lists

- combination of elements that are belongs to same kind or heterogenious
- list is related to array

In [1]:
```
1 data1=[2,"hanuman",2.5]
```

In [2]:
```
1 data1
```

Out[2]:
```
[2, 'hanuman', 2.5]
```

In [3]:
```
1 data2=[3,"ramu",3.5]
```

In [4]:
```
1 data2
```

Out[4]:
```
[3, 'ramu', 3.5]
```

In [5]:
```
1 print(data1+data2)
```

```
[2, 'hanuman', 2.5, 3, 'ramu', 3.5]
```

In [9]:
```
1 data2[2]="nvz"
```

In [11]:
```
1 data2
```

Out[11]:
```
[3, 'ramu', 'nvz']
```

In [12]:
```
1 dir(list)
```

Out[12]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__iadd__',
 '__imul__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__rmul__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']
```

In [17]:
```
1 data3=[10,20,30,40,50]
```

In [18]:
```
1 data3
```

Out[18]:
```
[10, 20, 30, 40, 50]
```

In [19]:
```
1 data3.append(10)
```

In [20]:
```
1 data3
```

Out[20]: [10, 20, 30, 40, 50, 10]

In [21]:
```
1 data3.extend([12,13])
```

In [22]:
```
1 data3
```

Out[22]: [10, 20, 30, 40, 50, 10, 12, 13]

In [23]:
```
1 data3
```

Out[23]: [10, 20, 30, 40, 50, 10, 12, 13]

In [24]:
```
1 data3.index(30)
```

Out[24]: 2

In [25]:
```
1 data3.index(10)
```

Out[25]: 0

In [29]:
```
1 data3.count(30)
```

Out[29]: 1

In [27]:
```
1 data3
```

Out[27]: [10, 20, 30, 40, 50, 10, 12, 13]

In [40]:
```
1 data3.insert(0,"vijay")
```

In [41]:
```
1 data3
```

Out[41]: ['vijay', 10, 20, 30, 100, 'vijay', 'ravi', 'ravi', 100, 40, 50, 10, 12, 13]

In [48]:
```
1 data4=[10,30,40,50,90,47]
```

In [49]:
```
1 data4
```

Out[49]: [10, 30, 40, 50, 90, 47]

In [50]:
```
1 data4.reverse()
```

In [52]:
```
1 data4
```

Out[52]: [47, 90, 50, 40, 30, 10]

In [56]:
```
1 data4.sort(reverse=True)
```

In [57]:
```
1 data4
```

Out[57]: [90, 50, 47, 40, 30, 10]

In [55]:
```
1 print(len(data4))
```

```
6
```

In [58]:
```
1 data4
```

Out[58]: [90, 50, 47, 40, 30, 10]

In [59]:
```
1 print(min(data4))
```

```
10
```

In [60]:
```
1 print(max(data4))
```

```
90
```

In [80]:
```
1 s=input().split()
2 # s=s.split('@')
3 print(type(s))
4 print(s)
```

```
1 2 23 234 23 4234 213 123
<class 'list'>
['1', '2', '23', '234', '23', '4234', '213', '123']
```

In [74]:
```
1 s.split(' ')
```

Out[74]: ['10,20,30,40,50']

In [66]:
```
1 s
```

Out[66]: '10,20,30,40,50'

In [91]:
```
# input: 5
#        1 3 4 5 6
#        4 5 6 2 4
# Output:
#        5 8 10 7 10


n = int(input())
f = input().split()
s = input().split()
fr = []
sc = []
for i in range(len(f)):
    fr.append(int(f[i]))
for i in range(len(s)):
    sc.append(int(s[i]))
for i in range(len(fr)):
    print(fr[i]+sc[i],end=" ")
```

```
3
1 3 4 5 6 7 8 9
1 3 40 0 0 0 0
2 6 44 5 6 7 8

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-91-7daf624f67a0> in <module>
     16     sc.append(int(s[i]))
     17 for i in range(len(fr)):
---> 18     print(fr[i]+sc[i],end=" ")

IndexError: list index out of range
```

In [96]:
```
ke =[2,4,54,5,34,5,23,4,2,34]
print(type(ke))
print(ke)
```

```
<class 'list'>
[2, 4, 54, 5, 34, 5, 23, 4, 2, 34]
```

In [97]:
```
print(dir(list))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir_
_', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem
__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass
__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__',
'__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setat
tr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'c
lear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'revers
e', 'sort']
```

In [104]:
```
1 k = [2,23,234234,23,42,34,23423,5235]
```

In [105]:
```
1 k
```

Out[105]: [2, 23, 234234, 23, 42, 34, 23423, 5235]

In [106]:
```
1 print(k.pop())
```

```
5235
```

In [107]:
```
1 k
```

Out[107]: [2, 23, 234234, 23, 42, 34, 23423]

In [109]:
```
1 k.remove(23)
```

In [110]:
```
1 k
```

Out[110]: [2, 234234, 23, 42, 34, 23423]

In [112]:
```
1 k[::-1]
```

Out[112]: [23423, 34, 42, 23, 234234, 2]

In [113]:
```
1 k[0]
```

Out[113]: 2

In [114]:
```
1 k[4:6]
```

Out[114]: [34, 23423]

In [115]:
```
1 k.clear()
```

In [116]:
```
1 k
```

Out[116]: []

In [117]:
```
1 k
```

Out[117]: []

In [119]:
```
1 del k
```

Tuples:

```
- IMMutable (Doesn't Changes) => Static Values
- Indexing (slicing can be performed)
- tuple() or ()
```

In [121]:
```
1 print(dir(tuple))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewa
rgs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__l
e__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__red
uce_ex__', '__repr__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__s
ubclasshook__', 'count', 'index']
```

In [4]:
```
1 s =(1,123,'sdasdasdad',943857834,45.0)
```

In [5]:
```
1 s
```

Out[5]: (1, 123, 'sdasdasdad', 943857834, 45.0)

In [12]:
```
1 ki=list(s)
```

In [13]:
```
1 ki.remove(1)
```

In [14]:
```
1 print(ki)
```

```
[123, 'sdasdasdad', 943857834, 45.0]
```

In [15]:
```
1 s = tuple(ki)
2 s
```

Out[15]: (123, 'sdasdasdad', 943857834, 45.0)

In [16]:
```
1 s.index(123)
```

Out[16]: 0

In [20]:
```
1 s[::-1]
```

Out[20]: (45.0, 943857834, 'sdasdasdad', 123)

Sets:

- Denotes with { , } .
- Un-ordered Data structure
  - Doesn't have index
- Mutable (Ability to manipulate)
- Doesn't allow duplicate values.

In [21]:
```
s={}
type(s)
```

Out[21]: dict

In [22]:
```
s={1:2,2:3}
type(s)
```

Out[22]: dict

In [24]:
```
s={1,2,3,4,"Hello",2.4}
type(s)
```

Out[24]: set

In [25]:
```
print(dir(s))
```

```
['__and__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__
iand__', '__init__', '__init_subclass__', '__ior__', '__isub__', '__iter__', '_
_ixor__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__or__', '__rand
__', '__reduce__', '__reduce_ex__', '__repr__', '__ror__', '__rsub__', '__rxor_
_', '__setattr__', '__sizeof__', '__str__', '__sub__', '__subclasshook__', '__x
or__', 'add', 'clear', 'copy', 'difference', 'difference_update', 'discard', 'i
ntersection', 'intersection_update', 'isdisjoint', 'issubset', 'issuperset', 'p
op', 'remove', 'symmetric_difference', 'symmetric_difference_update', 'union',
'update']
```

In [29]:
```
s1={1,2,3,4,7,9,0,54,11,21}
s1.add(5)
s1
```

Out[29]: {0, 1, 2, 3, 4, 5, 7, 9, 11, 21, 54}

In [30]:
```
l1=[1,2,3,4,54,11,12,"Anu","LOL","Hello",7.6]
l1
```

Out[30]: [1, 2, 3, 4, 54, 11, 12, 'Anu', 'LOL', 'Hello', 7.6]

In [34]:
```
l1=set(l1)
l1
```

Out[34]: {1, 11, 12, 2, 3, 4, 54, 7.6, 'Anu', 'Hello', 'LOL'}

In [35]:
```
l1
```

Out[35]: {1, 11, 12, 2, 3, 4, 54, 7.6, 'Anu', 'Hello', 'LOL'}

In [36]:
```
l2=l1.copy()
```

In [37]:
```
l2
```

Out[37]: {1, 11, 12, 2, 3, 4, 54, 7.6, 'Anu', 'Hello', 'LOL'}

In [38]:
```
l3=l2
l3
```

Out[38]: {1, 11, 12, 2, 3, 4, 54, 7.6, 'Anu', 'Hello', 'LOL'}

In [45]:
```
s1={1,2,3,4,5,7}
s2={6,7,8}
s2.difference(s1)
s1
```

Out[45]: {1, 2, 3, 4, 5, 7}

In [46]:
```
s2.difference_update(s1)
s1
```

Out[46]: {1, 2, 3, 4, 5, 7}

In [44]:
```
s2
```

Out[44]: {6, 8}

In [47]:
```
s1
```

Out[47]: {1, 2, 3, 4, 5, 7}

In [54]:
```
s1.discard(3)
```

In [49]:
```
s1
```

Out[49]: {1, 2, 4, 5, 7}

In [53]:
```
s1.remove(4)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-53-c000b6ac3c0c> in <module>
----> 1 s1.remove(4)

KeyError: 4
```

In [52]:
```
s1
```

Out[52]: {1, 2, 5, 7}

In [56]:
```
s1={1,2,3,4,5}
s2={2,3,4,1,2}
s1.intersection_update(s2)
```

In [57]:
```
s1
```

Out[57]: {1, 2, 3, 4}

In [58]:
```
s2
```

Out[58]: {1, 2, 3, 4}

In [61]:
```
s1={1,2,3,4}
s2={5}
s1.isdisjoint(s2)
```

Out[61]: True

In [62]:
```
s1
```

Out[62]: {1, 2, 3, 4}

In [66]:
```
s2={1,2,3,4,5}
s2.issubset(s1)
```

Out[66]: False

In [67]:
```
s1
```

Out[67]: {1, 2, 3, 4}

In [68]:
```
s2
```

Out[68]: {1, 2, 3, 4, 5}

In [70]:
```
s1.issuperset(s2)
```

Out[70]: False

In [71]:
```
s1
```

Out[71]: {1, 2, 3, 4}

In [72]:
```
s1.pop()
```

Out[72]: 1

In [76]:
```
1 s1.add(5)
2 s1
```

Out[76]: {2, 3, 4, 5}

In [77]:
```
1 s1
```

Out[77]: {2, 3, 4, 5}

In [78]:
```
1 s2
```

Out[78]: {1, 2, 3, 4, 5}

In [80]:
```
1 s1.symmetric_difference_update(s2)
```

In [81]:
```
1 s1
```

Out[81]: {1}

In [82]:
```
1 s1
```

Out[82]: {1}

In [83]:
```
1 s2
```

Out[83]: {1, 2, 3, 4, 5}

In [84]:
```
1 s3={5,6,7,8,9,10}
2 s2.union(s3)
```

Out[84]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [85]:
```
1 s2
```

Out[85]: {1, 2, 3, 4, 5}

In [86]:
```
1 s2.update([6,7,8,9])
```

In [87]:
```
1 s2
```

Out[87]: {1, 2, 3, 4, 5, 6, 7, 8, 9}

In [88]:
```
1 s2.update({10,11})
```

In [89]:
```
1 s2
```

Out[89]: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}

In [91]:
```
s4={21,22,23,3,4,5}
s5={20,22,33,31}
s4.symmetric_difference_update(s5)
```

In [92]:
```
s4
```

Out[92]: {3, 4, 5, 20, 21, 23, 31, 33}

Dictionary:

-dict(), (keys:value) -> items()
- example: s = {'name':'Rajesh','age':25,'sal':3400}
- ss = {'name':['rajesh','suresh']}

In [93]:
```
s = {'name':'Rajesh','age':25,'sal':3400}
s
```

Out[93]: {'name': 'Rajesh', 'age': 25, 'sal': 3400}

In [94]:
```
print(s['name'])
```

```
Rajesh
```

In [95]:
```
s.keys()
```

Out[95]: dict_keys(['name', 'age', 'sal'])

In [96]:
```
s.values()
```

Out[96]: dict_values(['Rajesh', 25, 3400])

In [98]:
```
for i in s.keys():
    print(i,end=" ")
```

```
name age sal
```

In [99]:
```
for i in s.values():
    print(i,end=" ")
```

```
Rajesh 25 3400
```

In [100]:
```
s.items()
```

Out[100]: dict_items([('name', 'Rajesh'), ('age', 25), ('sal', 3400)])

In [101]:
```
for i in s.items():
    print(i)
```

```
('name', 'Rajesh')
('age', 25)
('sal', 3400)
```

In [105]:
```
dir(dict)
```

Out[105]:
```
['__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'clear',
 'copy',
 'fromkeys',
 'get',
 'items',
 'keys',
 'pop',
 'popitem',
 'setdefault',
 'update',
 'values']
```

In [106]:
```
s
```

Out[106]:
```
{'name': 'Rajesh', 'age': 25, 'sal': 3400}
```

In [107]:
```
1 s.update({'sub':['maths','social','english']})
```

In [108]:
```
1 s
```

Out[108]:
```
{'name': 'Rajesh',
 'age': 25,
 'sal': 3400,
 'sub': ['maths', 'social', 'english']}
```

In [109]:
```
1 print(s)
```

```
{'name': 'Rajesh', 'age': 25, 'sal': 3400, 'sub': ['maths', 'social', 'englis
h']}
```

In [111]:
```
1 s.get('name')
```

Out[111]:
```
'Rajesh'
```

In [112]:
```
1 s.get('sub')
```

Out[112]:
```
['maths', 'social', 'english']
```

In [114]:
```
1 s.pop('name')
```

Out[114]:
```
'Rajesh'
```

In [115]:
```
1 s
```

Out[115]:
```
{'age': 25, 'sal': 3400, 'sub': ['maths', 'social', 'english']}
```

In [116]:
```
1 s1 = s.copy()
2 print(s1)
```

```
{'age': 25, 'sal': 3400, 'sub': ['maths', 'social', 'english']}
```

In [119]:
```
1 s.popitem()
```

Out[119]:
```
('sal', 3400)
```

In [120]:
```
1 s
```

Out[120]:
```
{'age': 25}
```

In [121]:
```
1 s1
```

Out[121]:
```
{'age': 25, 'sal': 3400, 'sub': ['maths', 'social', 'english']}
```

In [129]:
```
1 s1
```

Out[129]:
```
{'age': 25,
 'sal': 3400,
 'sub': ['maths', 'social', 'english'],
 ('key1', 'key2'): 0}
```

In [130]:
```
1 s1.setdefault('sal')
```

Out[130]:
```
3400
```

In [135]:
```
1 s1.setdefault('mobile',"Nothing")
```

In [134]:
```
1 s1
```

Out[134]:
```
{'age': 25,
 'sal': 3400,
 'sub': ['maths', 'social', 'english'],
 ('key1', 'key2'): 0,
 'mobile': None}
```

In [136]:
```
1 s1.setdefault('None','Nothing')
```

Out[136]:
```
'Nothing'
```

In [137]:
```
1 s1
```

Out[137]:
```
{'age': 25,
 'sal': 3400,
 'sub': ['maths', 'social', 'english'],
 ('key1', 'key2'): 0,
 'mobile': None,
 'None': 'Nothing'}
```

In [138]:
```
1 s1.popitem()
```

Out[138]:
```
('None', 'Nothing')
```

In [139]:
```
1 s1
```

Out[139]:
```
{'age': 25,
 'sal': 3400,
 'sub': ['maths', 'social', 'english'],
 ('key1', 'key2'): 0,
 'mobile': None}
```

In [150]:
```
1 s1
```

Out[150]:
```
{'age': 25,
 'sal': 3400,
 'sub': ['maths', 'social', 'english'],
 ('key1', 'key2'): 0,
 'mobile': None}
```

In [140]:
```
1 s1.fromkeys('sub')
```

Out[140]:
```
{'s': None, 'u': None, 'b': None}
```

In [144]:
```
1 dict = {'a':2,'b':'hello','c':'hai'}
```

In [145]:
```
1 dict
```

Out[145]:
```
{'a': 2, 'b': 'hello', 'c': 'hai'}
```

In [153]:
```
1 print(dict.fromkeys('c'))
```

```
{'c': None}
```

In [148]:
```
1 dict['c']
```

Out[148]:
```
'hai'
```

In [154]:
```
1 s1
```

Out[154]:
```
{'age': 25,
 'sal': 3400,
 'sub': ['maths', 'social', 'english'],
 ('key1', 'key2'): 0,
 'mobile': None}
```

In [155]:
```
1 s1
```

Out[155]:
```
{'age': 25,
 'sal': 3400,
 'sub': ['maths', 'social', 'english'],
 ('key1', 'key2'): 0,
 'mobile': None}
```

In [156]:
```
1 s1.setdefault('age')
```

Out[156]:
```
25
```

In [160]:
```
1 s1.setdefault('student1')
```

In [161]:
```
1 s1
```

Out[161]:
```
{'age': 25,
 'sal': 3400,
 'sub': ['maths', 'social', 'english'],
 ('key1', 'key2'): 0,
 'mobile': None,
 'student': 'Ramu',
 'student1': None}
```

In [162]:
```
1 s1.setdefault('course','B-Tech')
```

Out[162]:
```
'B-Tech'
```

In [163]:
```
1 s1
```

Out[163]:
```
{'age': 25,
 'sal': 3400,
 'sub': ['maths', 'social', 'english'],
 ('key1', 'key2'): 0,
 'mobile': None,
 'student': 'Ramu',
 'student1': None,
 'course': 'B-Tech'}
```

In [164]:
```
1 s1['student1']="Ravi"
```

In [165]:
```
1 s1
```

Out[165]:
```
{'age': 25,
 'sal': 3400,
 'sub': ['maths', 'social', 'english'],
 ('key1', 'key2'): 0,
 'mobile': None,
 'student': 'Ramu',
 'student1': 'Ravi',
 'course': 'B-Tech'}
```

In [174]:
```
1 d1={}
2 li=["name","role","Course"]
3 li1=["Hello","Hi","Bye"]
4 d1=d1.fromkeys(li,li1)
```

In [175]:
```
1 d1
```

Out[175]:
```
{'name': ['Hello', 'Hi', 'Bye'],
 'role': ['Hello', 'Hi', 'Bye'],
 'Course': ['Hello', 'Hi', 'Bye']}
```

Packages and Modules:

```
Package -> folder
    Module -> first.py
        Functions -> any(user task has to be assigned)
```

In [176]:
```
1 import math
```

In [177]:
```
1 math.pow(2,7)
```

Out[177]:
```
128.0
```

In [179]:
```
1 print(dir(math))
```

```
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acos
h', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cos
h', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floo
r', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfini
te', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'mod
f', 'nan', 'pi', 'pow', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan',
'tanh', 'tau', 'trunc']
```

In [180]:
```
1 math.pi
```

Out[180]:
```
3.141592653589793
```

In [181]:
```
1 math.factorial(100)
```

Out[181]:
```
933262154439441526816992388562667004907159682643816214685929638952175999932299156089414639761565182862536979208272237582511852109168640000000000000000000000
```

In [213]:
```
1 from IVC2 import first as p
2 n = int(input("Enter a number: "))
3 p.pat(n)
```

```
Enter a number: 3
***
* *
***
```

In [218]:
```
1 from IVC2 import second
2 n = input().split()
3 print(second.Evenlist(n))
```

```
4 5 6 7 8
[4, 6, 8]
```

In [219]:
```
1 s = 'dskjhfkdkj324534@#$%'
```

In [220]:
```
dc = cc = sc = 0
for i in s:
    if i.isdigit():
        dc+=1
    elif i.isalpha():
        cc+=1
    else:
        sc+=1
print(dc,sc,cc)
```

```
6 4 10
```

In [ ]:
```

```

In [ ]:
```

```

Input: 5

Output:

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

In [3]:
```
s = 'amazon nazaom'
```

In [4]:
```
s = s.split()
```

In [5]:
```
s
```

Out[5]:
```
['amazon', 'nazaom']
```

In [6]:
```
if s[0] == s[1]:
    print('yes')
else:
    print('no')
```

```
no
```

In [229]:
```
import math
```

In [230]:
```
1 math.__doc__
```

Out[230]:
```
'This module provides access to the mathematical functions\ndefined by the C st
andard.'
```

In [231]:
```
1 len.__doc__
```

Out[231]:
```
'Return the number of items in a container.'
```

In [1]:
```
1  def Evenlist(s):
2      '''Sample Example to print the even numebrs
3      in a list by using Modules'''
4      p = []
5      for i in range(len(s)):
6          if int(s[i])%2==0:
7              p.append(int(s[i]))
8      return p
9  n = input().split()
10 print(Evenlist(n))
```

```
2 4 5 66
[2, 4, 66]
```

In [5]:
```
1 import math
2 math.__doc__
```

Out[5]:
```
'This module provides access to the mathematical functions\ndefined by the C st
andard.'
```

In [7]:
```
1 print(print.__doc__)
```

```
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file:  a file-like object (stream); defaults to the current sys.stdout.
sep:   string inserted between values, default a space.
end:   string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
```

In [9]:
```
1 print(Evenlist.__doc__)
```

```
Sample Example to print the even numebrs
    in a list by using Modules
```

In [1]:
```
1 from IVC2 import second
```

In [2]:
```
1 second.__doc__
```

Out[2]:
```
' EvenList function'
```

In [7]:
```
s=input().split(" ")
if(sorted(s[0])==sorted(s[1])):
    print("Yes")
else:
    print("No")
```

```
amazon naazom
Yes
```

In [8]:
```
s="amazon"
sorted(s)
```

Out[8]:
```
['a', 'a', 'm', 'n', 'o', 'z']
```

In [9]:
```
s1="naazom"
sorted(s1)
```

Out[9]:
```
['a', 'a', 'm', 'n', 'o', 'z']
```

In [ ]:
```

```