```
>>> list1=[10,20,30,40,50]
>>> min(list1)
10
>>> max(list1)
50
>>> str1="PYTHON"
>>> min(str1)
'H'
>>> max(str1)
'Y'
>>> list2=[10,20,30,10,20,30,40,50,10,20,30]
>>> list2.index(10)
0
>>> list2.index(10,2,7)
3
>>> list2.insert(3,90)
>>> list2
[10, 20, 30, 90, 10, 20, 30, 40, 50, 10, 20, 30]
>>>
```

Example

```
# How to read or input more than one value in single line
a=input()
print(a)
list1=a.split()
print(list1)
list1=list(map(int,list1))
print(list1)
```

https://www.hackerrank.com/challenges/find-second-maximum-number-in-a-list/problem?isFullScreen=false

```
n=int(input())
list1=list(map(int,input().split()))
list1.sort()
c=list1.count(list1[-1])
print(list1[-(c+1)])
```

https://www.hackerrank.com/challenges/python-lists/problem?isFullScreen=false

```
list1=[]
n=int(input())
for i in range(n):
    cmd=input().split()
    if cmd[0]=="insert":
        list1.insert(int(cmd[1]),int(cmd[2]))
    elif cmd[0]=="append":
        list1.append(int(cmd[1]))
    elif cmd[0]=="remove":
        list1.remove(int(cmd[1]))
    elif cmd[0]=="print":
        print(list1)
    elif cmd[0]=="pop":
        list1.pop()
    elif cmd[0]=="reverse":
        list1.reverse()
    elif cmd[0]=="sort":
        list1.sort()
```

Nested List

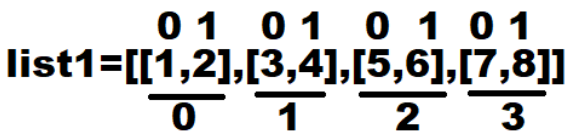
List inside list is called nested list

Defining list as an element inside list is called nested list.

Syntax: list-name=[[e1,e2,e3],[e4,e5,e6],[e7,e8,e9]]

In nested list data is organized logically by dividing into rows and columns

Nested list can be used to represent matrix.



print(list1[-1][-1]) --> 8
print(list1[-2][0]) --> 5

print(list1[0]) [1,2]
print(list1[1]) [3,4]
print(list1[2]) [5,6]
print(list1[3]) [7,8]
print(list1[0][0]) 1
print(list1[0][1]) 2

print(list1[1][0]) --> 3
print(list1[1][1]) --> 4
print(list1[2][0]) --> 5
print(list1[2][1]) --> 6
print(list1[3][0]) --> 7
print(list1[3][1]) --> 8

Example:

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

# How to read using for loop with index
for i in range(3): # 0 1 2
    for j in range(3): # 0 1 2
        print(matrix[i][j],end=' ')
    print()

# How to reading using for loop without index
for a in matrix:
    for b in a:
        print(b,end=' ')
    print()
```

Output

```
1 2 3
4 5 6
7 8 9
1 2 3
4 5 6
7 8 9
```

Example adding two matrices

```
A=[[1,2],[3,4]]
B=[[4,5],[6,7]]
C=[[0,0],[0,0]]

for i in range(2):
    for j in range(2):
        C[i][j]=A[i][j]+B[i][j]

print(A,B,C,sep="\n")
```

Output

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

Example:

```
# Write a program to read 3x3 matrix and display

matrix=[]

for i in range(3):
    row=[]
    for i in range(3):
        value=int(input("Enter Value "))
        row.append(value)
    matrix.append(row)

print(matrix)
```

Output

```
Enter Value 1
Enter Value 2
Enter Value 3
Enter Value 4
Enter Value 5
Enter Value 6
```

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

Example:

```
# Write a program to read 3 students 3 subjects and calculate
total,avg

marks=[]
for i in range(3):
    stud=[]
    for j in range(3):
        m=int(input("Enter Marks "))
        stud.append(m)
    marks.append(stud)

for stud in marks:
    total=sum(stud)
    avg=total/3
    result="pass" if stud[0]>=40 and stud[1]>=40 and stud[2]>=40 else
"fail"
    print(f'{stud}\t{total}\t{avg:.2f}\t{result}')
```

Output

```
Enter Marks 50
Enter Marks 60
Enter Marks 70
Enter Marks 89
Enter Marks 78
Enter Marks 87
Enter Marks 30
Enter Marks 70
Enter Marks 99
[50, 60, 70]180    60.00pass
[89, 78, 87]254    84.67pass
[30, 70, 99]199    66.33fail
```

```
>>> list1=["ABC","XYZ","PQR"]
>>> list1[0]
'ABC'
>>> list1[1]
'XYZ'
>>> list1[2]
'PQR'
>>> list1[1][0]
'X'
>>> list1[0][0]
'A'
>>> list1[-1][-1]
'R'
>>> a,b,c="RRR"
>>> print(a,b,c)
R R R
>>> a,b,c,d="JAVA"
>>> print(a,b,c,d,sep="\n")
J
A
V
A
>>> a,*b="JAVA"
>>> print(a,b,sep="\n")
J
['A', 'V', 'A']
>>> a="ABC"
>>> b="XYZ"
>>> c=[a,b]
>>> print(c)
['ABC', 'XYZ']
>>> d=[*a,*b]
>>> print(d)
['A', 'B', 'C', 'X', 'Y', 'Z']
>>> e=[*a,b]
>>> print(e)
```

['A', 'B', 'C', 'XYZ']

List Comprehension

Displays

For constructing a list, a set or a dictionary Python provides special syntax called "displays", each of them in two flavors:

1. either the container contents are listed explicitly

A=[10,20,30,40,50]
B=[1+2,3+4,5+6]
C=[input(),input(),input()]

2. they are computed via a set of looping and filtering instructions, called a *comprehension*.

List comprehension allows to create new list using for loop and if condition.

Syntax1: [expression for variable in iterable]
Syntax2: [expression for variable in iterable if test]