

## Nested List

```
In [ ]: Nested List --> List inside list is a concept of nested list.
```

```
Example:
x=[10,20,30,40,50,[98,78,56,45,67]]
```

## How to access elements of nested list?

```
In [4]: x=[10,20,30,40,50,[98,78,56,45,67]]
```

```
In [6]: x[5][2]
```

```
Out[6]: 56
```

```
In [8]: x=[10,20,30,"hello world"]
x[3][6]
```

```
Out[8]: 'w'
```

```
In [14]: x=[10,20,30,40,[10,20,30,["Python","Java"]]]
x[4][3][6]
```

```
Out[14]: 'Python'
```

```
In [30]: x=[10,20,30,40,[10,20,30,["Python","Java"]]]
x[-1][-1][-1]
```

```
Out[30]: 'Java'
```

## how to add element in nested list

```
In [22]: x=[10,20,30,40,[10,20,30,["Python","Java"]]]
x[-1][-1].append('c++')
```

```
Out[22]: [10, 20, 30, 40, [10, 20, 30, ['Python', 'Java', 'c++']]]
```

```
In [23]: x=[10,20,30,40,[10,20,30,["Python","Java"]]]
x[-1][-1].insert(1,"c++")
```

```
Out[23]: [10, 20, 30, 40, [10, 20, 30, ['Python', 'c++', 'Java']]]
```

```
In [24]: x=[10,20,30,40,[10,20,30,["Python","Java"]]]
x[-1][-1].extend([200,300,400])
```

```
Out[24]: [10, 20, 30, 40, [10, 20, 30, ['Python', 'Java', 200, 300, 400]]]
```

```
In [26]: x=[10,20,30,40,[10,20,30,["Python","Java"]]]
x[4].insert(1,100)
```

```
Out[26]: [10, 20, 30, 40, [10, 100, 20, 30, ['Python', 'Java']]]
```

```
In [34]: x=[10,20,30,40,[10,20,30,["Python","Java"]]]
x[-1].insert(-2,'c++')
```

```
Out[34]: [10, 20, 30, 40, [10, 20, 'c++', 30, ['Python', 'Java']]]
```

## Deletion in nested list

```
In [36]: x=[10,20,30,40,[10,20,30,["Python","Java"]]]
x[4].pop(2)
```

```
Out[36]: [10, 20, 30, 40, [10, 20, ['Python', 'Java']]]
```

```
In [38]: x=[10,20,30,40,[10,20,30,["Python","Java"]]]
x[4].pop()
```

```
Out[38]: [10, 20, 30, 40, [10, 20, 30]]
```

```
In [44]: x=[10,20,30,40,[10,20,30,["Python","Java"]]]
for i in x:
```

```
    print(i)

10
20
30
40
[10, 20, 30, ['Python', 'Java']]
```

## Matrix Representation of list

```
In [43]: x=[[1,2,3],
            [4,5,6],
            [7,8,9],
            [10,11,12]]
for i in range(0,len(x)):
```

```
    print(x[i])
print(x[2][1])
print(x[-2][-2])
```

```
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]
[10, 11, 12]
8
8
```

## Nested Dictionary

```
In [ ]: Nested Dictionary means dictionary inside dictionary
```

Syntax:

```
d = {1:{"name":"Pratyush","Class":"DS291122A","Module":"Python"},
     2:{"name":"Akshay","Class":"DS291122A","Module":"Python"}}
```

## How to access element in nested dictionary

```
In [50]: d={1:{"name":"Pratyush","Class":"DS291122A","Module":"Python"},
           2:{"name":"Akshay","Class":"DS291122A","Module":"Python"}}
d[1]["Class"]
```

```
Out[50]: 'DS291122A'
```

## How to add entry in nested Dictionary

```
In [59]: d = {1:{"name":"Pratyush","Class":"DS291122A","Module":"Python"},
              2:{"name":"Akshay","Class":"DS291122A","Module":"Python"}}
d[3]["Qualification"]="B.Tech"
```

```
print(d)
d[2].popitem()
{1: {'name': 'Pratyush', 'Class': 'DS291122A', 'Module': 'Python', 'Qualification': 'B.Tech'}, 2: {'name': 'Akshay', 'Class': 'DS291122A'}}
```

## Python program to Check weather a Given String is Palindrome or not

```
In [ ]: Palindrome String or Not:
ABCCBCBA --> ORIGINAL STRING
ABCCBCBA --> REVERSE STRING
ORIGINAL == REVERSE STRING
PALINDROME STRING
MADAM #PEEP #NALYALAH #NOON
```

### Solution

```
In [62]: #Approch 1
string_name = input("Enter a String :")
reverse_string = string_name[::-1]
if string_name==reverse_string:
    print("String is palindrome")
else:
    print("String is not palindrome")
```

Enter a String :hello  
String is not palindrome

```
In [69]: #Approch 2
string_name = input("Enter a String :")
reverse_name=""
for i in range(-1,-len(string_name)-1,-1):
    reverse_name=reverse_name+string_name[i]
    print(reverse_name)
```

```
if string_name==reverse_name:
    print("It is a palindrome string")
else:
    print("It is not a palindrome string")
```

Enter a String :MAM  
M  
MA  
MA  
It is a palindrome string

## Python Program to Check weather a given string is Anangram or Not

```
In [ ]: #Anagram or Not exmple--> Listen --> siLent , RACES-->CAREs
```

Listen --> siLent

sorted() --> sorted(siLent)

All the letters of the given words are present **in** another word then it **is** an angram string **else** **not**.

Logic:

Sort both the Strings:

LISTEN --> EILNST

SILENT --> EILNST

### Solution

```
In [71]: string_name = input("Enter a String ")
string2_name=input("Enter second string ")
if sorted(string_name)==sorted(string2_name):
    print("It is a anagram string")
else:
    print("It is not an anagram string")
```

Enter a String HELLO  
Enter second string Hi  
It is not an anagram string

## Write a Python program which iterates the integers from 1 to 50.

## For multiples of three print "Fizz" instead of the number and for the multiples of five

print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

### Solution

```
In [72]: for i in range(1,50):
        if i%3==0:
            print("Fizz")
        elif i%5==0:
            print("Buzz")
        elif i%3==0 and i%5==0:
            print("FizzBuzz")
```

Fizz  
Buzz  
Fizz  
Fizz  
Buzz  
Fizz  
Fizz  
Fizz  
Buzz  
Fizz  
Fizz  
Buzz  
Fizz  
Fizz  
Buzz  
Fizz  
Fizz  
Buzz  
Fizz  
Fizz

## Write a Python program which will return all the different unique vowels that are present in the given string.

## Test Case: 1.String may be in upper case or lowr case

## 2.Answer must give unqie vowels list

## 3.If any string is having n repeating characters then that character will be considered only once.

### Solution

```
In [79]: string_name=input("Enter a String : ")
set1 = set(string_name)
```

```
set2 = {"a","e","i","u","o"}
set3=set1.intersection(set2)
list(set3)
```

Enter a String : Mayank  
Out[79]: ['a']

## Program to find the ASCII value of the given character

### Solution

```
In [81]: string_name=input("Enter a String :") #Hello world
for i in string_name:
    print(ord(i),end=" ")
```

Enter a String :hello world  
104 101 108 108 111 32 119 111 114 108 100

## Python Program to find the factors of the number

```
In [ ]: 10 --> 1,2,5,10
These are the numbers that are divisble by 10
```

```
12 --> 1,2,3,4,6,12
```

```
24 --> 1,2,3,4,6,8,12,24
```

```
for i in range(1,11): # 1 2 3 4 5 6 7 8 9 10
    if 10%i==0:
        print(i)
```

### Solution

```
In [83]: num = int(input("Enter a Number :"))
for i in range(1,num+1):
    if num%i==0:
        print(i)
```

Enter a Number :10  
1  
2  
5  
10

## Python Program to Check weather a given Number is Prime or Not

```
In [ ]: #Prime Numbers -->
The number which is divisible by 1 and itself.
```

**if** number **is** 1 --> it **is** neither prime nor composite

```
2 --> 1,2
```

```
7 --> 1,7
```

```
5 --> 1,5
```

```
3 --> 1,3
```

```
91 --> 13,7 1 4 16 1
```

```
for i in range(2,num+1): # 2 3 4 5 6
    if num%i==0:
        print("Number is not prime")
        break
```

```
else:
    print("Number is prime")
```

### Solution

```
In [90]: num=int(input("Enter a Number : "))
if num==1:
```

```
    print("1 is neither prime nor composite")
elif num==1:
```

```
    for i in range(2,num): # 3 0 1 2 3
        if num%i==0:
            print("It is not a prime number")
            break
```

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

Enter a Number : 91  
It is not a prime number

## Python Program to Print Prime Number Between a Given Range

### Solution

```
In [1]: start = int(input("Enter start Value : "))
end = int(input("Enter end Value :"))
```

```
for i in range(start,end):
    if i==0 or i==1:
        continue
```

```
    else:
        for j in range(2,i):
            if i%j==0:
                break
```

```
    else:
        print(i)
```

Enter start Value : 10  
Enter end Value :20  
11  
13  
17  
19

## Pythn Program to Check weather a Given Number is Perfect or NoT

```
In [ ]: Perfect Number means --> If the summation of the factors of a number is equal to the original number then it is a perfect number.
```

Exangle:  
6 **is** a perfect number because factors of 6 are 1,2,3 **and** summation of 1,2,3 **is** also equal to 6.

### Solution

```
In [3]: numbers=int(input("Enter a Number"))
sum=0
```

```
for i in range(1,number): #1,2,3,4,5
```

```
if number%i==0: #sum=0
```

```
if sum==number: #sum=0+1--> 1
```

```
else:
    print("It is not a perfect number")
```

```
else:
    print("It is not a perfect number")
```

Enter a Number6  
It is a perfect number

## Remove duplicates from the given string

### Solution

```
In [4]: string_name=input("Enter a String :")
set1 = set(string_name)
```

```
list1=list(set1)
print("".join(list1))
```

Enter a String :hello  
ehl

## Python program to interchange first and last elements in a list

```
In [ ]: Input : [12, 35, 9, 56, 24]
Output : [24, 35, 9, 56, 12]
```

```
Input : [1, 2, 3]
Output : [3, 2, 1]
```

### Solution

```
In [103]: input = [12, 35, 9, 56, 24,56]
input[len(input)-1],input[0] = input[0],input[len(input)-1]
```

```
Out[103]: [56, 35, 9, 56, 24, 12]
```

## Python program to swap two elements in a list

```
In [ ]: input = [1,2,3,4,5] pos1=1 pos=3
output = [1,4,3,2,5]
```

### Solution

```
In [99]: input = [1,2,3,4,5]
pos1=1
```

```
pos3=3
input[pos1],input[pos3] = input[pos3],input[pos1]
```

```
input
```

```
Out[99]: [1, 4, 3, 2, 5]
```

## Python program to find second smallest number in a list

```
In [ ]: Input : list1 = [10, 20, 4]
Output : 10
```

```
Input : list2 = [20, 10, 20, 4, 100]
Output : 10
```

### Solution

```
In [95]: #Approch-1
list1 = [10, 20, 4]
list1.sort()
```

```
list1[1]
```

```
Out[95]: 10
```

```
In [96]: #Approch-2
list1 = [10, 20, 4]
minimum = min(list1)
```

```
list1.remove(minimum)
min(list1)
```

```
Out[96]: 10
```

## Check if two lists have at-least one element common or not

```
In [ ]: Input : a = [1, 2, 3, 4, 5]
Output : True
```

```
Input : a=[1, 2, 3, 4, 5]
Output : False
```

### Solution

```
In [94]: a = [1, 2, 3, 4, 5]
set1=set(a)
```

```
b=[5, 6, 7, 8, 9]
set2=set(b)
```

```
set3 = set1.intersection(set2)
if len(set3)>=1:
```

```
    print("True")
else:
```

```
    print("False")
```

```
False
```