->Sort,Filter And Search In Array.

In [1]:
```
#SORT(ESEDING ORDER)
import numpy as np
ar = np.array([5,7,1,6])
print (np.sort(ar))
```

```
[1 5 6 7]
```

In [3]:
```
#SEARCH
import numpy as np
ar = np.array([5,7,1,6])
s = np.where(ar == 7)
print (s)
```

```
(array([1], dtype=int64),)
```

In [5]:
```
#SEARCHSORTED(always array is sorted )
import numpy as np
ar = np.array([1,2,3,4,5])
ss = np.searchsorted(ar,2)
print (ss)
```

```
1
```

In [7]:
```
#FILTER
import numpy as np
ar = np.array([20,30,40,50])
fa= [True,False,True,False]#(je position ma true hase e avse)
New = ar[fa]
print(New)
```

```
[20 40]
```

->Aggreating Functions In Array.

In [9]:
```
#MAX VLAUE AND MIN VALUE,MEAN,CUMOPERATION.
import numpy as np
a = np.array([20,40,60,70])
print(np.max(a))
print(np.min(a))
print(np.mean(a))
print(np.cumsum(a))#(value+value)
print(np.cumproduct(a))
```

```
70
20
47.5
[ 20  60 120 190]
[     20     800   48000 3360000]
```

In [11]:
```
a = [100,150,199,200,250,130]
b = [10,50,30,40,30,10]
price = np.array(a)
quantity = np.array(b)
print(price,"\n",quantity)
print()

print(np.cumprod([price,quantity],axis =0))#price*quantity
```

```
[100 150 199 200 250 130]
 [10 50 30 40 30 10]

[[ 100  150  199  200  250  130]
 [1000 7500 5970 8000 7500 1300]]
```

->Statistical Functions In Array.

In [13]:

```
import numpy as np
import statistics as stats
baked_food = [200,300,150,130,200,280,170,188]#multipe time je value hoi ave
a = np.array(baked_food)
print(np.mean(baked_food))#sum of all the values/number of values
print(np.median(baked_food))#1.short the array 2.then center the value 3.the
print(stats.mode(baked_food))
print(np.std(baked_food))#standard deviation
print(np.var(baked_food))#variance
```

```
202.25
194.0
200
55.68157235567257
3100.4375
```

In [15]:

```
#COFFIECINT OF CORRELATION.
#-1 REPRESENT INVERSELY PROPORTIONAL RELATIONSHIP
#+1 REPRSENTS PROPORTIONAL RELATIONSHIP
#0 MEANS NO RELTIONSHIP
tobacco_consumption = [30,50,10,30,50,40]
deaths = [100,120,70,100,120,112]

print(np.corrcoef([tobacco_consumption,deaths]))
```

```
[[1.         0.99015454]
 [0.99015454 1.        ]]
```

In [17]:

```
#EXAMPLE OF CORRELATION.
price = [300,100,350,150,200]
sales = [10,20,7,17,3]

print(np.corrcoef([price,sales]))
```

```
[[ 1.         -0.66621445]
 [-0.66621445  1.        ]]
```