In [5]:

```
#sol1
import numpy as np

arr=np.array([67,56,45,89,99,76])
even = arr % 2 == 0
odd = arr % 2 != 0
print("All even elements of the original array arr:",arr[even])
print("All odd elements of the original array arr:",arr[odd])
arr = np.where(arr % 2 == 0,0,arr)
print("Original array in which 0 is there at the place of even number:",arr)
```

```
All even elements of the original array arr [56 76]
All odd elements of the original array arr [67 45 89 99]
Original array in which 0 is there at the place of even number [67  0 45 89 99  0]
```

In [30]:

```
#sol2
import numpy as np

arr=np.array([67,56,45,89,99,76])
copy_arr = np.where(arr % 2 != 0,'odd',arr)
print("Original array is: ",arr)
print("Array after required change is: ",copy_arr)
```

```
Original array is:  [67 56 45 89 99 76]
Array after required change is:  ['odd' '56' 'odd' 'odd' 'odd' '76']
```

In [45]:

```
a = np.array(['How','Are','You'])
b = np.repeat(a, 3)
c = np.tile(a, 3)
b = np.append(b,c,0) #list without comma
joined_string = ",".join(b) #string with comma
my_list = joined_string.split(",") #list with comma
print(my_list)
```

```
['How', 'How', 'How', 'Are', 'Are', 'Are', 'You', 'You', 'You', 'How', 'Are', 'You', 'How', 'Are', 'You', 'How', 'Are',
'You']
```

In [15]:

```
#intersect1d method
array1 = np.array(['a','A','E','b','c','B'])
print("Array1: ",array1)
array2 = np.array(['a','Y','B'])
print("Array2: ",array2)
print("Common values between two arrays:")
print(np.intersect1d(array1, array2))
```

```
Array1:  ['a' 'A' 'E' 'b' 'c' 'B']
Array2:  ['a' 'Y' 'B']
Common values between two arrays:
['B' 'a']
```

In [20]:
```
my_array = np.arange(12).reshape(3, 4)
print("Original array:")
print(my_array)

my_array[:,[2,3]] = my_array[:,[3,2]]
print("\nAfter swapping last 2 columns of the given array: ")
print(my_array)

my_array[[0,1],:] = my_array[[1,0],:]
print("\nAfter swapping starting 2 rows of the given array: ")
print(my_array)
```

```
Original array:
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]

After swapping last 2 columns of the given array:
[[ 0  1  3  2]
 [ 4  5  7  6]
 [ 8  9 11 10]]

After swapping starting 2 rows of the given array:
[[ 4  5  7  6]
 [ 0  1  3  2]
 [ 8  9 11 10]]
```

In [25]:
```
#The uniform() method returns random floating number between the two specified numbers (both included).
rand_arr = np.random.uniform(6,12, size=(5,3))
print(rand_arr)
```

```
[[10.56216378 11.01722648  8.05407174]
 [ 7.66141252 11.1867239   8.99687268]
 [11.6872732   7.76161484 10.58513173]
 [ 6.74894499  6.40590414  8.35119122]
 [11.89893807 11.62035001  7.55383214]]
```

In [52]:

```
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])
#print(iris_2d[:4]) #4 rows and all columns

print(iris_2d) #all rows and all columns
```

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]
 [5.4 3.7 1.5 0.2]
 [4.8 3.4 1.6 0.2]
 [4.8 3.  1.4 0.1]
 [4.3 3.  1.1 0.1]
 [5.8 4.  1.2 0.2]
 [5.7 4.4 1.5 0.4]
 [5.4 3.9 1.3 0.4]
 [5.1 3.5 1.4 0.3]
 [5.7 3.8 1.7 0.3]
```

In [53]:

```
sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])
mu, med, sd = np.mean(sepallength), np.median(sepallength), np.std(sepallength)
print(mu, med, sd)
```

```
5.843333333333334 5.8 0.8253012917851409
```

In [ ]: