```
In [1]:  import numpy as np
         import random
```

```
In [2]:  np.random.seed(21)
```

Calculate the average value of the second column

```
In [3]:  random_integers = np.random.randint(1,high=500000, size=(20, 5))
         avg = np.average(random_integers[:,1])
         print("The average value of the second column = ",avg)
```

```
The average value of the second column =  214895.8
```

The average value of the first 5 rows of the third and fourth columns

```
In [4]:  print(np.average(random_integers[0:5,2:4]))
```

```
286058.5
```

Create a 10 x 10 array from random_integers array named arrayTenByTen

```
In [5]:  arrayTenByTen = random_integers.reshape(10,10)
         arrayTenByTen
```

```
Out[5]:  array([[ 80842, 333008, 202553, 140037,  81969,  63857,  42105, 261540,
                 481981, 176739],
                [489984, 326386, 110795, 394863,  25024,  38317,  49982, 408830,
                 485118,  16119],
                [407675, 231729, 265455, 109413, 103399, 174677, 343356, 301717,
                 224120, 401101],
                [140473, 254634, 112262,  25063, 108262, 375059, 406983, 208947,
                 115641, 296685],
                [444899, 129585, 171318, 313094, 425041, 188411, 335140, 141681,
                  59641, 211420],
                [287650,   8973, 477425, 382803, 465168,   3975,  32213, 160603,
                 275485, 388234],
                [246225,  56174, 244097,   9350, 496966, 225516, 273338,  73335,
                 283013, 212813],
                [ 38175, 282399, 318413, 337639, 379802, 198049, 101115, 419547,
                 260219, 325793],
                [148593, 425024, 348570, 117968, 107007,  52547, 180346, 178760,
                 305186, 262153],
                [ 11835, 449971, 494184, 472031, 353049, 476442,  35455, 191553,
                 384154,  29917]])
```

Create a new array from the arrayTenByTen that is one rank and called arrayTenFlat

arrayTenFlat

```
In [6]: arrayTenFlat = arrayTenByTen.ravel()
        print(arrayTenFlat)
        arrayTenFlat.shape
```

```
[ 80842 333008 202553 140037  81969  63857  42105 261540 481981 176739
  489984 326386 110795 394863  25024  38317  49982 408830 485118  16119
  407675 231729 265455 109413 103399 174677 343356 301717 224120 401101
  140473 254634 112262  25063 108262 375059 406983 208947 115641 296685
  444899 129585 171318 313094 425041 188411 335140 141681  59641 211420
  287650   8973 477425 382803 465168   3975  32213 160603 275485 388234
  246225  56174 244097   9350 496966 225516 273338  73335 283013 212813
   38175 282399 318413 337639 379802 198049 101115 419547 260219 325793
  148593 425024 348570 117968 107007  52547 180346 178760 305186 262153
   11835 449971 494184 472031 353049 476442  35455 191553 384154  29917]
```

```
Out[6]: (100,)
```

What is the sum of arrayTenFlat?

```
In [7]: arrayTenFlat.sum()
```

```
Out[7]: 23526182
```

Iterate through arrayTenFlat

```
In [9]: #checking how many items in the array
        count = 0
        for i in arrayTenFlat:
            count = count + 1
        print(count, "elements")
```

```
100 elements
```

What is the value of the element (9,2) in random_integers array?

In [5]:
```python
print(random_integers)
random_integers[9][2]
```
```
[489984 326386 110795 394863  25024]
[ 38317  49982 408830 485118  16119]
[407675 231729 265455 109413 103399]
[174677 343356 301717 224120 401101]
[140473 254634 112262  25063 108262]
[375059 406983 208947 115641 296685]
[444899 129585 171318 313094 425041]
[188411 335140 141681  59641 211420]
[287650   8973 477425 382803 465168]
[  3975  32213 160603 275485 388234]
[246225  56174 244097   9350 496966]
[225516 273338  73335 283013 212813]
[ 38175 282399 318413 337639 379802]
[198049 101115 419547 260219 325793]
[148593 425024 348570 117968 107007]
[ 52547 180346 178760 305186 262153]
[ 11835 449971 494184 472031 353049]
[476442  35455 191553 384154  29917]]
```

Out[5]: 141681

What is the data type of arrayTenFlat?

In [16]:
```python
type(arrayTenFlat)
```

Out[16]: numpy.ndarray

In arrayTenFlat replace the value in index 5 with 42

In [17]:
```python
arrayTenFlat[5]= 42
arrayTenFlat
```

Out[17]:
```
array([ 80842, 333008, 202553, 140037,  81969,     42,  42105, 261540,
        481981, 176739, 489984, 326386, 110795, 394863,  25024,  38317,
         49982, 408830, 485118,  16119, 407675, 231729, 265455, 109413,
        103399, 174677, 343356, 301717, 224120, 401101, 140473, 254634,
        112262,  25063, 108262, 375059, 406983, 208947, 115641, 296685,
        444899, 129585, 171318, 313094, 425041, 188411, 335140, 141681,
         59641, 211420, 287650,   8973, 477425, 382803, 465168,   3975,
         32213, 160603, 275485, 388234, 246225,  56174, 244097,   9350,
        496966, 225516, 273338,  73335, 283013, 212813,  38175, 282399,
        318413, 337639, 379802, 198049, 101115, 419547, 260219, 325793,
        148593, 425024, 348570, 117968, 107007,  52547, 180346, 178760,
        305186, 262153,  11835, 449971, 494184, 472031, 353049, 476442,
         35455, 191553, 384154,  29917])
```

Save the array random_integers to a file. List the directory showing the saved array.

```
In [22]: np.save("numpy_file.npy",random_integers)
```

```
In [23]: pwd
```

```
Out[23]: '/Users/nididier/Desktop/PYTHON/PY_Output'
```

```
In [24]: ls
```

```
Conditionals.docx              guessGame .pdf
Dictionaries .pdf              guessGame.ipynb
Dictionaries.ipynb             mbox-short.txt
File handling .pdf             my_file.npy
File handling.ipynb            numpy_file.npy
Numpy.ipynb                    overtimePay - Jupyter Notebook.pdf
Untitled.ipynb                 overtimePay.ipynb
Use Variables.pdf              romeo.txt
WEEK-01.pdf                    wordFrequency.ipynb
conditionals1.pdf              workWithLists.ipynb
functionPay.pdf                workWithLists.pdf
```

```
In [ ]:
```