,5,1,3,2,5,4,2]
print("The original list:" +str(test_list))
res =[list(val) for key, val in groupby(sorted(test_list))]
print("Matrix after grouping:"+str(res))
```

```
The original list:[1, 3, 5, 1, 3, 2, 5, 4, 2]
Matrix after grouping:[[1, 1], [2, 2], [3, 3], [4], [5, 5]]
```

```
In [9]: #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 [10]: #Create an n x n square matrix, where all the sub-matrix has 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 [11]: #How to get list of parameters name from a function in Python
import inspect
import collections
```

```
print(inspect.signature(collections.Counter))

(iterable=None, /, **kwargs)
```

```
In [12]: #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 [13]: #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 [14]: *#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 [15]: *#Functions that accept variable length key value pair as arguments*
 def printValues(**kwargs):
 for key, value in kwargs.items():
 print("The value of {} is {}".format(key, value))

 # driver code
 if __name__ == '__main__':
 printValues(abbreviation="GFG", full_name="geeksforgeeks")

```

The value of abbreviation is GFG
The value of full_name is geeksforgeeks

```

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js