In [5]:

```
######## Question 1 part 1
import numpy as np

np.random.seed(89)
r_arr1 = np.random.rand(5,5)

rows, columns = r_arr1.shape
row_indent = 0
diag_sum = 0
diag_sum2 = 0

# Option 1 for Doing both Diagonals
for x in range(rows):
    diag_sum += r_arr1[row_indent, x]
    row_indent += 1

row_indent -= 1

for y in range(rows):
    diag_sum += r_arr1[row_indent, y]
    row_indent -= 1

if (rows % 2) == 1:
    mid_row = np.floor(rows/2).astype(int)
    mid_column = np.floor(columns/2).astype(int)
    diag_sum -= r_arr1[mid_column, mid_row]


# Option 2 for just the left-to-right, top-to-down diagonal
d_arr1 = np.diag(r_arr1)
for d_i in range(len(d_arr1)):
    diag_sum2 += d_arr1[d_i]

######## Question 1 part 2
t_arr1 = np.transpose(r_arr1)

f_arr1 = np.ndarray.flatten(r_arr1)

s_arr1 = np.array_split(r_arr1, 5)
```

In [6]:

```
print(r_arr1)
print("Method 1: ", diag_sum)
print("Method 2: ", diag_sum2)
```

```
[[0.49969432 0.25593713 0.25810063 0.09692171 0.56418511]
 [0.01599007 0.15259523 0.48024773 0.09987276 0.41696389]
 [0.91365081 0.35071951 0.11460437 0.71260839 0.10188615]
 [0.40570044 0.66548144 0.13835937 0.83043309 0.12319969]
 [0.58779155 0.06309849 0.49710274 0.92839462 0.80603084]]
Method 1:  4.3206887196012165
Method 2:  2.4033578517438836
```

In [32]:
```
print("Transpose: ")
print(t_arr1)
print("Flattened: ")
print(f_arr1)
print("Split: ")
print(s_arr1)
```

```
Transpose:
[[0.49969432 0.01599007 0.91365081 0.40570044 0.58779155]
 [0.25593713 0.15259523 0.35071951 0.66548144 0.06309849]
 [0.25810063 0.48024773 0.11460437 0.13835937 0.49710274]
 [0.09692171 0.09987276 0.71260839 0.83043309 0.92839462]
 [0.56418511 0.41696389 0.10188615 0.12319969 0.80603084]]
Flattened:
[0.49969432 0.25593713 0.25810063 0.09692171 0.56418511 0.01599007
 0.15259523 0.48024773 0.09987276 0.41696389 0.91365081 0.35071951
 0.11460437 0.71260839 0.10188615 0.40570044 0.66548144 0.13835937
 0.83043309 0.12319969 0.58779155 0.06309849 0.49710274 0.92839462
 0.80603084]
Split:
[array([[0.49969432, 0.25593713, 0.25810063, 0.09692171, 0.56418511]]), array([[0.01
599007, 0.15259523, 0.48024773, 0.09987276, 0.41696389]]), array([[0.91365081, 0.350
71951, 0.11460437, 0.71260839, 0.10188615]]), array([[0.40570044, 0.66548144, 0.1383
5937, 0.83043309, 0.12319969]]), array([[0.58779155, 0.06309849, 0.49710274, 0.92839
462, 0.80603084]])]
```

In [34]:

```
######## Question 2 part 1
import numpy as np

np.random.seed(100)
x = np.random.randint(0, 1000, size = (10, 10))

x_evencount = 0
x_rows, x_columns = x.shape

for row in range(x_rows):
    for column in range(x_columns):
        if (x[row, column] % 2) == 0:
            x_evencount += 1
#             print(x[row, column])

######## Question 2 part 2

norm_arr = np.random.normal(1, 0.5, (8, 9))
norm_rows, norm_columns = norm_arr.shape
norm_count = 0
norm_sum = 0

for i in range(norm_rows):
    for j in range(norm_columns):
        if ((i+j)%5 == 0):
            norm_count += 1
            norm_sum += norm_arr[i, j]


norm_multof5_mean = norm_sum / norm_count
```

In [36]:

```
x_evencount
```

Out[36]:

```
50
```

In [37]:

```
norm_multof5_mean
```

Out[37]:

```
1.1510333789549485
```

In [ ]: