```
t=(334,32,'netrich','gnt',0o63)
print(t)
```

```
(334, 32, 'netrich', 'gnt', 51)
```

```
l=[3,5,7,11,17,23,97,29]
#l=[5,10,22,11,15,17,20,30,23,43,91,29]
#l=[5,10,22,11,15,17,20,30,23,91,43]
temp = 'a'
for i in range(len(l)):
  for j in range(2,l[i]):
    if l[i] % j == 0:
      break
  else:
    if temp=='a':
      temp = i
    else:
      l[temp],l[i] = l[i],l[temp]
      temp='a'
print(l)
```

```
[5, 3, 11, 7, 23, 17, 29, 97]
```

```
s=['hello','good','morning']
#s[1]='python'
s[1:3]=['python','java']
print(s)
s[1:3]=['python']
print(s)
s[1:3]='python'
print(s)
```

```
['hello', 'python', 'java']
['hello', 'python']
['hello', 'p', 'y', 't', 'h', 'o', 'n']
```

```
def combinations(a,s1=''):
  if len(a)==0:
    print(s1)
  else:
    for i in range(len(a)):
      combinations(a[:i]+a[i+1:],s1+a[i])
combinations('abc')
```

```
abc
acb
bac
bca
cab
cba
```

```
l=[1,2,[3,4,5,[2.5,6,7,{8,9,10}],(0,'python'),11],12,13]
nl=[]
def new_list(l):
  for i in range(len(l)):
    if isinstance(l[i],int):
      nl.append(l[i])
      l[i]=''
    elif isinstance(l[i],list or set or tuple):
      new_list(l[i])
new_list(l)
print(nl)
```

```
[1, 2, 3, 4, 5, 6, 7, 11, 12, 13]
```

```
def unique(l,l1=[]):
  [l1:=l1+[j] for i in l for j in i if j not in l1]
  return l1

l=['python','java','html']
print(unique(l))
```

```
['p', 'y', 't', 'h', 'o', 'n', 'j', 'a', 'v', 'm', 'l']
```

```
l=['python','java','html']
k=lambda l1=[] : [l1:=l1+[j] for i in l for j in i if j not in l1]
k()
```

```
[['p'],
 ['p', 'y'],
 ['p', 'y', 't'],
 ['p', 'y', 't', 'h'],
 ['p', 'y', 't', 'h', 'o'],
 ['p', 'y', 't', 'h', 'o', 'n'],
 ['p', 'y', 't', 'h', 'o', 'n', 'j'],
 ['p', 'y', 't', 'h', 'o', 'n', 'j', 'a'],
 ['p', 'y', 't', 'h', 'o', 'n', 'j', 'a', 'v'],
 ['p', 'y', 't', 'h', 'o', 'n', 'j', 'a', 'v', 'm'],
 ['p', 'y', 't', 'h', 'o', 'n', 'j', 'a', 'v', 'm', 'l']]
```

```
class new:
  def num(l,m,n):
    print(l)
    print(m)
    print(n)
  @staticmethod
  def add(a,b):
    print('sum =',a+b)
```

```
test1=new()
test1.num(1,2)

test1.x=10
test1.num(test1.x,5)
```

```
<__main__.new object at 0x7fb802962dc0>
1
2
<__main__.new object at 0x7fb802962dc0>
10
5
```

```
new.num(1,2,3)
```

```
1
2
3
```

```
test2=new()
test2.add(1,2)
```

```
sum = 3
```

```
new.add(1,2)
```

```
sum = 3
```

```
def validate(f):
  def p1(num):
    if isinstance(num,int):
      f(num)
    else:
      print("invalid input")
  return p1

@validate
def strong_no(s):
  r=0
  s1=s
  sum =0
  while s!=0:
    r = s%10
    fact=1
    for i in range(1,r+1):
      fact=fact*i
    sum=sum+fact
    s=s//10
  if s1 == sum :
    print("strong number")
  else:
    print("not a strong")

strong_no(145)
```

```
strong number
```