Python Interview Practice Questions

In [2]:

```
#Format - 1

minimum = int(input("Enter the min : "))
maximum = int(input("Enter the max : "))

for n in range(minimum,maximum + 1):
    if n > 1:
        for i in range(2,n):
            if (n % i) == 0:
                break
        else:
            print(n)
```

```
Enter the min : 5
Enter the max : 10
5
7
```

In [1]:

```
# Format - 2

numr=int(input("Enter range:"))

print("Prime numbers:",end=' ')

for n in range(1,numr):

    for i in range(2,n):

        if(n%i==0):

            break

    else:

        print(n,end=' ')
```

```
Enter range:9
Prime numbers: 1 2 3 5 7
```

In [2]:

```
# Sorting elements in the list
# method - 1


mylist = [5,8,9,6,3]


for i in range(len(mylist)):
    for j in range(i+1,len(mylist)):
        if mylist[j] < mylist[i]:
            mylist[i],mylist[j] = mylist[j],mylist[i]



print(mylist)


# method - 2

mylist = [2,5,7,5,3,4,5,2]
ass = []

while mylist:
    minimum = mylist[0]
    for i in mylist:
        if i < minimum:
            minimum = i
    ass.append(minimum)
    mylist.remove(minimum)

print(ass)
```

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

In [ ]:

```
# Find maximum number from an array

arr = [5,4,3,6,9]

maximum = arr[0]

for i in arr:
    if i>maximum:
        maximum = i

print(maximum)
```

In [7]:

```
# For accending

mylist = [5,8,9,6,3]

new = sorted(mylist)
print(new)
```

```
[3, 5, 6, 8, 9]
```

In [8]:

```
# For decending

mylist = [5,8,9,6,3]

new = sorted(mylist,reverse = True)
print(new)
```

```
[9, 8, 6, 5, 3]
```

In [1]:

```
# find dublicate value and sort in accending order

mylist = [2,5,7,5,3,4,5,2]


def duplicate(mylist):
    empty = []
    ass = []

    for i in mylist:
        if i not in empty:
            empty.append(i)

    while empty:
        minimum = empty[0]
        for i in empty:
            if i < minimum:
                minimum = i
        ass.append(minimum)
        empty.remove(minimum)
    return ass
```

In [2]:

```
duplicate([2,5,7,5,3,4,5,2])
```

Out[2]:

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

In [1]:

```
# sorting list in ascending order

mylist = [5,8,9,6,3]


empty = []

while mylist:
    minimum = mylist[0]
    for i in mylist:
        if i < minimum:
            minimum = i
    empty.append(minimum)
    mylist.remove(minimum)

print(empty)
```

```
[3, 5, 6, 8, 9]
```

In [19]:

```
# Python program to find second largest number
# in a list

# List of numbers
list1 = [10, 20, 20, 4, 45, 45, 45, 99, 99]

# Removing duplicates from the list
list2 = list(set(list1))

# Sorting the  list if possible use above method to sort
list2.sort()

# Printing the second last element
print("Second largest element is:", list2[-2])
```

```
Second largest element is: 45
```

In [2]:

```
# Fabonacci series


def fab(n):
    if n == 0 :
        return 0
    elif n==1:
        return 1
    elif n==2:
        return 1
    else:
        return fab(n-1)+fab(n-2)

for i in range(1):
    print(fab(i), end = ', ')



# Function for nth Fibonacci number
def Fibonacci(n):

    # Check if input is 0 then it will
    # print incorrect input
    if n < 0:
        print("Incorrect input")

    # Check if n is 0
    # then it will return 0
    elif n == 0:
        return 0

    # Check if n is 1,2
    # it will return 1
    elif n == 1 or n == 2:
        return 1

    else:
        return Fibonacci(n-1) + Fibonacci(n-2)

# Driver Program
print(Fibonacci(9))
```

```
0, 34
```

In [20]:

```
# Print list in decending order

mylist = [5,8,9,6,3]

empty = []

while mylist:
    maximum = mylist[0]
    for i in mylist:
        if i>maximum:
            maximum = i
    empty.append(maximum)
    mylist.remove(maximum)


print(empty)
```

```
[9, 8, 6, 5, 3]
```

In [22]:

```
# Reversing the list

mylist = [5,8,9,6,3]
new = mylist[::-1]
print(new)
```

```
[3, 6, 9, 8, 5]
```

In [25]:

```
# Checking string is palindrome or not

mystring = 'malam'

new = mystring[::-1]

if mystring==new:
    print('Its palindrome')
else:
    print('Its not palindrome')
```

```
Its palindrome
```

In [17]:

```
# without indexing

my_string= "mam"

string=""

for i in my_string:
    string = i + string

if my_string == string:
    print('Its palindrom')
else:
    print('Its not palindrom')
```

```
Its palindrom
```

In [3]:

```
# Find the duplicate number in mylist

mylist = [1,1,1,1,2,3,3,6,9,7,6]

print(set([x for x in mylist if mylist.count(x)>1]))


# Using long-method

empty = []

for i in mylist:
    if mylist.count(i)>1:
        empty.append(i)

print(set(empty))
```

```
{1, 3, 6}
{1, 3, 6}
```

In [32]:

```
# Number of words in a given sentense

mystring = 'Maintaining & expending the database of prospects for the organization.'

len(mystring.split())
```

Out[32]:

```
10
```

In [36]:

```
# find a number in a list whether it is present or not

find = int(input('Enter the number you want to find : '))

mylist = [5,8,9,6,3]

if find in mylist:
    print('number found')
else :
    print('number not found')
```

```
Enter the number you want to find : 5
number found
```

In [3]:

```
# extract digit from a given string

mystring = 'Maintaining 12345 & expending 12345 the database of prospects for 12345 the org

digit = ''

for i in mystring:
    if i.isdigit()==True:
        digit = digit + i

print(digit)
```

```
123451234512345
```

In [45]:

```
# convert the string into words

mystring = 'Maintaining 12345 & expending 12345 the database of prospects for 12345 the org

mystring.rstrip().split(' ')
```

Out[45]:

```
['Maintaining',
 '12345',
 '&',
 'expending',
 '12345',
 'the',
 'database',
 'of',
 'prospects',
 'for',
 '12345',
 'the',
 'organization']
```

In [4]:

```
# Check whether a number is palindrome

num = 1221

temp = num


rev = 0

while (num>0):
    rem = num%10

    rev = rev*10+ rem

    # use to add digit in last suppose in 34 we want
    # to make 345 then what we do multiply 34 by 10 then add 5

    num = num //10  # Use to access quotient

print(rev)
print(temp)

if temp==rev:
    print("palindrom")
else :
    print("Not Palindrom")
```

```
1221
1221
palindrom
```

In [63]:

```
# Very important suppose we are saving some number with varaible and that variable
# is assign to some new variable so if you make changes in old variable no change will happ
# in newly assign variable because string and integer are immutable

num = 1221
print(num)

temp = num
print(temp)

print(id(num))
print(id(temp))

num = num+1

print(num)
print(temp)

print(id(num))
print(id(temp))
```

```
1221
1221
2809952307920
2809952307920
1222
1221
2809952307536
2809952307920
```

In [72]:

```
# Factorial of a number

def fact(num):
    if num<0:
        return "number invalid"
    elif num == 1:
        return 1

    elif num == 0:
        return 0
    else:
        return num * fact(num-1)

# find the factorial series

for i in range(5):
    print(fact(i))
```

```
0
1
2
6
24
```

In [71]:

```
fact(3)
```

Out[71]:

```
6
```

In [17]:

```
# Factoorial without recurrance

num = 4
fact = 1

if num < 0:
    print("please enter valid number")
elif num == 0:
    print(0)
elif num ==1 :
    print(1)
else:
    for i in range(2,num+1):
        fact = fact*i
    print(fact)
```

```
24
```

In [7]:

```
# amstrong mumber

num = int(input("Enter a number : "))
temp = num

add = 0

while temp > 0 :
    rem = temp%10
    add = add + rem**3
    temp = temp//10


if num == add:
    print('Its Amstrong Number')
else:
    print('Its Not Amstrong')
```

```
Enter a number : 153
Its Amstrong Number
```

In [1]:

```
# Find the length of array without using len function

arr = [1,5,9,6,3]

counter = 0

for i in arr:
    counter = counter + 1

print(counter)
```

```
5
```

In [3]:

```
# Python program to clear a list
# using clear() method

# Creating list
GEEK = [6, 0, 4, 1]
print('GEEK before clear:', GEEK)

# Clearing list
GEEK.clear()
print('GEEK after clear:', GEEK)
```

```
GEEK before clear: [6, 0, 4, 1]
GEEK after clear: []
```

In [ ]:

```
# Remove the empty list from list

test_list = [5, 6, [], 3, [], [], 9]

# printing original list
print("The original list is : " + str(test_list))

# Remove empty List from List
# using list comprehension
res = [ele for ele in test_list if ele != []]

# printing result
print("List after empty list removal : " + str(res))
```

In [24]:

```
# right angle triangle

num = 5

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

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

In [2]:

```
num = 5

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

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

In [1]:

```
num = 5

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

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

In [11]:

```
num = 5

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

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

In [1]:

```
# Creating star pyramid

rows = int(input("Enter number of rows: "))

k = 0

for i in range(1, rows+1):
    for space in range(1, (rows-i)+1):
        print(end="  ")

    while k!=(2*i-1):
        print("* ", end="")
        k += 1

    k = 0
    print()
```

```
Enter number of rows: 5
        * 
      * * * 
    * * * * * 
  * * * * * * * 
* * * * * * * * * 
```

In [38]:

```
# finding the leap year

num = int(input('Enter a number : '))

if num % 4 == 0:
    print('leap year')
elif num % 400 == 0:
    print('leap year')
elif num % 100 == 0:
    print('Not leap year')
else:
    print('Not leap year')
```

```
Enter a number : 66666
Not leap year
```

In [5]:

```
# prime number

num = int(input('Enter a number : '))
flag = 0

for i in range(2,num):
    if num%i == 0:
        flag = flag + 1

if flag==0:
    print('Its prime number')
else:
    print('Not Prime')
```

```
Enter a number : 7
Its prime number
```

In [6]:

```
# optimized code

num = int(input('Enter a number : '))
flag = 0

for i in range(2,int(num/2)+1):
    if num%i == 0:
        flag = flag + 1

if flag==0:
    print('Its prime number')
else:
    print('Not Prime')
```

```
Enter a number : 7
Its prime number
```

In [53]:

```
# optimized code

num = int(input('Enter a number : '))
flag = 0

for i in range(2,(int(num**0.5)+1)):
    if num%i == 0:
        flag = flag + 1

if flag==0:
    print('Its prime number')
else:
    print('Not Prime')
```

```
Enter a number : 999999937
Its prime number
```

In [54]:

```
# Reversing a list

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

a = mylist[::-1]

print(a)
```

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

In [59]:

```
# Reversing a list

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

for i in range(int(len(mylist)/2)):
    mylist[i],mylist[len(mylist)-i-1] = mylist[len(mylist)-i-1],mylist[i]


print(mylist)
```

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

In [6]:

```
#Print the list of integers from  through  as a string, without spaces.

num = int(input('Enter a number'))

for i in range(1,num+1):
    print(str(i),end='')
```

```
Enter a number5
12345
```

In [4]:

```
# Count of maximum occouring letter and number of times it is repeated

mystring = 'Hello everyone, good afternoon'

mydict = {}

for i in mystring:
    if i in mydict:
        mydict[i] = mydict[i] + 1
    else:
        mydict[i] = 1

print(mydict)


mylist = []
new_dict = {'key':'a','value':0}

for i in mydict:
    if mydict[i] > new_dict['value']:
        new_dict['key'] = i
        new_dict['value'] = mydict[i]

print(new_dict)
```

```
{'H': 1, 'e': 5, 'l': 2, 'o': 6, ' ': 3, 'v': 1, 'r': 2, 'y': 1, 'n': 3,
',': 1, 'g': 1, 'd': 1, 'a': 1, 'f': 1, 't': 1}
{'key': 'o', 'value': 6}
```

In [1]:

```
# maximum occourance of 0's in a string

string = input('Enter a binary number : ')

mylist = []

for i in string.split('1'):
    mylist.append(len(i))

print(max(mylist))
```

```
Enter a binary number : 100010000
4
```

SQL Questions

In [ ]:

```
# Creating database and table

CREATE DATABASE IF NOT EXIST MYSQL;

USE MYSQL;

DESCRIBE MYSQL;

CREATE TABLE IF NOT EXIST MYSQL;

DESCRIBE MYSQL;
```

In [ ]:

```
# COPING THE SAME STRUCTURE OF ANOTHER TABLE

CREATE TABLE IF NOT EXIST NEW_MYSQL AS (SELECT * FROM MYSQL_TABLE WHERE 1=2);
```

In [ ]:

```
# COPING THE SAME TABLE OF ANOTHER TABLE

CREATE TABLE IF NOT EXIST RAJGURU AS (SELECT * FROM GAURAV);
```

In [ ]:

```
# Nth HIGHEST SALARY

SELECT * FROM PAYAL
ORDER BY SALARY DESC
LIMIT N-1,1
```

In [ ]:

```
# WITHOUT USING LIMIT AND TOP

SELECT * FROM EMPLOYEE E1
WHERE N-1 = (SELECT COUNT(DISTINCT(E2.SALARY)) FROM EMPLOYEE E2 WHERE E2.SALARY > E1.SALARY
```

In [ ]:

```
# USING OFFSET

SELECT * FROM EMPLOYEE
ORDER BY SALARY DESC
LIMIT 1 OFFSET N-1;
```

In [ ]:

```
# FIND ALL EMPLOYEE WHO HOLDS MANAGER POSITION

SELECT * FROM EMPLOYEE
WHERE EMP_ID IN (SELECT MANAGER_ID FROM EMPLOYEE)
```

In [ ]:

```
# FIND NAME OF EMPLOYEE WHOS NAME BEGINS WITH A

SELECT * FROM EMPLOYEE WHERE NAME LIKE 'A%'
```

In [ ]:

```
# SQL QUERY TO DISPLAY DATE

# RETURN DATETIME IN YYYY-MM-DD (STRING)

SELECT CURRENT_DATE
SELECT CURRENT_DATE()
SELECT CURDATE()

# RETURN DATETIME IN YYYY-MM-DD-HH-MM-SS (STRING)

SELECT DATE(NOW())
SELECT DATE(CURRENT_TIMESTAMP())
```

In [ ]:

```
# FIND ALTERNATE RECORD IN TABLE

SELECT * FROM EMPLOYEE WHERE ID%2 == 0;

SELECT * FROM EMPLOYEE WHERE ID%2 == 1;
```

In [ ]:

```
# FETCH COMMON RECORD FROM 2 OR MORE TABLE

SELECT * FROM EMPLOYEE
INNER JOIN STUDENT
ON EMPLOYEE.ID = STUDENT.ID
```

In [ ]:

```
# FIND DUBLICATE RECORDS FROM THE TABLE

SELECT NAME,COUNT(NAME),PLACE,COUNT(PLACE) FROM EMPLOYEE
GROUP BY NAME,PLACE
HAVING COUNT(NAME)>1,COUNT(PLACE)>1;
```

In [ ]:

```
# REMOVE DUBLICATE ROWS FROM THE TABLE

DELETE E1 FROM EMPLOYEE E1
INNER JOIN EMPLOYEE E2
ON E1.ID < E2.ID AND E1.NAME = E2.NAME;
```

In [ ]:

```
# SELECTING DUBLICATE RECORDS

SELECT * FROM EMPLOYEE E1
INNER JOIN EMPLOYEE E2
ON E1.ID < E2.ID AND E1.NAME = E2.NAME;
```

In [ ]:

```
# FIND THE NTH RECORD FROM THE TABLE

SELECT * FROM EMPLOYEE
LIMIT N-1,1

SELECT * FROM EMPLOYEE
LIMIT 1 OFFSET N-1
```

In [ ]:

```
# FIND FIRST FIVE AND LAST FIVE RECORDS

SELECT * FROM EMPLOYEE
ORDER BY ID
LIMIT 5


# LAST 5 RECORDS

SELECT * FROM (SELECT * FROM EMPLOYEE ORDER BY ID DESC LIMIT 5)
ORDER BY ID ;

(SELECT * FROM EMPLOYEE ORDER BY ID DESC LIMIT 5)
ORDER BY ID ASC;


SELECT * FROM EMPLOYEE
WHERE ID > (SELECT COUNT(ID) FROM EMPLOYEE)-5;

SELECT * FROM EMPLOYEE
WHERE ID > (SELECT MAX(ID)-5 FROM EMPLOYEE);
```

In [ ]:

```
# fIND FIRST AND LAST RECORD FROM TABLE



SELECT * FROM EMPLOYEE
LIMIT 1

SELECT * FROM EMPLOYEE
WHERE ID = (SELECT MIN(ID) FROM EMPLOYEE)

# LAST RECORD

SELECT * FROM EMPLOYEE
ORDER BY ID DESC
LIMIT 1

SELECT * FROM EMPLOYEE
WHERE ID = (SELECT MAX(ID) FROM EMPLOYEE)
```

In [ ]:

```
# FIND DISTINCT RECORD WITHOUT USING DISTINCT KEYWORD

SELECT DEPARTMENT FROM EMPLOYEE
GROUP BY DEPARTMENT

# USING SET FUNCTION

SELECT DEPARTMENT FROM EMPLOYEE
UNION
SELECT DEPARTMENT FROM EMPLOYEE
```

In [ ]:

```
# fIND MAXIMUM SALARY OF EACH DEPARTMENT AND ARRANGE IN ASCENDING ORDER OF COUNT OF EMPLOYE

SELECT MAX(SALARY),COUNT(ID) FROM EMPLOYEE
GROUP BY DEPARTMENT
ORDER BY COUNT(ID);
```

In [ ]:

```
# HOW TO CHANGE THE DATATYPE OF COLUMN

ALTER TABLE PAYAL MODIFY COLUMN NAME INT;
```

In [ ]:

```
# FIND NUMBER OF MALE AMD FEMALE IN GENDER COLUMN IN SQL

select

sum(case when EmployeeGender='Male' then 1 else 0 end) as Total_Number_Of_Male_Employee,
sum(case when EmployeeGender='Female' then 1 else 0 end) as Total_Number_Of_Female_Employee

from DemoTable;
```

In [ ]:

```
# how to find employee who are also managers

select e.name, m.name from emp e ,emp m
where e.mgr_id = m.emp_id;
```

In [ ]:

```
# Duplicate rows in the table

select * from emp a
where row_id = (select max(row_id) from emp b where a.empno = b.empno)
```

In [ ]:

```
# delete duplicate rows

delete from emp a
where row_id != (select max(row_id) from emp b where a.empno = b.empno)
```

Some more python questions

In [8]:

```
# Fizz Buzz Method

num = int(input('Enter a number : '))

for i in range(1,num+1):
    if i%3 == 0:
        print('Fizz')
    elif i%5 == 0:
        print('Buzz')
    elif i%5 == 0 and i%3 == 0:
        print('FizzBuzz')
    else:
        print(i)
```

```
Enter a number : 3
1
2
Fizz
```

In [13]:

```
def fizz(num):
    dic = {3:'fizz',5:'buzz'}
    for i in range(1,num+1):
        result = ''
        for k,v in dic.items():
            if i%k==0:
                result = result+v
        if not result :
            result = i
        print(result)

fizz(15)
```

```
1
2
fizz
4
buzz
fizz
7
8
fizz
buzz
11
fizz
13
14
fizzbuzz
```

In [7]:

```
# Character occorance

# Least repeating character in string

string = 'aaaasssssssssddddddddf'

lrc = {}

for i in string:
    if i in lrc:
        lrc[i]=lrc[i]+1
    else:
        lrc[i] = 1

print(lrc)   # Print the occorance of all the characters
result = min(lrc,key = lrc.get)
print(result)
```

```
{'a': 4, 's': 9, 'd': 8, 'f': 1}
a
```

In [11]:

```
# Display Count of any perticular element in a string

string = 'aaaasssssssssddddddddf'

search = 'a'

lrc = {}



for i in string:
    if i in lrc:
        lrc[i]=lrc[i]+1
    else:
        lrc[i] = 1

print(lrc)

print(lrc[search])
```

```
{'a': 4, 's': 9, 'd': 8, 'f': 1}
4
```

In [13]:

```
# You are given a string "S". Suppose a character "c"' occurs consecutively "X"times
# in the string. Replace these consecutive occurrences of the character ''c" with (X,c) in
# S should be a number

from itertools import groupby


for k, c in groupby(input("Enter a Number : ")):
    print((len(list(c)), int(k)), end=' ')
```

```
Enter a Number : 11223
(2, 1) (2, 2) (1, 3)
```

In [1]:

```
#Write a python code to print the follwing pattern.

for i in range(7,0,-1):
    for j in range(i,0,-1):
        print(i,end=' ')
    print()
```

```
7 7 7 7 7 7 7
6 6 6 6 6 6
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

In [2]:

```
rows = int(input("Enter number of rows: "))

for i in range(rows, 0, -1):
    for j in range(1, i+1):
        print(j, end=" ")

    print()
```

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

In [6]:

```
# Reversing a number

num = 1234
temp = 1234

rev = 0

while temp:
    rem = temp%10
    rev = rev*10 + rem
    temp = temp//10

print('The reverse of number is ' ,rev)
```

```
The reverse of number is  4321
```

In [10]:

```
# Remove a character from a string

name = 'payal'

name.replace('l','')
```

Out[10]:

```
'paya'
```

In [11]:

```
# Anagram words containing same letters and characters

def anagramCheck(str1, str2):
    if (sorted(str1) == sorted(str2)) :
        return True
    else :
        return False

str1 = input("Please enter String 1 : ")
str2 = input("Please enter String 2 : ")

if anagramCheck(str1,str2):
    print("Anagram")
else:
    print("Not an anagram")
```

```
Please enter String 1 : payal
Please enter String 2 : ayalp
Anagram
```

In [15]:

```
# Sort characters of string in sorted order

mystring = input('Enter a string : ')

ass_new = ''.join(sorted(mystring))

desc_new = ''.join(sorted(mystring ,reverse = True))


print(ass_new)
print(desc_new)
```

```
Enter a string : payal
aalpy
yplaa
```

In [18]:

```
# Find missing number in an array

array = list(range(0,101))
check = list(range(0,101))

missing = []

for i in array:
    if i not in check:
        print(i)
```

In [19]:

```
# Occorance of word in list

weekdays = ['sun','mon','tue','wed','thu','fri','sun','mon','mon']

print(weekdays.count('mon'))
```

```
3
```

In [22]:

```
# Python Program to Count the Number of Vowels in a String

string= input("Enter string:")

vowels=0

for i in string:
    if(i=='a' or i=='e' or i=='i' or i=='o' or i=='u' or i=='A' or i=='E' or i=='I' or i=='
        vowels=vowels+1

print("Number of vowels are:")
print(vowels)
```

```
Enter string:payal
Number of vowels are:
2
```

In [26]:

```
# Count of vowels and consonants

# taking input from the user

string = input('Enter a String : ')

vowels = 0

consonants = 0

for i in string:
    if i in ('a','e','i','o','u','A','E','I','O','U'):
        vowels+=1
    elif i.isalpha():
        consonants+=1

print('Vowels :',vowels,'Consonants:',consonants)
```

```
Enter a String : payal
Vowels : 2 Consonants: 3
```

In [ ]: