Q.1) SWAP TWO NUMBERS

In [1]:
```
x=int(input("enter the first number "))
y=int(input("enter the Second number "))
print(x,y)
x=x+y
y=x-y
x=x-y
print(x,y)
```

```
enter the first number 4
enter the Second number 5
4 5
5 4
```

Q.2) Write a Program to extract each digit from an integer in the reverse order.

For example, If the given int is 7536, the output shall be “6 3 5 7“, with a space separating the digits.

In [3]:
```
n = int(input("enter the numbers : "))
L = []

while n > 0:
    a = n % 10
    L.append(a)
    n = n // 10

output = ' '.join(map(str, L))
print(output)
```

```
enter the numbers : 7536
6 3 5 7
```

reverse the number

In [11]:
```
n = input("enter the number :")
output = n[::-1]
print(output)
```

```
enter the number :7536
6357
```

reverse the integer number

In [12]:
```
n = int(input("enter the number :"))

rev = 0

while n > 0:
    a = n % 10
    rev = rev*10 + a
    n = n // 10

print(rev)
```

```
enter the number :7536
6357
```

Q.3) Write a program that will tell the number of dogs and chicken are there when the user will provide the value of total heads and legs.

For example: Input: heads -> 4 legs -> 12

Output: dogs -> 2 chicken -> 2

In [4]:
```
heads = int(input("enter number of heads :"))
legs = int(input("enter number of legs :"))

if heads < legs:
    if legs % 2 != 0:
        print("The number of legs must be even.")
    else:
        d = (legs - 2 * heads) / 2
        c = heads - d

else :
    print('pgla gya kya')

print("number of dogs : ", d)
print("number of chickens : ", c)
```

```
enter number of heads :4
enter number of legs :12
number of dogs :  2.0
number of chickens :  2.0
```

In [2]:
```
# number of head   -->    1    1
 # number of legs  -->    4   2

head=int(input("enter the number of heads: "))
legs=int(input("enter the number of legs: "))

if head/legs==0.25:
    number_of_dogs=legs/4
    print("there are ",round(number_of_dogs),'number of dogs')
elif head/legs==0.5:
    number_of_chicken=legs/2
    print("there are ",round(number_of_chicken),'number of chicken')
else:
    print("enter correct numbers")
```

```
enter the number of heads: 2
enter the number of legs: 4
there are  2 number of chicken
```

Q.4) Given the height, width and breadth of a milk tank, you have to find out how many glasses of milk can be obtained? Assume all the inputs are provided by the user.

Input:

Dimensions of the milk tank

H = 20cm, L = 20cm, B = 20cm

Dimensions of the glass

h = 3cm, r = 1cm

In [18]:
```
import math

H = int(input("enter the height of tank :"))
L = int(input("enter the length of tank :"))
B = int(input("enter the breadth of tank :"))

h = int(input("enter the height of glass :"))
r = int(input("enter the radius of glass :"))

volume_tank = H * L * B

volume_glass = math.pi * (r ** 2) * h

num_glasses = volume_tank // volume_glass

print(f"Number of glasses of milk: {int(num_glasses)}")
```

```
enter the height of tank :14
enter the length of tank :12
enter the breadth of tank :15
enter the height of glass :2
enter the radius of glass :4
Number of glasses of milk: 25
```

Q.5) Display Fibonacci series up to 10 terms.

In [21]:
```
n = int(input("enter the number :"))
start = 0
next = 1
print(start)
print(next)

for i in range(1,n+1):
    next_term = start+next
    print(next_term)
    start = next
    next = next_term
```

```
enter the number :10
0
1
1
2
3
5
8
13
21
34
55
89
```

Q.6) Find the factorial of a given number.

In [22]:
```
n = int(input("enter the number : "))
fact = 1

for i in range(1,n+1):
    fact = fact*i

print(fact)
```

```
enter the number : 5
120
```

Q.7) Write a program to print whether a given number is a prime number or not

In [12]:
```
n = int(input("Enter the number: "))

flag = True

for i in range(2,n):
    if n % i == 0:
        flag = False
        break

if flag == True:
    print(n, "is a prime number")

else:
    print(n, "is not a prime number")
```

```
Enter the number: 17
17 is a prime number
```

print the prime number between the range

In [38]:
```
lower = int(input("enter the lower range : "))
upper = int(input("enter the upper range : "))

print(" ")
print("Prime numbers between", lower, "and", upper, "are:")

for num in range(lower, upper + 1):
    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                break
        else:
            print(num)
```

```
enter the lower range : 1
enter the upper range : 100

Prime numbers between 1 and 100 are:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

Q.8) Print all the Armstrong numbers in a given range.

In [12]:
```
n = int(input("enter the number : "))

number = n
degree = len(str(n))
result = 0

while n > 0:
    a = n % 10
    result = result + a**degree
    n = n//10

if (number == result):
    print("armstrong number")
else:
    print("not armstrong number")
```

```
enter the number : 9474
armstrong number
```

Armstrong number between a range 100 to 1000

In [20]:
```
armstrong_list = []

for i in range(100,1000):
    a = i%10
    num = i//10
    b = num%10
    c = num//10

    if (a**3)+(b**3)+(c**3)==i:
          armstrong_list.append(i)

print(armstrong_list)
```

```
[153, 370, 371, 407]
```

Q.9) Write a program that will give you the sum of 3 digits

In [15]:
```
n = int(input("enter the 3 digits number : "))

a = n%10
n = n//10
b = n%10
n = n//10
c = n%10

print("sum of your number is : ",a+b+c)
```

```
enter the 3 digits number : 142
sum of your number is :  7
```

Q.10) Write a program that will reverse a four digit number.Also it checks whether the reverse is true

In [36]:
```
n = int(input("Enter the number: "))
rev = 0

while n > 0:
    a = n % 10
    rev = rev * 10 + a
    n = n // 10

print(rev)

if n == rev:
    print("Your reverse number is true")
else:
    print("Kuch gadbad h bro")
```

```
Enter the number: 147
741
Kuch gadbad h bro
```

Q.11) Write a program to find the euclidean distance between two coordinates.

In [37]:
```
import math
x1=float(input("x1: "))
y1=float(input("y1: "))
x2=float(input("x2: "))
y2=float(input("y2: "))
x=[x1,y1]
y=[x2,y2]
print("="*100)
print("Eucledian distance for given co-ordinate will be",round(math.dist(x,y),
```

```
x1: 1
y1: 4
x2: 7
y2: 5
====================================================================================
======================
Eucledian distance for given co-ordinate will be 6.08
```

In [38]:
```
x1=float(input("x1: "))
y1=float(input("y1: "))
x2=float(input("x2: "))
y2=float(input("y2: "))

d = ((x2-x1)**2 + (y2-y1)**2)**(0.5)

print("="*100)
print("Eucledian distance for given co-ordinate will be",round(d,2))
```

```
x1: 1
y1: 4
x2: 7
y2: 5
====================================================================================
======================
Eucledian distance for given co-ordinate will be 6.08
```

Q.12) Write a program that will tell whether the given number is divisible by 3 & 6

In [40]:
```
num=int(input("enter the number "))

if num%3==0 and num%6==0:
    print("the number is divisible by 3 and 6")
else:
    print("the number is not divisible by 3 and 6")
```

```
enter the number 18
the number is divisible by 3 and 6
```

Q.13) Write a program that will take three digits from the user and add the square of each

digit.

In [42]:
```
n = int(input("enter the 3 digit number : "))
result = 0

for i in range(1,n+1):
    a = n%10
    result = result + a**2
    n = n//10

print("sum of square of each digit is : ",result)
```

```
enter the 3 digit number : 123
sum of square of each digit is :  14
```

Q.14) Display three string “Name”, “Is”, “James” as “Name ** Is ** James

In [43]:
```
print("name","is","james",sep="**")
```

```
name**is**james
```

Q.15) Convert Decimal number to octal using print() output formatting

The octal number of decimal number 8 is 10

In [44]:
```
number=12
print(oct(number)[-2:])
```

```
14
```

Q.16) Display float number with 2 decimal places using print()

In [45]:
```
num=56.87547
y=float(round(num,2))
print(y)
```

```
56.88
```

Q.17) Print all factors of a given number provided by the user

In [47]:
```
n=int(input("enter the number: "))
for i in range(1,n+1):
    if n%i==0:
        print(i,end=",")
```

```
enter the number: 45
1,3,5,9,15,45,
```

Q.18) Accept a list of 5 float numbers as an input from the user

In [2]:
```
l = []

while len(l)<5:
    n = float(input("enter the float numbers : "))
    l.append(n)
print(l)
```

```
enter the float numbers : 14.23
enter the float numbers : 25.36
enter the float numbers : 10.25
enter the float numbers : 14.78
enter the float numbers : 75.96
[14.23, 25.36, 10.25, 14.78, 75.96]
```

Q.19) Write all content of a given number by skipping number 5

In [9]:
```
n = int(input("enter the number : "))

for i in range(1,n+1):
    if i == 5:
        continue
    print(i)
```

```
enter the number : 10
1
2
3
4
6
7
8
9
10
```

Q.20) Accept any three string from one input() call

In [16]:
```
n = input("enter the three names with keeping space in between: ")

sep=n.split()
print(sep)
name1=sep[0]
name2=sep[1]
name3=sep[2]
print(name1)
print(name2)
print(name3)
```

```
enter the three names with keeping space in between: sunny singh jadon
['sunny', 'singh', 'jadon']
sunny
singh
jadon
```

Q.21) Format variables using a string.format() method. Write a program to use string.format() method to format the following three variables as per the expected output

In [18]:
```
totalmoney=1200
quantity=4
price=450

print("i have {} dollars so i can buy {} football for {} dollars".format(total
```

```
i have 1200 dollars so i can buy 4 football for 450 dollars
```

Q.22) Check file is empty or not Write a program to check if the given file is empty or no

In [ ]:
```
import os

size=os.stat("test file.txt").st_size

if size==0:
    print("file is empty")
else:
    print("file is not empty")
```

Q.23) Read line number 4 from the following file

In [ ]:
```
with open("test file.txt","r") as fp:
    lines=fp.readlines()
    print(lines[3])
```

Q.24) Write a program to find the simple interest when the value of principle,rate of interest and time period is given.

In [23]:
```
P = float(input("principle amount: "))
R = float(input("Rate of interest: "))
T = float(input("for the time period: "))

Simple_interest= (P*R*T)/100

print(Simple_interest)

total_due=P+Simple_interest

print("Total due amount will need to pay wll be ",total_due)
```

```
principle amount: 140000
Rate of interest: 12
for the time period: 2
33600.0
Total due amount will need to pay wll be  173600.0
```

Q.25) Write a program to find the volume of the cylinder. Also find the cost when ,when the cost of 1litre milk is 40Rs.

In [28]:
```
import math

radius = float(input("enter the radius of cylinder in cm : "))
height = float(input("enter the height of cylinder in cm : "))

volume = math.pi*(radius**2)*height

litre = volume/1000
cost = litre*40

print("Volume of cylinder will be ",volume)
print("How much milk we can carry in this cylinder ",round(litre,2), 'ltr')
print("this cost of that milk will be",round(cost,2))
```

```
enter the radius of cylinder in cm : 145
enter the height of cylinder in cm : 25
Volume of cylinder will be  1651299.638543135
How much milk we can carry in this cylinder  1651.3 ltr
this cost of that milk will be 66051.99
```

print some patterns

5 4 3 2 1

4 3 2 1

3 2 1

2 1

1

In [4]:
```
n = int(input("no. of rows : "))

for i in range(n,0,-1):
    for j in range(i,0,-1):
        print(j, end = ' ')
    print()
```

```
no. of rows : 5
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

```
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

In [6]:
```
n = int(input("Enter no. of rows : "))

for i in range(1,n+1):
    for j in range(1,i+1):
        print('*', end = ' ')
    print()
```

```
Enter no. of rows : 5
*
* *
* * *
* * * *
* * * * *
```

In [7]:
```
n = int(input("Enter no. of rows : "))

for i in range(n,0,-1):
    for j in range(i,0,-1):
        print('*', end = ' ')
    print()
```

```
Enter no. of rows : 5
* * * * *
* * * *
* * *
* *
*
```

In [8]:
```
n = int(input("Enter no. of rows : "))

for i in range(1,n+1):
    for j in range(1,i+1):
        print('*', end = ' ')
    print()

for k in range(n,0,-1):
    for l in range(k,0,-1):
        print('*', end = ' ')
    print()
```

```
Enter no. of rows : 5
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * *
* * *
* *
*
```

In [9]:
```
n = int(input("Enter no. of rows : "))

for i in range(1,n+1):
    for j in range(1,i+1):
        print('*', end = ' ')
    print()

for k in range(n-1,0,-1):
    for l in range(k,0,-1):
        print('*', end = ' ')
    print()
```

```
Enter no. of rows : 5
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

```
    *
  * * *
* * * * *
```

In [14]:
```
n = int(input("enter no. of rows : "))

for i in range(1,n+1):
    print(' '*n,end = ' ')
    print('* '*i)
    n = n - 1
```

```
enter no. of rows : 5
      *
     * *
    * * *
   * * * *
  * * * * *
```

```
1

2 1

3 2 1

4 3 2 1

5 4 3 2 1
```

In [17]:
```
n = int(input("enter number of rows : "))

for i in range(1,n+1):
    for j in range(i,0,-1):
        print(j, end = ' ')
    print()
```

```
enter number of rows : 5
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
```

Find the sum of the series upto n terms.

Write a program to calculate the sum of series up to n term. For example, if n =5 the series will become 2 + 22 + 222 + 2222 + 22222 = 24690. Take the user input and then calculate. And the output style should match which is given in the example.

In [21]:
```
n = int(input("Enter the number of rows : "))

sum = 0
start = 2

for i in range(1,n+1):
    if i < n:
        print(start, end='+')
    else:
        print(start)

    sum = sum + start
    start = start * 10 +2

print("sum of the series is :",sum)
```

```
Enter the number of rows : 5
2+22+222+2222+22222
sum of the series is : 24690
```

Write a program to print all the unique combinations of 1,2,3 and 4

In [24]:
```
n = int(input("Enter the number of rows : "))

for i in range(1,n+1):
    for j in range(1,n+1):
        for k in range(1,n+1):
            for l in range(1,n+1):
                print(i,j,k,l)
```

```
Enter the number of rows : 2
1 1 1 1
1 1 1 2
1 1 2 1
1 1 2 2
1 2 1 1
1 2 1 2
1 2 2 1
1 2 2 2
2 1 1 1
2 1 1 2
2 1 2 1
2 1 2 2
2 2 1 1
2 2 1 2
2 2 2 1
2 2 2 2
```

Write a program that will take a decimal number as input and prints out the binary equivalent of the number

In [25]:
```
n = int(input('enter the number'))

binary = []
while n > 0:

    binary.append(n%2)
    n = n//2

for i in binary[::-1]:
    print(i,end='')
```

```
enter the number10
1010
```

Write a program that will take 2 numbers as input and prints the LCM and HCF of those 2 numbers

In [33]:
```
a = int(input("enter the first number : "))
b = int(input("enter the second number : "))

if a > b:
    greater = a
else:
    greater = b

while True:
    if(greater%a == 0 and greater%b == 0):

        lcm = greater
        break

    greater = greater + 1

print("LCM of both the numbers is :",lcm)
```

```
enter the first number : 15
enter the second number : 45
LCM of both the numbers is : 45
```

In [1]:
```
# HCF OR GCD

a = int(input("enter the first number : "))
b = int(input("enter the second number : "))

if a < b:
    smaller = a
else:
    smaller = b

for i in range(1,smaller+1):
    if(a%i==0 and b%i==0):
        HCF = i

print("HCF/GCD of both the numbers is :",HCF)
```

```
enter the first number : 15
enter the second number : 3
HCF/GCD of both the numbers is : 3
```

write a program to find GCD

In [29]:
```
a = int(input("Enter your first number: "))
b = int(input("Enter your second number: "))

original_a = a
original_b = b

while b:
    a, b = b, a % b

print("GCD of Both the numbers is :",a)
```

```
Enter your first number: 15
Enter your second number: 40
GCD of Both the numbers is : 5
```

Create Short Form from initial character

In [3]:
```
full_form = input("enter the full form as you want : ")
short_form = ''

l = full_form.split()

for i in l:
    short_form = short_form + i[0].upper()

print("short form is :",short_form)
```

```
enter the full form as you want : data base management system
short form is : DBMS
```

Append second string in the middle of first string

In [4]:
```
s1 = input('enter the 1st string : ')
s2 = input('enter the 2nd string : ')

print(s1[0:int(len(s1)/2)] + s2 + s1[int(len(s1)/2):])
```

```
enter the 1st string : sunny
enter the 2nd string : singh
susinghnny
```

Given string contains a combination of the lower and upper case letters. Write a program to arrange the characters of a string so that all lowercase letters should come first.

In [5]:
```
s = 'PyNaTive'

upper = ''
lower = ''

for i in s:
    if i.islower():
        lower = lower + i
    else:
        upper = upper + i

print(lower + upper)
```

```
yaivePNT
```

Take a alphanumeric string input and print the sum and average of the digits that appear in the string, ignoring all other characters.

In [9]:
```
s = 'hel123O4every093'

sum = 0
count = 0

for i in s:
  if i.isdigit():
    sum = sum + int(i)
    count += 1

print(sum)
print(sum/count)
print(count)
```

```
22
3.142857142857143
7
```

Removal of all characters from a string except integers

In [16]:
```
s = 'I am 25 years and 10 months old'

res = ''

for i in s:
  if i.isdigit():
    res = res + i

print(res)
```

```
2510
```

Reverse words in a given String

In [17]:
```
s = 'code practice quiz geeks'

L = []

for i in s.split():
    L.append(i)

L = L[::-1]
print(" ".join(L))
```

```
geeks quiz practice code
```

Find uncommon words from two Strings.

In [21]:
```
A = "apple banana mango"
B = "banana fruits mango"

l = []

for i in A.split():
    if i not in B and i not in l:
        l.append(i)

for j in B.split():
    if j not in A and j not in l:
        l.append(j)

print(l)
```

```
['apple', 'fruits']
```

Write a program that can remove all the duplicate characters from a string. User will provide the input.

In [24]:
```
a = 'aaabbbbccccddddeeeffffffgggghhhhhiiiiijjjjjkkkkkkllllmmmnnnnnooooooooppppp

b = ''

for i in a:
    if i not in b:
        b = b + i

print(b)
```

```
abcdefghijklmnopqrstuvwxyz
```

In [28]:
```
print('*'*127)
```

```
***************************************************************************************
***********************************************
```

Combine two lists index-wise(columns wise)

In [4]:
```
list1 = ["M", "na", "i", "Kh"]
list2 = ["y", "me", "s", "an"]

[[i,j] for (i,j) in zip(list1,list2)]
```

Out[4]:
```
[['M', 'y'], ['na', 'me'], ['i', 's'], ['Kh', 'an']]
```

Add new item to list after a specified item

Write a program to add item 7000 after 6000 in the following Python List

In [9]:
```
list1 = [10, 20, [300, 400, [5000, 6000], 500], 30, 40]
list1[2][2].append(7000)
list1
```

Out[9]:
```
[10, 20, [300, 400, [5000, 6000, 7000], 500], 30, 40]
```

Update no of items available

In [1]:
```
candy_list = ['Jelly Belly','Kit Kat','Double Bubble','Milky Way','Three Muske
no_of_items = [10,20,34,74,32]

for (i,j) in zip(candy_list,no_of_items):
    print(i,'-',j)
```

```
Jelly Belly - 10
Kit Kat - 20
Double Bubble - 34
Milky Way - 74
Three Musketeers - 32
```

Running Sum on list

In [19]:
```
l = [1,2,3,4,5,6,7,8,9]

sum = 0
L = []

for i in l:
    sum = sum + i
    L.append(sum)

print(L)
```

```
[1, 3, 6, 10, 15, 21, 28, 36, 45]
```

Split String of list on K character.

In [28]:
```
L = ['CampusX is a channel', 'for data-science', 'aspirants.']

s = []

for i in L:
    s.extend(i.split())
print(s)
```

```
['CampusX', 'is', 'a', 'channel', 'for', 'data-science', 'aspirants.']
```

Convert Character Matrix to single String using string comprehension.

In [35]:
```
l = [['c', 'a', 'm', 'p', 'u', 'x'], ['i', 's'], ['b', 'e', 's', 't'], ['c', '
" ".join(["".join(i) for i in l])
```

Out[35]:
```
'campux is best channel'
```

Write a program that can perform union operation on 2 lists

In [36]:
```
L1 = [1,2,3,4,5,1]
L2 = [2,3,5,7,8]

union = []

for i in L1:
    if i not in union:
        union.append(i)

for j in L2:
    if j not in union:
        union.append(j)

print(union)
```

```
[1, 2, 3, 4, 5, 7, 8]
```

Check is tuples are same or not?

In [39]:
```
t1 = (1,2,3,0)
t2 = (1,2,3,0)

flag = True
for i,j in zip(t1,t2):

    if i == j:
        continue
    else:
        flag = False
        break

if flag:
    print('same')
else:
    print('not same')
```

```
same
```

Write a program to find set of common elements in three lists using sets.

Input : ar1 = [1, 5, 10, 20, 40, 80] ar2 = [6, 7, 20, 80, 100] ar3 = [3, 4, 15, 20, 30, 70, 80, 120]

Output : [80, 20]

In [40]:
```
ar1 = [1, 5, 10, 20, 40, 80]
ar2 = [6, 7, 20, 80, 100]
ar3 = [3, 4, 15, 20, 30, 70, 80, 120]

s1 = set(ar1)
s2 = set(ar2)
s3 = set(ar3)

result = list((s1 & s2) & s3)
print(result)
```

```
[80, 20]
```

Count no of tuples, list and set from a list

In [2]:
```
list1 = [{'hi', 'bye'},{'Geeks', 'forGeeks'},('a', 'b'),['hi', 'bye'],['a', 'b
l = [0,0,0]

for i in list1:
    if type(i) == list:
        l[0] = l[0] + 1
    elif type(i) == set:
        l[1] = l[1] + 1
    elif type(i) == tuple:
        l[2] = l[2] + 1
    else:
        pass

print('Lists-{}\nSets-{}\nTuples-{}'.format(l[0],l[1],l[2]))
```

```
Lists-2
Sets-2
Tuples-1
```

print number of vowels in the given sentence

In [4]:
```
vowels = set('aeiouAEIOU')

s = set("hands-on data science mentorship prograAm with live classes at afforda

print('No of unique vowels-',len(s & vowels))
```

```
No of unique vowels- 6
```

Write a program to Check if a given string is binary string of or not.

In [10]:
```
# A string is said to be binary if it's consists of only two unique characters

s = input("Enter your string : ")

if len(set(s)) == 2:
    print("Binary")
else:
    print("Not binary")
```

```
Enter your string : 1010101010101
Binary
```

find union of n arrays

In [16]:
```
l = [[1, 2, 2, 4, 3, 6],
     [5, 1, 3, 4],
     [9, 5, 7, 1],
     [2, 4, 1, 3]]

s = set()

for i in l:
    s.update(i)
print(s)
```

```
{1, 2, 3, 4, 5, 6, 7, 9}
```

Intersection of two lists.

In [23]:
```
# use list comprehension

list1 = [1,2,3,4,2,2,3,4,2,8,7,8,8,9]
list2 = [1,3,2,2,2,2,2,4,5,3,9,0,2,3]

[i for i in list1 if i in list2]
```

Out[23]:
```
[1, 2, 3, 4, 2, 2, 3, 4, 2, 9]
```

Key with maximum unique values

In [24]:
```
test_dict = {"CampusX" : [5, 7, 7, 7, 7], "is" : [6, 7, 7, 7], "Best" : [9, 9,

max_val = 0
max_key = ''
for i in test_dict:
    if max_val < len(set(test_dict[i])):
        max_val = len(set(test_dict[i]))
        max_key = i

print(max_key)
```

```
Best
```

Convert a list of Tuples into Dictionary.

In [2]:
```
L1 = [("akash", 10), ("gaurav", 12), ("anand", 14), ("suraj", 20), ("akhil", 2
L = [('A', 1), ('B', 2), ('C', 3)]
dict_ = {}

for i,j in L1:
    dict_[i] = j

print(dict_)
```

```
{'akash': 10, 'gaurav': 12, 'anand': 14, 'suraj': 20, 'akhil': 25, 'ashish':
30}
```

Sort Dictionary key and values List.

In [38]:
```
d = {'c': [45,3,27], 'b': [12, 10], 'a': [19, 4]}

res = {}

for i in sorted(d):
    res[i] = sorted(d[i])

print(res)
```

```
{'a': [4, 19], 'b': [10, 12], 'c': [3, 27, 45]}
```

In [39]:
```
sorted(d)
```

Out[39]:
```
['a', 'b', 'c']
```

In [42]:
```
sorted(d['c'])
```

Out[42]:
```
[3, 27, 45]
```

Convert List to List of dictionaries. Given list values and keys list, convert these values to key value pairs in form of list of dictionaries.

In [46]:
```
test_list = ["CampusX", 10]
key_list = ["name", "id"]

n = len(test_list)

res = []

for i in range(0,n,2):
     res.append({key_list[0]: test_list[i],key_list[1]:test_list[i+1]})

print(res)
```

```
[{'name': 'CampusX', 'id': 10}]
```

Replace words from Dictionary. Given String, replace it's words from lookup dictionary.

EX :

s1= 'CampusX best for DS students.'

s2 = {"best" : "is the best channel", "DS" : "Data-Science"}

output = CampusX is the best channel for Data-Science students.

In [7]:
```
s1= 'CampusX best for DS students.'
s2 = {"best" : "is the best channel", "DS" : "Data-Science"}

result = []

for i in s1.split():
    if i in s2:
        result.append(s2[i])
    else:
        result.append(i)

print(result)
print('*'*127)
print(" ".join(result))
```

```
['CampusX', 'is the best channel', 'for', 'Data-Science', 'students.']
***************************************************************************************************
***********************************************
CampusX is the best channel for Data-Science students.
```

SOME PYTHON FUNCTION PROBLEMS****

1) Write a Python function that takes a list and returns a new list with unique elements of the first list.

In [13]:
```
def unique_Value(a):
    result = []
    for i in l:
        if i not in result:
            result.append(i)
    return result

l= [1,2,3,3,3,3,4,5]
unique_Value(l)
```

Out[13]:
```
[1, 2, 3, 4, 5]
```

2) Write a Python function that accepts a hyphen-separated sequence of words as parameter and returns the words in a hyphen-separated sequence after sorting them alphabetically.

In [19]:
```
def sort_sequence(seq):
    temp = []

    for i in sorted(seq.split('-')):
         temp.append(i)

    return "-".join(temp)

s = 'green-red-yellow-black-white'
sort_sequence(s)
```

Out[19]: 'black-green-red-white-yellow'

3) Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.

In [26]:
```
def lower_upper(s):
    lower_count = 0
    upper_count = 0

    for i in s:
        if i.islower():
            lower_count += 1
        elif i.isupper():
            upper_count += 1

    return lower_count, upper_count

# Test the function
s = 'CampusX is an Online Mentorship Program fOr EnginEering studentS.'
x, y = lower_upper(s)
print('No. of Lower case characters:', x)
print('No. of Upper case Characters:', y)
```

```
No. of Lower case characters: 47
No. of Upper case Characters: 9
```

4) Write a Python program to print the even numbers from a given list.

In [28]:
```
def even_num(l):
    result = []

    for i in l:
        if i%2 == 0:
            result.append(i)
        else:
            pass
    return result

l = [12,3,2,4,6,78,789]
even_num(l)
```

Out[28]: [12, 2, 4, 6, 78]

4) Write a Python function to check whether a number is perfect or not.

In [ ]:
```
The first perfect number is 6, because 1, 2, and 3 are its proper positive div
Equivalently, the number 6 is equal to half the sum of all its positive divisc

The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the pe
```

In [1]:
```
def perfect_num(num):
    sum_divisors = 0
    for i in range(1, num):
        if num % i == 0:
            sum_divisors += i

    return sum_divisors == num

# Example usage
print(perfect_num(29))
print(perfect_num(28))
```

```
False
True
```

5) Write a Python function to concatenate any no of dictionaries to create a new one.

In [4]:
```
def sunny(*kwargs):
    d = {}

    for i in kwargs:
        d.update(i)
    return d

d1 = {'a' : 1, 'b' : 2, 'c' : 30}
d2 = {'e' : 4, 'f' : 5}
d3 = {'g' : 6}
d4 = {'h' : 7}

sunny(d1,d2,d3,d4)
```

Out[4]:
```
{'a': 1, 'b': 2, 'c': 30, 'e': 4, 'f': 5, 'g': 6, 'h': 7}
```

6) Write a python function that accepts a string as input and returns the word with most occurence.

In [17]:
```
def most_used(s):
    d = {}

    for i in s.split():
        if i not in d:
            d[i] = 1
        else:
            d[i] = d[i] + 1

        max_value = max(d.values())

    for i in d:
        if d[i] == max_value:
            print(i, '->', d[i])
            break

most_used("hi hi hi how are you you you are are very beautiful are")
```

```
are -> 4
```

7) Write a Python program to add three given lists using Python map and lambda.

In [18]:
```
L1 = [1,2,3]
L2 = [4,5,6]
L3 = [7,8,9]

# [12,15,18]

list(map(lambda x,y,z:x+y+z,L1,L2,L3))
```

Out[18]:
```
[12, 15, 18]
```

8) Use reduce to convert a 2D list to 1D

In [19]:
```
l = [[1, 2, 3],
    [3, 6, 7],
    [7, 5, 4]]

import functools
functools.reduce(lambda x,y:x+y,l)
```

Out[19]:
```
[1, 2, 3, 3, 6, 7, 7, 5, 4]
```

9) A dictionary contains following information about 5 employees:

Write a program using map/filter/reduce to a list of employees(first name + last name) who are highly skilled

In [20]:
```
employees = [
    {
        'fname':'Nitish',
        'lname':'Singh',
        'age' : 33,
        'grade':'skilled'
    },
    {
        'fname':'Ankit',
        'lname':'Verma',
        'age' : 34,
        'grade':'semi-skilled'
    },
    {
        'fname':'Neha',
        'lname':'Singh',
        'age' : 35,
        'grade':'highly-skilled'
    },
    {
        'fname':'Anurag',
        'lname':'Kumar',
        'age' : 30,
        'grade':'skilled'
    },
    {
        'fname':'Abhinav',
        'lname':'Sharma',
        'age' : 37,
        'grade':'highly-skilled'
    }

]
```

In [21]:
```
employees
```

Out[21]:
```
[{'fname': 'Nitish', 'lname': 'Singh', 'age': 33, 'grade': 'skilled'},
 {'fname': 'Ankit', 'lname': 'Verma', 'age': 34, 'grade': 'semi-skilled'},
 {'fname': 'Neha', 'lname': 'Singh', 'age': 35, 'grade': 'highly-skilled'},
 {'fname': 'Anurag', 'lname': 'Kumar', 'age': 30, 'grade': 'skilled'},
 {'fname': 'Abhinav', 'lname': 'Sharma', 'age': 37, 'grade': 'highly-skille
d'}]
```

In [24]:
```
list(map(lambda x : x['fname'] + ' ' + x['lname'],
list(filter(lambda x : True if x['grade'] == 'highly-skilled' else False, empl
```

Out[24]:
```
['Neha Singh', 'Abhinav Sharma']
```

10) Using filter() and list() functions and .lower() method filter all the vowels in a given string.

In [25]:
```
str1="FIFA world cup in 2022 will take place in Qatar"

list(filter(lambda x : True if x.lower() in 'aeiou' else False, str1))
```

Out[25]:
```
['I', 'A', 'o', 'u', 'i', 'i', 'a', 'e', 'a', 'e', 'i', 'a', 'a']
```

Print the sum of the current number and the previous number Write a program to iterate the first 10 numbers and in each iteration, print the sum of the current and previous number.

Expected result: 0,1,3,5,7,9,11,13,15,17

In [5]:
```
n = int(input("enter the number : "))
previous_num = 0

for i in range(0,n):
    sum = previous_num + i
    previous_num = i
    print(sum, end = ',')
```

```
enter the number : 10
0,1,3,5,7,9,11,13,15,17,
```

Write a python program to search a given number from a list

In [10]:
```
l = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]

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

for i in l:
    if i == n:
        print(n, "is present in the list")
        break
else:
    print(n, "is not present in the list")
```

```
enter the number : 5
5 is present in the list
```

check palindrome number

In [13]:
```
n = int(input("enter the number : "))

rev_num = str(n)[::-1]

if n == int(rev_num):
    print("the number is palindrome")
else:
    print("not palindrome")
```

```
enter the number : 545
the number is palindrome
```

Create a new list from a two list using the following condition Given a two list of numbers, write a program to create a new list such that the new list should contain odd numbers from the first list and evennumbers from the second list.

In [14]:
```
l1 = [1,2,3,4,5,6]
l2 = [7,8,9,10,11,12]
l3 = []

for i in l1:
    if i%2 != 0:
        l3.append(i)
for j in l2:
    if j%2 == 0:
        l3.append(j)

print(l3)
```

```
[1, 3, 5, 8, 10, 12]
```

Print multiplication table form 1 to 10

In [15]:
```
for i in range(1,11):
    for j in range(1,11):
        print(i*j, end = ' ')
    print()
```

```
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
3 6 9 12 15 18 21 24 27 30
4 8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
```

The current population of a town is 10000. The population of the town is increasing at the rateof 10% per year. You have to write a program to find out the population at the end of each of the last 10years. For eg current population is 10000 so the output should be like this:

#10th year - 10000 #9th year - 9000 #8th year - 8100 and so on

In [21]:
```
p=10000

for i in range(1,10):
    p=p-0.1*p
    print(round(p),end=" ")
```

```
9000 8100 7290 6561 5905 5314 4783 4305 3874
```

Write a program that take a user input of three angles and will find out whether it can form a triangle or not

In [22]:
```
a=float(input("1st angle of triangle: "))
b=float(input("2nd angle of triangle: "))
c=float(input("3rd angle of triangle: "))

if (a+b+c)==180 and a!=0 and b!=0 and c!=0:
    print("yes it can form a triangle")
else:
    print("No it cant form a triangle")
```

```
1st angle of triangle: 100
2nd angle of triangle: 40
3rd angle of triangle: 40
yes it can form a triangle
```

Write a program that will determine weather when the value of temperature and humidity is

In [23]:
```
temp = int(input("enter the value of temp "))
humidity = int(input("enter the value of humidity "))

if temp>=30 and humidity>=90:
    print("Hot and Humid")
elif temp>=30 and humidity<90:
    print("Hot")
elif temp<30 and humidity>=90:
    print("cool and humid")
else:
    print("cool")
```

```
enter the value of temp 40
enter the value of humidity 50
Hot
```

Write a program that will give you the in hand salary after deduction of HRA(10%),

DA(5%),

PF(3%),

and tax

(if salary is between 5-10 lakh–10%),

(11-20lakh–20%),

(20< _ – 30%)

(0-1lakh print k).

In [27]:
```
salary = int(input("enter your salary : "))

if salary in range(500000,1000000):
    salary = salary - salary*0.1
elif salary in range(1100000,2000000):
    salary = salary - salary*0.2
elif salary > 2000000:
    salary = salary - salary*0.3
else:
    print("no tax")

print("Salary after tax cutting",salary)

HRA = salary*0.1
DA = salary*0.05
PF = salary*0.03

remain_salary = salary - (HRA + DA + PF)

if remain_salary in range(1000,99999):
    print("in hand salary after tax/HRA/DA/PF cutting",remain_salary/1000,"k")
elif remain_salary in range(100000,9999999):
    print("in hand salary after tax/HRA/DA/PF cutting",remain_salary/100000,"l
else:
    print("in hand salary after tax/HRA/DA/PF cutting",remain_salary/10000000,
```

```
enter your salary : 500000
Salary after tax cutting 450000.0
in hand salary after tax/HRA/DA/PF cutting 3.69 lakh
```

Write a program that can multiply 2 numbers provided by the user without using the * operatoR

In [7]:
```
n1 = int(input("enter the first number : "))
n2 = int(input("enter the second number : "))
result = 0

for i in range(0,n2):
    result = result + n1

print(result)
```

```
enter the first number : 5
enter the second number : 4
20
```

Write a program that can find the factorial of a given number provided by the user

In [9]:
```
n=int(input("enter the number : "))
factorial=1

if n < 0 :
    print("factorial dont exist for negative number")
elif n==0:
    print("for zero factorial is always one")
else:
    for i in range(1,n+1):
        factorial=factorial*i
    print(factorial)
```

```
enter the number : 5
120
```

Calculate the sum of all numbers from 1 to a given number

In [10]:
```
n = int(input("enter the number : "))

sum = 0

for i in range(1,n+1):
    sum = sum + i
print(sum)
```

```
enter the number : 10
55
```

Write a program to print multiplication table of a given number

In [15]:
```
n = int(input("enter the number : "))

for i in range(1,11):
    result = n*i

    print(n,'*',i,'=',result)
```

```
enter the number : 10
10 * 1 = 10
10 * 2 = 20
10 * 3 = 30
10 * 4 = 40
10 * 5 = 50
10 * 6 = 60
10 * 7 = 70
10 * 8 = 80
10 * 9 = 90
10 * 10 = 100
```

Display numbers from a list using loop Write a program to display only those numbers from a list that satisfy the following conditions

The number must be divisible by five

If the number is greater than 150, then skip it and move to the next number

If the number is greater than 500, then stop the loo

In [19]:
```
n = [12,75,150,180,145,525,50]

for i in n:
    if i>500:
        break
    elif i>150:
        continue
    else:
        if i%5==0:
            print(i)
```

```
75
150
145
```

Count the total number of digits in a number Write a program to count the total number of digits in a number using a while loop.

In [23]:
```
n = input("enter the number : ")

x = list(n)
count = 0

while count < len(x):
    for i in x:
        count = count + 1
    print(count)
```

```
enter the number : 25478524
8
```

Calculate the cube of all numbers from 1 to a given number Write a program to rint the cube of all numbers from 1 to a given number

In [26]:
```
n = int(input("give a number : "))

for i in range(1,n+1):
    x = i**3
    print(x)
```

```
give a number : 10
1
8
27
64
125
216
343
512
729
1000
```

Write a program to print the first 25 odd number

In [2]:
```
flag = 0
i = 1
odd_list = []

while True:
    if i % 2 != 0:
        odd_list.append(i)
        flag = flag + 1
    elif flag == 25:
        break
    i = i + 1

print(odd_list)
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 4
1, 43, 45, 47, 49]
```

Write a function called exponent(base, exp) that returns an int value of base raises to the power of exp. Note here exp is a non-negative integer, and the base is an integer.

In [3]:
```
def exponent(base,exp):
    result=pow(base,exp)
    print(result)
exponent(10,2)
```

```
100
```

Check if the first and last number of a list is the same Write a function to return True if the first and last number of a given list is same. If numbers are different then return False

In [5]:
```
l = [1,2,3,4,5,4,3,5,3,6,7,5,1]

n1 = l[0]
n2 = l[-1]

def same(list):
    if n1 == n2:
        return True
    else:
        return False

same(l)
```

Out[5]: True

Calculate the multiplication and sum of two numbers

Given two integer numbers return their product only if the product is equal to or lower than 1000, else return their sum.

In [7]:
```
n1 = int(input("enter the first number : "))
n2 = int(input("enter the second nummber : "))

def sumMUL(n,m):
    multi = n1*n2
    summ = n1+n2

    if multi < 1000:
        return multi
    else:
        return summ

sumMUL(n1,n2)
```

```
enter the first number : 100
enter the second nummber : 10
```

Out[7]: 110

Create a function with a default argument Write a program to create a function show_employee() using the following conditions.

It should accept the employee's name and salary and display both. If the salary is missing in the function call then assign default value

In [8]:
```
def show_employee(name, salary = 9000):
    print('employee name is {} and his/her salary is {}'.format(name,salary))
```

In [13]:
```
show_employee("sunny",100000)
```

```
employee name is sunny and salary is 100000
```

In [14]:
```
show_employee("sunny")
```

```
employee name is sunny and salary is 9000
```

Create an inner function to calculate the addition in the following way

Create an outer function that will accept two parameters, a and b

Create an inner function inside an outer function that will calculate the addition of a and b

At last, an outer function will add 5 into addition and return it

In [15]:
```
def outerfunc(n1,n2):
    def innerfunc(n1,n2):
        return n1+n2
    add=innerfunc(n1,n2)
    return add+5
```

In [16]:
```
outerfunc(5,10)
```

Out[16]:
```
20
```

Write a program to create a recursive function to calculate the sum of numbers from 0 to 10

In [18]:
```
def addition(num):
    if num==0:
        return 0
    else:
        return num+addition(num-1)
```

In [19]:
```
addition(10)
```

Out[19]: 55

Write a program to create a recursive function to calculate the factorial of numbers from 0 to 10

In [20]:
```
def factorial(num):
    if (num==1):
        return 1
    else:
        return num*factorial(num-1)
```

In [21]:
```
factorial(5)
```

Out[21]: 120

count of a given substring from a string

In [22]:
```
sentence="radha is most beutiful,radha is queen of vraj,radha is most beloved
x=sentence.count("radha")
print(x)
```

```
3
```

Print characters from a string that are present at an even index number Write a program to accept a string from the user and display characters that are present at an even index number.

In [24]:
```
string=input("enter the text: ")
x=list(string)
print(string)
for i in x[0::2]:
    print(i,end=" ")
```

```
enter the text: Sunny Singh Jadon
Sunny Singh Jadon
S n y S n h J d n
```

Remove first n characters from a string Write a program to remove characters from a string starting from zero up to n and return a new string

In [26]:
```
def remove (word,n):
    x=len(word)
    p=list(word)

    for i in p:
        if n<=x:
            z=word[n:]
    print(z)

remove("pynative",2)
```

```
native
```

Create a string made of the first, middle and last character

In [27]:
```
name="Sunny Singh Jadon"

a=name[0]
print(a)

c=name[-1]
print(c)

l=len(name)
x=int(l//2)

b=name[x]

print(b)
"".join([a,b,c])
```

```
S
n
n
```

Out[27]: 'Snn'

Find the index position of a particular character in another string

In [28]:
```
a=input("enter the text: ")
b=input("enter the character: ")
print(a.index(b))
```

```
enter the text: sunny singh jadon
enter the character: g
9
```

Write a program which can remove a particular character from a string

In [29]:
```
a=input("enter the string: ")
b=input('enter the character you  want to remove: ')

a=a.replace(b,"")
print(a)
```

```
enter the string: sunny singh jadon
enter the character you  want to remove: g
sunny sinh jadon
```

Find all occurrences of a substring in a given string by ignoring the case Write a program to

find all occurrences of “USA” in a given string ignoring the case.

str1 = "Welcome to USA. usa awesome, isn't it?"

expected ans --> USA:-->2

In [30]:
```
str1 = "Welcome to USA. usa awesome, isn't it?"
str2=str1.upper()
print(str2)
str2.count("USA")
```

```
WELCOME TO USA. USA AWESOME, ISN'T IT?
```

Out[30]:
```
2
```

Write a python program to convert a string to title case without using the title()

In [34]:
```
str1 = input("enter your string : ")

a = str1.split()
result = ''

for i in a:
    result = result + i.capitalize() + " "
print(result)
```

```
enter your string : sunny singh jadon
Sunny Singh Jadon
```

Calculate the sum and average of the digits present in a string Given a string s1, write a program to return the sum and average of the digits that appear in the string, ignoring all other characters.

In [36]:
```
str1 = "PYnative29@#8496"
str2=list(str1)

total=0
counter=0

for i in str2:
    if i.isdigit()==True:
        total=total+int(i)
    counter+=1
print("sum of digits in the given string is",total)
print("avg of digits in the given string is",round(total/counter,2))
```

```
sum of digits in the given string is 38
avg of digits in the given string is 2.38
```

Remove empty strings from a list of strings

In [2]:
```
str_list = ["Emma", "Jon", "", "Kelly", None, "Eric", ""]

for i in str_list:
    if i == "" or i==None:
        str_list.remove(i)
print(str_list)
```

```
['Emma', 'Jon', 'Kelly', 'Eric']
```

Remove special symbols / punctuation from a string

In [6]:
```
sentence = "/*Jon is @developer & musician"
import re

clean_sentence=re.sub('[^A-Za-z0-9\s]+',"",sentence)
print(clean_sentence)
```

```
Jon is developer  musician
```

Removal all characters from a string except integers

In [15]:
```
s = "I am 22 years and 06 months old"
l = []

for i in s.split():
    if i.isdigit() == True:
        l.append(i)

print("".join(l))
```

```
2206
```

extract all the emailid for the given string

In [18]:
```
s = "hi i am sunnysinghjadon2001@gmail.com and sunnysingh2002#gmail.com thanki
l = []

for i in s.split():
    if ".com" in i:
        l.append(i)
print(" ".join(l))
```

```
sunnysinghjadon2001@gmail.com sunnysingh2002#gmail.com
```

In [19]:
```
s = "hi i am sunnysinghjadon2001@gmail.com and sunnysingh2002#gmail.com thanki
l = []

for i in s.split():
    if ".com" in i:
         print(i)
```

```
sunnysinghjadon2001@gmail.com
sunnysingh2002#gmail.com
```

Write a program to remove the item present at index 4 and add it to the 2nd position and at the end of the list

In [26]:
```
l = [1,2,3,4,5,6,7,8,9,10]

l.pop(4)
print(l)

l.insert(2,5)
print(l)

l.append(5)
print(l)
```

```
[1, 2, 3, 4, 6, 7, 8, 9, 10]
[1, 2, 5, 3, 4, 6, 7, 8, 9, 10]
[1, 2, 5, 3, 4, 6, 7, 8, 9, 10, 5]
```

Slice list into 3 equal chunks and reverse each chunk

In [27]:
```
sample_list = [11,49,8,23,14,12,78,45,89]

n=int(len(sample_list)/3)

l1=sample_list[0:n]
l2=sample_list[n:n*2]
l3=sample_list[n*2:n*3]

print("chunk 1",l1)
print("Reversed of chunk 1",l1[::-1])

print("chunk 2",l2)
print("Reversed of chunk 2",l2[::-1])

print("chunk 3",l3)
print("Reversed of chunk 3",l3[::-1])
```

```
chunk 1 [11, 49, 8]
Reversed of chunk 1 [8, 49, 11]
chunk 2 [23, 14, 12]
Reversed of chunk 2 [12, 14, 23]
chunk 3 [78, 45, 89]
Reversed of chunk 3 [89, 45, 78]
```

extract all the values from key value pair

In [32]:
```
d = {'jan': 47, 'feb': 52, 'march': 47, 'April': 44, 'May': 52, 'June': 53, 'j
l = []

for i in d.values():
    if i not in l:
            l.append(i)
print(l)
```

```
[47, 52, 44, 53, 5]
```

Remove duplicates from a list and create a tuple and find the minimum and maximum number

In [33]:
```
sample_list = [87, 45, 41, 65, 94, 41, 99, 94]

list1=set(sample_list)  # removed all the duplicates
tuple1=tuple(list1)

print("Unique element in the sample list:\n",tuple1)
print("Maximum number in the sample list:\n",max(tuple1))
print("Minimun number in the sample list:\n",min(tuple1))
```

```
Unique element in the sample list:
 (65, 99, 41, 45, 87, 94)
Maximum number in the sample list:
 99
Minimun number in the sample list:
 41
```

Write a python program to find the max item from a list without using the max function

In [38]:
```
l1=[4,6,2,8,1]
l1.sort(reverse=True)
print(l1[0])
```

```
8
```

Take a number from the user and find the number of digits in it

In [39]:
```
n=int(input("enter the number: "))

digit_list=list(map(int,str(n)))

print(digit_list)
print(len(digit_list))
```

```
enter the number: 25558745
[2, 5, 5, 5, 8, 7, 4, 5]
8
```

Concatenate two lists index-wise

In [40]:
```
list1 = ["M", "na", "i", "Ke"]
list2 = ["y", "me", "s", "lly"]

list3=[i+j for i,j in zip(list1,list2)]
print(list3)
```

```
['My', 'name', 'is', 'Kelly']
```

```
Concatenate two lists in the following order
list1 = ["Hello ", "take "]
list2 = ["Dear", "Sir"]
Expected result:
['Hello Dear', 'Hello Sir', 'take Dear', 'take Sir']
```

In [41]:
```
list1 = ["Hello ", "take "]
list2 = ["Dear", "Sir"]

for i in list1:
    for j in list2:
        print(i+j)
```

```
Hello Dear
Hello Sir
take Dear
take Sir
```

In [43]:
```
result = [i+j for i in list1 for j in list2]
print(result)
```

```
['Hello Dear', 'Hello Sir', 'take Dear', 'take Sir']
```

Given a two Python list. Write a program to iterate both lists simultaneously and display items from list1 in original order and items from list2 in reverse order

In [44]:
```
list1 = [10, 20, 30, 40]
list2 = [100, 200, 300, 400]

for x,y in zip(list1,list2[::-1]):
        print(x,y)
```

```
10 400
20 300
30 200
40 100
```

Remove empty strings from the list of strings

In [45]:
```
list1 = ["Mike", "", "Emma", "Kelly", "", "Brad"]
result=list(filter(lambda x: x!="",list1))
print(result)
```

```
['Mike', 'Emma', 'Kelly', 'Brad']
```

convert list to dictionary

In [46]:
```
keys = ['Ten', 'Twenty', 'Thirty']
values = [10, 20, 30]

my_dict=zip(keys,values)
result=dict(my_dict)
print(result)
```

```
{'Ten': 10, 'Twenty': 20, 'Thirty': 30}
```

merge two python dictionary to one

In [47]:
```
dict1 = {'Ten': 10, 'Twenty': 20, 'Thirty': 30}
dict2 = {'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
dict3={**dict1,**dict2}
print(dict3)
```

```
{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
```

In [48]:
```
 # another way of achieving same is:
dict1 = {'Ten': 10, 'Twenty': 20, 'Thirty': 30}
dict2 = {'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
dict3=dict1.copy()
dict3.update(dict2)
print(dict3)
```

```
{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
```

In [51]:
```
# extract marks of history

sampleDict = {"class": {"student": {"name": "Mike","marks": {"physics": 70,"hi
sampleDict["class"]["student"]["marks"]["history"]
```

Out[51]:
```
80
```

Initialize dictionary with default values In Python, we can initialize the keys with the same values

In [53]:
```
employees = ['Kelly', 'Emma']
defaults = {"designation": 'Developer', "salary": 8000}

result=dict.fromkeys(employees,defaults)

print(result)
print("="*125)
print("Details of kelly: ",result["Kelly"])
```

```
{'Kelly': {'designation': 'Developer', 'salary': 8000}, 'Emma': {'designatio
n': 'Developer', 'salary': 8000}}
================================================================================
================================================
Details of kelly:  {'designation': 'Developer', 'salary': 8000}
```

```
Write a Python program to create a new dictionary by extracting the
mentioned keys from the below dictionary.

sample_dict = { "name": "Kelly", "age": 25, "salary": 8000, "city": "New
york"}
key to extract keys = ["name", "salary"]

expected output : {'name': 'Kelly', 'salary': 8000}
```

In [56]:
```
sample_list = {
    "name" : "Kelly",
    "age" : 25,
    "salary" : 8000,
    "city" : "New York"
}

keys = ["name", "salary"]
result = dict()

for i in keys:
    result.update({i : sample_list[i]})

print(result)
```

```
{'name': 'Kelly', 'salary': 8000}
```

In [57]:
```
#using dictionary comprehension

sample_dict = {
       "name": "Kelly",
       "age": 25,
       "salary": 8000,
       "city": "New york"
}

keys=["name","salary"]

new_dict={k:sample_dict[k] for k in keys}

print(new_dict)
```

```
{'name': 'Kelly', 'salary': 8000}
```

rename keys value

In [58]:
```
sample_dict = {
       "name": "Kelly",
       "age": 25,
       "salary": 8000,
       "city": "New york"
}

sample_dict["location"]=sample_dict.pop("city")
print(sample_dict)
```

```
{'name': 'Kelly', 'age': 25, 'salary': 8000, 'location': 'New york'}
```

In [59]:
```
# check 200 is there or not in the dictionary

sample_dict = {'a': 100, 'b': 200, 'c': 300}

if 200 in sample_dict.values():
     print("200 is present in given dict")
```

```
200 is present in given dict
```

Get the key of a minimum value from the following dictionary

sample_dict = { 'Physics': 82, 'Math': 65, 'history': 75 }

In [68]:
```
sample_dict = { 'Physics': 82, 'Math': 65, 'history': 75 }

l = []

min(sample_dict,key=sample_dict.get)
```

Out[68]:
```
'Math'
```

In [65]:
```
sample_dict.get
```

Out[65]:
```
<function dict.get(key, default=None, /)>
```

Write a Python program to change Brad's salary to 8500 in the following dictionary.

sample_dict = { 'emp1': {'name': 'Jhon', 'salary': 7500}, 'emp2': {'name': 'Emma', 'salary': 8000}, 'emp3': {'name': 'Brad', 'salary': 500} }

In [72]:
```
sample_dict = { 'emp1': {'name': 'Jhon', 'salary': 7500},
                'emp2': {'name': 'Emma', 'salary': 8000},
                'emp3': {'name': 'Brad', 'salary': 500}
}

sample_dict['emp3']['salary'] = 8500
sample_dict
```

Out[72]:
```
{'emp1': {'name': 'Jhon', 'salary': 7500},
 'emp2': {'name': 'Emma', 'salary': 8000},
 'emp3': {'name': 'Brad', 'salary': 8500}}
```

In [74]:
```
# add one dict and list into a dict

sample_set = {"Yellow", "Orange", "Black"}
sample_list = ["Blue", "Green", "Red"]

sample_set.update(sample_list)
print(sample_set)
```

```
{'Orange', 'Red', 'Blue', 'Black', 'Green', 'Yellow'}
```

Check if two sets have any elements in common. If yes, display the common elements

In [75]:
```
set1 = {10, 20, 30, 40, 50}
set2 = {60, 70, 80, 90, 10}

if set1.isdisjoint(set2):
    print("above two set dont have any common element")
else:
    set1.intersection(set2)
    print(set1.intersection(set2))
```

```
{10}
```

```
Remove items from set1 that are common to both set1 and set2
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
Expected output:{30,40, 50}
```

In [81]:
```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}

set1.intersection_update(set2)
set1
```

Out[81]:
```
{30, 40, 50}
```

```
Sort a tuple of tuples by 2nd item
tuple1 = (('a', 23),('b', 37),('c', 11), ('d',29))
Expected:(('c', 11), ('a', 23), ('d', 29), ('b', 37))
```

In [4]:
```
tuple1 = (('a', 23),('b', 37),('c', 11), ('d',29))
sorted(list(tuple1),key=lambda x:x[1])
```

Out[4]:
```
[('c', 11), ('a', 23), ('d', 29), ('b', 37)]
```

In [10]:
```
t = (45, 45, 45, 45)

all_same=all(i==t[0] for i in t)

if all_same:
    print("same")
else:
    print("different")
```

```
same
```

Print current date and time in Python

In [14]:
```
import datetime

# for date and time
print(datetime.datetime.now())

# for time only
print(datetime.datetime.now().time())
```

```
2024-07-21 10:16:24.688656
10:16:24.688656
```

In [15]:
```
from datetime import datetime

given_date = datetime(2001, 1, 29)

given_date.strftime("%A %d %b %Y")
```

Out[15]:
```
'Monday 29 Jan 2001'
```

In [16]:
```
given_date = datetime(2020, 2, 25)
string_date=given_date.strftime("%Y-%m-%d %H:%M:%S")
string_date
```

Out[16]:
```
'2020-02-25 00:00:00'
```

Calculate number of days between two given dates

In [20]:
```
# 2020-02-25
date_1 = datetime(2020, 2, 25).date()

# 2020-09-17
date_2 = datetime(2020, 9, 17).date()

delta=None

if date_1>date_2:
    delta=date_1-date_2
else:
    delta=date_2-date_1
print("Difference of the days will be :",delta.days)
```

```
Difference of the days will be : 205
```

Write a Python program to create a Vehicle class with max_speed and mileage instance attributes

In [21]:
```
class Vehicle:

     def __init__ (self,max_speed,mileage):
            self.max_speed=max_speed
            self.mileage=mileage

maruti=Vehicle(120,15)
print(maruti.max_speed,maruti.mileage)
```

```
120 15
```

Create a Vehicle class without any variables and methods

In [22]:
```
class vehicle:
    pass
```

Create a child class Bus that will inherit all of the variables and methods of the Vehicle class

In [1]:
```
class Vehicle:
    def __init__ (self,max_speed,mileage):
        self.max_speed=max_speed
        self.mileage=mileage

class Bus(Vehicle):
    pass

School_bus=Bus(70,10)
print(School_bus.max_speed,School_bus.mileage)
```

```
70 10
```

Create a Bus class that inherits from the Vehicle class. Give the capacity argument of Bus.seating_capacity() a default value of 50.

In [18]:
```
class Vehicle:

    def __init__(self, name, max_speed, mileage):
        self.name = name
        self.max_speed = max_speed
        self.mileage = mileage

    def sitting_capacity(self,capacity):
        return f"sitting capacity of bus {self.name} is {capacity} passanger"
```

In [19]:
```
class Bus(Vehicle):

     def sitting_capacity(self,capacity=50):
        return super().sitting_capacity(capacity=50)
```

In [20]:
```
School_bus=Bus("Volvo",120,23)
```

In [21]:
```
School_bus.sitting_capacity()
```

Out[21]:
```
'sitting capacity of bus Volvo is 50 passanger'
```

Define a property that must have the same value for every class instance (object) Define a class attribute”color” with a default value white. I.e., Every Vehicle should be white.

In [8]:
```
class Vehicle:

    Color="White"

    def __init__(self, name, max_speed, mileage):
        self.name = name
        self.max_speed = max_speed
        self.mileage = mileage

class Bus(Vehicle):
    pass
class Car(Vehicle):
    pass
```

In [10]:
```
School_bus=Bus("Volvo",120,23)

print(School_bus.name,School_bus.Color,School_bus.max_speed,School_bus.mileage
```

```
Volvo White 120 23
```

In [11]:
```
Maruti=Car("Swift",90,41)

print(Maruti.name,Maruti.Color,Maruti.max_speed,Maruti.mileage)
```

```
Swift White 90 41
```

In [12]:
```
class Vehicle:
    def __init__(self, name, mileage, capacity):
        self.name = name
        self.mileage = mileage
        self.capacity = capacity

    def fare(self):
        return self.capacity * 100

class Bus(Vehicle):
    def fare(self):
        amount=super().fare()
        amount+=amount*0.1
        return amount

School_bus = Bus("School Volvo", 12, 50)

print("Total Bus fare is:", School_bus.fare())
```

```
Total Bus fare is: 5500.0
```

In [13]:
```
class Vehicle:
    def __init__(self, name, mileage, capacity):
        self.name = name
        self.mileage = mileage
        self.capacity = capacity

class Bus(Vehicle):
    pass

School_bus = Bus("School Volvo", 12, 50)
```

In [14]:
```
type(School_bus)
```

Out[14]:
```
__main__.Bus
```

In [15]:
```
class Vehicle:
    def __init__(self, name, mileage, capacity):
        self.name = name
        self.mileage = mileage
        self.capacity = capacity

class Bus(Vehicle):
    pass

School_bus = Bus("School Volvo", 12, 50)
```

In [16]:
```
isinstance(School_bus,Vehicle)
```

Out[16]:
```
True
```

In [ ]: