In [ ]:

```
Q1). Write a program to reverse an integer in Python---done
Q2). Write a program in Python to check whether an integer is Armstrong number or not.
Q3). Write a program in Python to check given number is prime or not.
Q4). Write a program in Python to print the Fibonacci series using iterative method.
Q5). Write a program in Python to print the fibonacci series using recursive method.
Q6). Write a program in Python to check whether a number is palindrome or not using it
Q7). Write a program in Python to check whether a number is palindrom or not using rec
Q8). Write a program in Python to find greatest among three integers.
Q9). Write a program in Python to check if a number is binary?
Q10). Write a program in Python to find sum of digits of a number using recursion?
Q11). Write a program in Python to swap two numbers without using third variable?
Q12). Write a program in Python to swap two numbers using third variable?
Q13). Write a program in Python to find prime factors of a given integer.
Q14). Write a program in Python to add two integer without using arithmetic operator?
Q15). Write a program in Python to find given number is perfect or not?
Q16). Python Program to find the Average of numbers with explanations.
Q17). Python Program to calculate factorial using iterative method.
Q18). Python Program to calculate factorial using recursion.
Q19). Python Program to check a given number is even or odd.
Q20). Python program to print first n Prime Number with explanation.
Q21). Python Program to print Prime Number in a given range
Q22). Python Program to find Smallest number among three.
Q23). Python program to calculate the power using the POW method.
Q24). Python Program to calculate the power without using POW function.(using for loop
Q25). Python Program to calculate the power without using POW function.(using while lo
Q26). Python Program to calculate the square of a given number.
Q27). Python Program to calculate the cube of a given number
Q28). Python Program to calculate the square root of a given number.
Q29). Python program to calculate LCM of given two numbers.
Q30). Python Program to find GCD or HCF of two numbers.
Q31). Python Program to find GCD of two numbers using recursion.
Q32). Python Program to Convert Decimal Number into Binary.
Q33). Python Program to convert Decimal number to Octal number.
Q34). Python Program to check the given year is a leap year or not.
Q35). Python Program to convert Celsius to Fahrenheit.
Q36). Python Program to convert Fahrenheit to Celsius.
Q37). Python program to calculate Simple Interest with explanation.

Python Coding Questions on String
Q1). Python program to remove given character from String.
Q2). Python Program to count occurrence of a given characters in string.
Q3). Python Program to check if two Strings are Anagram.
Q4). Python program to check a String is palindrome or not.
Q5). Python program to check given character is vowel or consonant.
Q6). Python program to check given character is digit or not.
Q7). Python program to check given character is digit or not using isdigit() method.
Q8). Python program to replace the string space with a given character.
Q9). Python program to replace the string space with a given character using replace()
Q10). Python program to convert lowercase char to uppercase of string.
Q11). Python program to convert lowercase vowel to uppercase in string.
Q12). Python program to delete vowels in a given string.
Q13). Python program to count Occurrence Of Vowels & Consonants in a String.

Q14). Python program to print the highest frequency character in a String.
Q15). Python program to Replace First Occurrence Of Vowel With '-' in String.
Q16). Python program to count alphabets, digits and special characters.
Q17). Python program to separate characters in a given string.
Q18). Python program to remove blank space from string.
```

```
Q19). Python program to concatenate two strings using join() method.
Q20). Python program to concatenate two strings using join() method.
Q21). Python program to remove repeated character from string.
Q22). Python program to calculate sum of integers in string.
Q23). Python program to print all non repeating character in string.
Q24). Python program to copy one string to another string.
Q23). Python Program to sort characters of string in ascending order.
Q24). Python Program to sort character of string in descending order.

Python Coding Questions on Array
Q1). Write a program in Python for, In array 1-100 numbers are stored, one number is m
Q2). Write a program in Python for, In a array 1-100 multiple numbers are duplicates,
Q3). Write a program in Python for, How to find all pairs in array of integers whose s
Q4). Write a program in Python for, How to compare two array is equal in size or not.
Q5). Write a program in Python to find largest and smallest number in array.
Q6). Write a program in Python to find second highest number in an integer array.
Q7). Write a program in Python to find top two maximum number in array?
Q8). Write a program in Python to remove duplicate elements form array.
Q9). Python program to find top two maximum number in array.
Q10). Python program to print array in reverse Order.
Q11). Python program to reverse an Array in two ways.
Q12). Python Program to calculate length of an array.
Q13). Python program to insert an element at end of an Array.
14). Python program to insert element at a given location in Array.
15). Python Program to delete element at end of Array.
16). Python Program to delete given element from Array.
17). Python Program to delete element from array at given index.
18). Python Program to find sum of array elements.
19). Python Program to print all even numbers in array.
20). Python Program to print all odd numbers in array.
21). Python program to perform left rotation of array elements by two positions.
22). Python program to perform right rotation in array by 2 positions.
23). Python Program to merge two arrays.
24). Python Program to find highest frequency element in array.
25). Python Program to add two number using recursion.
26). Python Program to find sum of digit of number using recursion.


Python Linked List Coding interview questions
Q1). Python Program to Create and Traverse Singly linked list.
Q2). Python program to insert a node in linked list.
Q3). Write a program in Python to reverse a singly linked list.
Q4). Python Program to search an element in singly linked list.
Q5). Linked list Deletion in Python: At beginning, End, Given location
Q6). Write a program in Python to find 3rd element of Linked List from last in single
Q7). Write a program in Python to find middle element of a linked list
```

In [1]:
```
# Q1). Write a program to reverse an integer in Python using while
number=12345
reverse=0
digit=0
while number !=0:
    digit=number % 10
    reverse=reverse*10+digit
    number//=10
print(str(reverse))
```

```
54321
```

In [13]:
```
# Q1). Write a program to reverse an integer in Python using for loop
```

```
number=123454556
digit,rev =0,0
for i in range(len(str(number))):
    digit=number % 10
    rev=rev*10 + digit
    number//=10
print(digit)
print(rev)
print(number)
```

```
1
655454321
0
```

In [20]:
```
# Q2). Write a program in Python to check whether an integer is Armstrong number or no

number=153
total,digit=0,0
temp=number
while temp>0:
    digit=temp % 10
    total+=digit**3
    temp//=10


if number==total:
    print('yes')
else:
    print('No')
```

```
yes
```

In [2]:
```
# Q3). Write a program in Python to check given number is prime or not.

def prime(number):
    for i in range(2,number):
        if number % i ==0:
            return False
        return True
print(prime(17))
```

```
True
```

In [61]:
```
# Q4). Write a program in Python to print the Fibonacci series while loop
n=5
n1,n2=0,1
c=0
count=1
while(count<=n):
    print(c)
    count+=1
    n1=n2
    n2=c
    c=n1+n2
```

```
0
1
1
2
3
```

In [57]:
```
# Q4). Write a program in Python to print the Fibonacci series using for loop

n=5
n1=0
n2=1
print(n1,n2,end=' ')
print(n2,end=' ')
for i in range(2,n+1):
    n3=n1+n2
    print(n3,end=' ')
    n1,n2=n2,n3
```

```
0 1 1 1 2 3 5
```

In [3]:
```
# Q5). Write a program in Python to print the fibonacci series using recursive method.
def fibo(n):
    if n<=1:
        return n
    else:
        return fibo(n-1)+fibo(n-2)
x=6
for i in range(x):
    print(fibo(i))
```

```
0
1
1
2
3
5
```

In [ ]:
```
# Q6). Write a program in Python to check whether a number is palindrome or not using
```

In [9]:
```
Python built-in data types:
1)integer
2)String
3)Float
4)Complex
5)Boolean

Collection:
1)List:
    -append
    -extend
    -pop(index)
    -remove(value)
    -reverse()
    -sort(reverse=True)
    -insert(index)
    -count()
2)Tuple:
    -Dublicate allowed
    -count()
```

```
    -index()
    -ordered


3)Set:
    -unorderd
    -union
    -intersection
    -differece
    -symmetric--uniqe from two set
    -update
    -copy
    -remove()
    -discard()

4)Dictioanry:
    -update
    -dict--constructor
    -get()
    -keys()
    -values()
    -items()
    -fromKeys()
    -setdefault
    -delete
    -clear
    -copy()
    -dict-comprehension
```

In [46]:
```
# l=[]
# l1=[]
# for i in range(3):
#     l1.clear()
#     for j in range(1,4):
#         l1.append(j)
#     l.append(l1)
# print(l)

li=[[i for i in range(1,4)] for j in range(1,4)] # Nested list comprehension
print(li)

# output [[1,2,3],[1,2,3],[1,2,3]]
```

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

In [6]:
```
li=[1,2,3,4,5,6,7,8,9,10]
l=['even' if i%2==0 else 'odd' for i in li ]
print(l)
```

```
['odd', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'even']
```

In [78]:
```
# li=[4,5,7,9,10]
# for i in range(len(li)):
#     print(i,li[i])

# for i,j in enumerate(li):
#     print(i,j)
```

```
# li=[4,5,7,9,10]
# di={i:j for i,j in enumerate(li)}
# print(di)

# Return the name whose marks is 23

dict={'a':24,'b':22,'d':23,'c':26}
di={k:v for k,v in (sorted(dict.items(),key=lambda x:x[0]))}
print(di)

print(sorted(dict.values()))
```

```
{'a': 24, 'b': 22, 'c': 26, 'd': 23}
[22, 23, 24, 26]
```

In [94]:

```
# Q1). Python program to remove given character from String.
# st=' python   20220 '
# a=st.strip()
# print(a)
# b=st.replace('  ','')
# print(b)
# c=st.replace('0','',1)
# print(c)

# st='Welcome python 2022'
# print("-".join(st.split()))

# a='python'
# print(a[::-1])

a='nikh'
b=''
for i in range(len(a)-1,-1,-1):
    b=b+a[i]
print(b)




# Q2). Python Program to count occurrence of a given characters in string.

# Q3). Python Program to check if two Strings are Anagram.
# Q4). Python program to check a String is palindrome or not.
# Q5). Python program to check given character is vowel or consonant.
# Q6). Python program to check given character is digit or not.
# Q7). Python program to check given character is digit or not using isdigit() method.
# Q8). Python program to replace the string space with a given character.
# Q9). Python program to replace the string space with a given character using replace
# Q10). Python program to convert lowercase char to uppercase of string.
# Q11). Python program to convert lowercase vowel to uppercase in string.
# Q12). Python program to delete vowels in a given string.
# Q13). Python program to count Occurrence Of Vowels & Consonants in a String.
```

```
hkin
```

In [95]:

```
a='nikhil'
b=''
for i in range(len(a)-1,-1,-1):
```

```
    b=b+a[i]
print(b)
```

```
hkin
```

In [96]:
```
str1=input("Enter the string :")
str2=""
for i in str1:
    str2=i+str2
print(str2)
```

```
Enter the string :python
nohtyp
```

In [105…
```
st='welcome python 2022'
a=st.split()
b=a[::-1]
# print(' '.join(b))
c=[]
for i in range(len(a)-1,-1,-1):
     c.append(a[i][::-1])
print(*c)
print(type(c))


# output='2022 python welcome'
```

```
2202 nohtyp emoclew
<class 'list'>
```

In [101…
```
s="ankita"
i=len(s)-1
target=""
while i>=0:
    target=target+s[i]
    i=i-1
print(target)
```

```
atikna
```

In [7]:
```
#lambda arg:expression

li=[5,6,7,8]
# def sq(n):
#     return n*n

a=list(map(lambda x:x**x,li))
# print(a)


#map(function,iterable)
#filter(function,iterable)

li=[8,6,7,5,15,30]

b=list(filter(lambda x:x%2!=0  ,li))
# print(b)
c=list(filter(lambda x:x%3==0 or x%5==0,li))
print(c)
#reduce
from functools import reduce
```

```
li=[4,6,8]
d=reduce(lambda x,y:x if x<y else y,li)
b=reduce(lambda x,y:x+y,li)
print(d)
print(b)
```

```
[6, 5, 15, 30]
4
18
```

In [ ]:

In [ ]:
```
lst = [10, 20, 4, 45, 99]
maximum1 = max(lst)
maximum2 = max(lst, key=lambda x: min(lst)-1 if (x == maximum1) else x)
print(maximum2)
```

In [ ]:
```
Git cmd:
git init----normal folder is convert to git repository.
git status---get file state of the repo(stage,untracked, modifield, etc)
git log----history of the commit state cof current repo.
touch filename.extention.------create the file
start filename.extention.------open the file
```

In [147…
```
l=[[i for i in range(1,4)] for j in range(1,4)]
print(l)
```

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

In [ ]:
```
a=input('Enter three number: ').split()
if a[0]>a[1] and a[0]>a[2]:
    print('largest is:',a[0])
elif a[1]>a[0] and a[1]>a[2]:
    print('largest is:',a[1])
else:
    print('largest is:',a[2])
```

In [1]:
```
a=list(map(int,input('Enter number: ').split()))
# print(sum(a))
total=0
for x in a:
    total+=x
print(total)
```

```
Enter number: 1 4 7
12
```

In [5]:
```
st='welcome a to you internet word'
vow=['a','e','i','o','u']
newst=''
for i in st:
    for j in vow:
        if j in i:
            newst=st.replace(j,'-')
# for j in vow:
#     if j in st:
#         newst=st.replace(j,'-')
print(newst)
```

```
welc-me a t- y-u internet w-rd
```

In [8]:
```
z=[i for i in range(1,101) if i%5==0]
print(z)
```

```
[5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100]
```

In [14]:
```
input=[2,4,5,6,8]
out=[pow(i,3) for i in input]
print(out)
```

```
[8, 64, 125, 216, 512]
```

In [16]:
```
input=[2,4,5,6,8]
out=[i*i for i in input]
print(out)
```

```
[4, 16, 25, 36, 64]
```

In [18]:
```
# print factorial given number
fact=1
n=5
while n>0:
    fact=fact*n
    n=n-1
print(fact)
```

```
120
```

In [28]:
```
# print occurrences of each char in string
a='aabbcdd'
di={}
for i in a:
    if i not in di:
        di[i]=1
    else:
        di[i]+=1
for x,y in di.items():
    print(x,'count is:',y)
```

```
a count is: 2
b count is: 2
c count is: 1
d count is: 2
```

In [33]:
```
# remove duplicate char from string
# Method 1
st='vijayvijagtap'
di=set()
for x in st:
    di.add(x)
    n=''.join(di)
print(n)

# Method 2
st='vijayvijagtap'
li=[]
for x in st:
    if x not in li:
        li.append(x)
        new=''.join(li)
print(new)
```

```
vigjatyp
vijaygtp
```

In [42]:
```
# sort the given string a-z
st='djscmad'
out=''.join(st)
print(sorted(out))

# Method  2
from itertools import accumulate
st='djscmad'
out=list(accumulate(sorted(st)))[-1]
print(out)
```

```
['a', 'c', 'd', 'd', 'j', 'm', 's']
acddjms
```

In [47]:
```
#swap key-value pair in dictionary
dict={'ab':1,'cd':2,'ef':3}
out={key:value for value,key in dict.items()}
print(out)
```

```
{1: 'ab', 2: 'cd', 3: 'ef'}
```

In [50]:
```
dict={'ab':1,'cd':2,'ef':3}
out={key:value for value,key in dict}
print(out)
```

```
{'b': 'a', 'd': 'c', 'f': 'e'}
```

In [58]:
```
# sum of values
dict={'ab':1,'cd':2,'ef':3}
a=sum(dict.values())
print(a)
```

```
6
```

In [67]:
```
# program to reverse order of words...
str='welcome python developer'
a=str.split()
l1=[]
print(a[::-1])
```

```
['developer', 'python', 'welcome']
```

In [75]:
```
# Native data type to json format conversion we use dumps()
import json
dict={'vijay':'Kumar',"rakesh":"kumar","vishal":"dube"}
print('Before conversion',dict)
print(type(dict))
json_data=json.dumps(dict)
print('After conversion to json')
print(json_data)
print(type(json_data))
#*********************************************
# json format to native data type conversion we use the loads()
py_data=json.loads(json_data)
print(py_data)
print(type(py_data))
```

```
Before conversion {'vijay': 'Kumar', 'rakesh': 'kumar', 'vishal': 'dube'}
<class 'dict'>
After conversion to json
{"vijay": "Kumar", "rakesh": "kumar", "vishal": "dube"}
<class 'str'>
{'vijay': 'Kumar', 'rakesh': 'kumar', 'vishal': 'dube'}
<class 'dict'>
```

In [80]:
```
#input=[1,2,3,4]
#output={1:1,2:4,3:9,4:16,5:25}
input=[1,2,3,4,5]
out={i:i*i for i in input}
print(out)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

In [91]:
```
a='123,456.87'
a=a.replace(',','temp')
print(a)
a=a.replace('.',',')
print(a)
a=a.replace('temp','.')

print(a)
```

```
123temp456.87
123temp456,87
123.456,87
```

In [1]:
```
# reverse the list without using inbuild function
li=[1,2,3,4,5]
list1=[]
for i in range(len(li),0,-1):
    list1.append(i)
print(list1)
# method 2
li=[1,2,3,4,5]
list1=[]
length=len(li)
while length>0:
    list1.append(li[length-1])
    length=length-1
print(list1)
```

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

In [100…
```
# find the second max value with indexing
list1=[2,9,5,3,6]
list1=sorted(list1)[-2]
print(list1)

#method 2
list1=[2,9,5,3,6]
mx=list1[0]
smx=0
for x in list1:
    if x>mx:
        mx=x
    if x<mx:
```

```
        smx=x
print(smx)
```

```
6
6
```

In [101…

```
def add(li):
    return sum(li)

li=[3,1,2,10,1]
add(li)
```

Out[101]:

```
17
```

In [10]:

```
# list comprehension
li=[3,1,2,10,1,3]
li1=[i for i in li if li.count(i)>1]  # print duplicate from the list
print(li1)

li2=[i for i in li if i%2==0]         # print even number from list
print(li2)

li2=[i for i in li]                   # skip the duplicate
print(set(li2))

li3=[i for i in li]
print(li3)
```

```
[3, 1, 1, 3]
[2, 10]
{10, 1, 2, 3}
[3, 1, 2, 10, 1, 3]
```

In [106…

```
def lst(li):
    for i in range(1, len(li)):
        li[i] += li[i - 1]
    return li

li = [3,1,2,10,1]
print(lst(li))
```

```
[3, 4, 6, 16, 17]
```

In [2]:

```
x = int(input())
y = int(input())
z = int(input())
n = int(input())
output = []
abc = []
for X in range(x+1):
    for Y in range(y+1):
        for Z in range(z+1):
            if X+Y+Z != n:
                abc = [X,Y,Z]
                output.append(abc)
print(output)
```

```
1
1
1
1
[[0, 0, 0], [0, 1, 1], [1, 0, 1], [1, 1, 0], [1, 1, 1]]
```

In [3]:
```
s1={1,5,4,2}
s2={5,4,6,7}
print(s1^s2)
print(s1.intersection(s2))
print(s1.union(s2))
print(s1 & s2)
```

```
{1, 2, 6, 7}
{4, 5}
{1, 2, 4, 5, 6, 7}
{4, 5}
None
```

In [17]:
```
def find_max_sum(li,li2):
    limx=li[0]
    li2mx=li2[0]
    for i in range(len(li)):
        if li[i]>limx:
            limx=li[i]
    print(limx)

    for j in range(len(li2)):
        if li2[j]>li2mx:
            li2mx=li2[j]
    print(li2mx)
    return limx+li2mx



li=[4,9,3]
li2=[7,6,8]
find_max_sum(li,li2)
```

```
9
8
```
Out[17]:
```
17
```

In [28]:
```
st='3x4b2z'   # output-xxxbbbbzz
new=''
for i in range(len(st)):
    if st[i].isdigit():
        print(st[i]*int(st[i+1]),end='')

# print(new)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_4552\1155649559.py in <module>
      3 for i in range(len(st)):
      4     if st[i].isdigit():
----> 5         print(st[i]*int(st[i+1]),end='')
      6
      7 # print(new)

ValueError: invalid literal for int() with base 10: 'x'
```

In [30]:
```
d = {0: 'a', 1: 'b', 2: 'c'}
for x in d.keys():
    print(d[x])
```

```
a
b
c
```

In [44]:
```
x = [i**+1 for i in range(3)]; print(x);
```

```
[0, 1, 2]
```

In [46]:
```
a="abc"
b="xyz"
print([i+j for i in a for j in b])
```

```
['ax', 'ay', 'az', 'bx', 'by', 'bz', 'cx', 'cy', 'cz']
```

In [57]:
```
input=[
    {"dsi":'abc','asv':'uk1'},
    {"dsi":'abc','asv':'uk1'},
    {'dsi':'test','asv':'us1'},
    {'dsi':'test','asv':'us2'}]
output=[]

for i in input:
    if i not in output:
        output.append(i)

for j in output:
    if j['dsi']=='test':
        j['asv']=['us1','us2']
    if j['dsi']=='abc':
        j['asv']=['uk1','uk1']

print(output)
```

```
[{'dsi': 'abc', 'asv': ['uk1', 'uk1']}, {'dsi': 'test', 'asv': ['us1', 'us2']}, {'ds
i': 'test', 'asv': ['us1', 'us2']}]
```

In [73]:
```
data = [{"dsi":'abc','asv':'uk1'},
        {"dsi":'abc','asv':'uk1'},
        {'dsi':'test','asv':'us1'},
        {'dsi':'test','asv':'us2'}]

def fun(data, key):
    di = {}
    for x in data:
        if x[key] not in di:
```

```
            di[x[key]] = []
        di[x[key]].append(x)
    return di

def fun2(data, key, new_key):
    a = fun(data, key)
    li = []
    for k, v in a.items():
        di2 = {}
        di2[key] = k
#         di2[new_key] = v[0][new_key] # this is the line that needs to change
        di2[new_key] = [i[new_key] for i in v]
        li.append(di2)
    return li

output = fun2(data, 'dsi', 'asv')
print(output)
```

```
[{'dsi': 'abc', 'asv': ['uk1', 'uk1']}, {'dsi': 'test', 'asv': ['us1', 'us2']}]
```

In [ ]:
```
data = [{"dsi":'abc','asv':'uk1'},
        {"dsi":'abc','asv':'uk1'},
        {'dsi':'test','asv':'us1'},
        {'dsi':'test','asv':'us2'}]

def fun(data, key):
    di = {}
    for x in data:
        if x[key] not in di:
            di[x[key]] = []
        di[x[key]].append(x)
    return di

def fun2(data, key, new_key):
    a = fun(data, key)
    li = []
    for k, v in a.items():
        di2 = {}
        di2[key] = k
        di2[new_key] = [i[new_key] for i in v]
        li.append(di2)
    return li

output = fun2(data, 'dsi', 'asv')
print(output)
```

In [5]:
```
# Passing function as parameter
def f1(a):
    return 'i am f1 '+a()

def f2():
    return 'i am f2'

f1(f2)
```

Out[5]:
```
'i am f1 i am f2'
```

In [9]:
```
# function returning another function
def f1():
```

```
    def f2():
        return 'i am f2'
    print('i am f1')
    return f2

ob=f1()
print(ob())
```

```
i am f1
i am f2
```

In [10]:
```
# lambda function return multiple value

add_sub=lambda x,y:(x+y,x-y)
a,b=add_sub(5,2)
print(a)
print(b)
```

```
7
3
```

In [15]:
```
# Nested Lambda function
add=lambda x=4:(lambda y:x+y)
a=add()
print(a(11))
```

```
15
```

In [16]:
```
# Passing lambda function to another function
def f1(x):
    print(x(5))

a=f1(lambda x:x)
```

```
5
```

In [17]:
```
# Returning Lambda Function from a Function
def f1():
    y=20
    return (lambda x:x+y)

a=f1()
print(a(10))
```

```
30
```

In [19]:
```
# globals and global keyword
i=0
def f1():
    global i
    i+=1
    print('f1',i)
#     f1()  recursion call max 1000 times
f1()
```

```
f1 1
```

In [20]:
```
# globals
i=10
def f1():
    i=5
    x=globals()['i']
```

```
    print('f1',x+i)
f1()
```

```
f1 15
```

In [ ]:

In [22]:
```
#set recursion limit
# Pycharm dfault recursion limit 1000
#Jupyter default recursion limit is 3000
import sys
print('default recursion limit:',sys.getrecursionlimit())
sys.setrecursionlimit(500)
print('User define limit:',sys.getrecursionlimit())
```

```
default recursion limit: 3000
User define limit: 500
```

In [23]:
```
# DECORATOR- Decorator is function that accept a function as parameter and return it.
# Decorator take the result of the function modifield the result and return it.
# In decorator ,function are taken as the argument into another function then
# call inside wrapper function
#Example:

def decor(fun):
    def inner():
        a=fun()
        add=a+15
        return add
    return inner

def num():
    return 10

result=decor(num)
print(result())
```

```
25
```

In [2]:
```
number=145
rev,digit=0,0
while number !=0:
    digit=number%10
    rev=rev*10+digit
    number=number//10
print(str(rev))

number=145
rev,digit=0,0
while number !=0:
    digit=number%10
    rev=rev*10+digit
    number=number//10
print(str(rev))
```

```
541
541
```

In [4]:
```
number=14545
rev,digit=0,0
```

```
for x in range(len(str(number))):
    digit=number%10
    rev=rev*10+digit
    number=number//10
print(str(rev))
```

```
54541
```

In [5]:
```
# Armstrong number
number=154
total,digit=0,0
temp=number
while temp>0:
    digit=temp % 10
    total+=digit**3
    temp//=10
if number==total:
    print('it armstrong number')
else:
    print('Not armstrong number')
```

```
Not armstrong number
```

In [6]:
```
def is_prime(num):
    for i in range(2,number):
        if number % i==0:
            return False
        return True

print(is_prime(17))
```

```
False
```

In [7]:
```
#fibonacci series
n=6
n1=0
n2=1
print(n1,n2,end=' ')
# print(n2,end=' ')
for i in range(2,n+1):
    n3=n1+n2
    print(n3,end=' ')
    n1,n2=n2,n3
```

```
0 1 1 2 3 5 8
```

In [8]:
```
#fibonacci series function

def fibo_series(n):
    if n<=1:
        return n
    else:
        return fibo_series(n-1) + fibo_series(n-2)
n=6
for i in range(n):
    print(fibo_series(i))
```

```
0
1
1
2
3
5
```

In [9]:
```
from string import digits

stb = 'abc147vf5'
digit = str.maketrans('', '', digits)
print(digit)
output = stb.translate(digit)
print(output)
```

```
{48: None, 49: None, 50: None, 51: None, 52: None, 53: None, 54: None, 55: None, 56:
None, 57: None}
abcvf
```

In [10]:
```
st='abcabcaac'
di={}
for i in st:
    if i not in di:
        di[i]=1
    else:
        di[i]+=1
for key,value in di.items():
    print(key,'count is:',value)
```

```
a count is: 4
b count is: 2
c count is: 3
```

In [11]:
```
st='python'
i=len(st)-1
new=''
while i>=0:
    new=new+st[i]
    i=i-1
print(new)
```

```
nohtyp
```

In [12]:
```
li=[1,2,4]
l2=[]
while bool(li):
    element=li.pop()
    l2.append(element)
print(l2)
```

```
[4, 2, 1]
```

In [13]:
```
def convert(list):
    result_dict = {list[i]: list[i + 1] for i in range(0, len(list), 2)}
    return result_dict


list = [1, 2, 3, 4, 5, 6]
print(convert(list))
```

```
{1: 2, 3: 4, 5: 6}
```

In [ ]: