This code has been compiled by

Rohit Lal BT17ECE067 https://rohitlal.live/

1 Import Libraries and initialize the matrix

[1]:
```
import numpy as np

a = [[63, 134, 79, 35, 152, 31, 146, 72, 9,81, 105, 26, 79, 35, 152, 31, 146,␣
 ↪72, 9, 81],
    [99 ,95 ,61 ,92 ,131, 163, 177, 76 ,10, 1 ,9 ,173, 61, 92, 131, 163, 177, 76␣
 ↪,10, 1],
    [111, 198, 90 ,96 ,26 ,88 ,77 ,138 ,5 ,136 ,185 ,92 ,90 ,96 ,26 ,88 ,77 ,138␣
 ↪,5 ,136],
    [40, 8 ,104, 175, 132, 36, 169, 66, 191 ,64, 7 ,121 ,104, 175 ,132, 36, 169,␣
 ↪66 ,191, 64],
    [58, 12, 43, 198 ,190 ,77 ,171 ,61 ,130, 67 ,52 ,165 ,43 ,198 ,190 ,77 ,171,␣
 ↪61 ,130 ,67],
    [89, 80 ,82 ,159 ,126, 101, 58 ,136 ,97 ,7, 106, 47, 82, 159, 126, 101, 58,␣
 ↪136, 97, 7],
    [160, 81 ,151, 158, 106, 28, 196, 100, 38, 34, 174, 160, 151, 158, 106, 28␣
 ↪,196 ,100, 38 ,34],
    [145, 85, 146, 146, 106, 74 ,27 ,162 ,41 ,42 ,53 ,45 ,146, 146, 106, 74 ,27␣
 ↪,162, 41, 42],
    [149, 167, 50, 93 ,162, 169, 121, 75, 66, 136, 117, 28, 50, 93 ,162, 169,␣
 ↪121, 75, 66, 136],
    [164 ,197, 154, 16 ,134 ,175 ,148 ,124 ,157 ,191 ,5 ,151 ,154, 16 ,134 ,175␣
 ↪,148 ,124 ,157, 191],
    [192, 31, 74 ,132 ,141, 65, 138, 20 ,167 ,30 ,90 ,160 ,74, 132, 141, 65,␣
 ↪138, 20 ,167 ,30],
    [174, 140, 157, 127, 176, 9 ,91 ,20 ,83 ,103 ,179, 104, 157, 127, 176, 9 ,91␣
 ↪,20, 83, 103],
    [147, 53 ,99, 20, 87, 8 ,113, 136, 65, 40 ,196 ,29 ,99 ,20 ,87 ,8 ,113 ,136␣
 ↪,65 ,40],
    [107, 41 ,8 ,86 ,195 ,43 ,155 ,20 ,160 ,112 ,144 ,97 ,8 ,86 ,195 ,43 ,155␣
 ↪,20 ,160, 112],
    [18 ,107 ,184 ,139 ,107 ,173 ,110 ,45 ,60 ,89 ,183 ,8 ,184 ,139 ,107 ,173␣
 ↪,110 ,45 ,60 ,89],
    [52 ,115 ,124 ,78 ,2, 112, 69 ,76 ,53 ,74 ,190 ,142, 124, 78 ,2 ,112 ,69 ,76␣
 ↪,53 ,74],
    [161 ,54 ,136 ,100 ,193 ,126 ,13 ,120 ,99 ,33 ,13 ,134 ,136 ,100, 193 ,126␣
 ↪,13 ,120 ,99 ,33],
    [179 ,145 ,140, 25 ,82 ,173 ,78 ,122 ,151 ,187 ,183 ,150 ,140, 25 ,82 ,173␣
 ↪,78 ,122 ,151, 187],
    [118, 130 ,175, 171, 52, 144 ,3 ,64 ,172 ,132, 153, 189, 175, 171, 52 ,144,␣
 ↪3 ,64 ,172, 132],
```

```
    [91 ,82 ,58 ,133, 16, 59 ,53 ,122 ,60 ,174 ,193, 64 ,58 ,133 ,16 ,59 ,53␣
 ↪,122 ,60 ,174]]

b = [[84, 62 ,175, 38, 159 ,123 ,90 ,115, 142 ,163, 80, 82, 62, 175, 38, 159,␣
 ↪123, 90 ,115 ,142],
    [58 ,70 ,156, 44, 80, 180, 53, 43, 122, 4 ,21 ,23, 70, 156, 44, 80 ,180 ,53␣
 ↪,43 ,122],
    [173, 0 ,64 ,47 ,155 ,194 ,8 ,32 ,196 ,196 ,96 ,71 ,0 ,64 ,47 ,155, 194 ,8␣
 ↪,32 ,196],
    [147, 167, 49 ,152 ,129 ,167, 25 ,7 ,110 ,106 ,123 ,79 ,167 ,49 ,152 ,129␣
 ↪,167 ,25 ,7 ,110],
    [142, 125, 80, 168 ,152 ,2 ,194 ,100 ,6 ,56 ,50 ,127 ,125 ,80 ,168, 152 ,2␣
 ↪,194 ,100, 6],
    [193, 129, 53, 72 ,103, 92 ,142 ,23 ,102 ,70 ,104, 23, 129, 53 ,72 ,103 ,92␣
 ↪,142, 23 ,102],
    [80, 67, 77, 198, 140, 60 ,161 ,33 ,150 ,15 ,166 ,192 ,67 ,77 ,198 ,140 ,60␣
 ↪,161 ,33 ,150],
    [25, 6, 118, 132, 84, 25, 187, 24, 189, 161 ,167 ,100, 6 ,118 ,132 ,84 ,25␣
 ↪,187 ,24 ,189],
    [177 ,5 ,31 ,133 ,3 ,163 ,143 ,128, 184, 133, 167 ,94 ,5 ,31 ,133 ,3 ,163,␣
 ↪143, 128, 184],
    [75 ,130, 67, 152, 162, 74 ,108 ,42 ,97 ,124 ,165, 102, 130, 67 ,152, 162,␣
 ↪74 ,108, 42, 97],
    [182, 144, 54 ,145 ,3 ,62 ,51 ,83 ,16 ,79 ,82 ,104 ,144 ,54 ,145, 3 ,62 ,51␣
 ↪,83 ,16],
    [3 ,97 ,19 ,17 ,127 ,36 ,128, 32 ,1 ,96 ,199, 189, 97, 19, 17 ,127 ,36, 128,␣
 ↪32 ,1],
    [40 ,154 ,186 ,41 ,79 ,151, 13, 196, 75 ,166 ,160 ,104 ,154, 186, 41 ,79␣
 ↪,151 ,13 ,196 ,75],
    [91 ,130 ,110 ,164 ,141 ,181 ,193 ,58 ,56 ,98 ,145 ,196 ,130 ,110 ,164 ,141␣
 ↪,181 ,193 ,58 ,56],
    [34 ,55 ,180 ,103, 118 ,16 ,27 ,95 ,169, 70, 198, 8 ,55 ,180 ,103 ,118, 16␣
 ↪,27 ,95 ,169],
    [164, 70, 164, 147, 95 ,195 ,9 ,107, 155, 55, 14 ,159 ,70 ,164 ,147 ,95 ,195␣
 ↪,9 ,107 ,155],
    [52, 27 ,98 ,180 ,93 ,32 ,146, 32, 132, 123 ,60 ,166 ,27 ,98, 180 ,93 ,32␣
 ↪,146, 32 ,132],
    [189, 134, 69, 187, 42 ,199, 60 ,26 ,70 ,95 ,186 ,132 ,134 ,69 ,187 ,42 ,199␣
 ↪,60 ,26 ,70],
    [135 ,93 ,94 ,182 ,139 ,46 ,122 ,52 ,74 ,112 ,9 ,188 ,93 ,94 ,182, 139, 46,␣
 ↪122, 52, 74],
    [62 ,8 ,13, 146 ,115 ,135 ,29 ,138 ,85 ,133, 104, 118 ,8 ,13 ,146 ,115, 135,␣
 ↪29 ,138 ,85]]

a,b = np.array(a), np.array(b)
```

2 Question 2.a : Find Transpose of Matrix

[2]:
```
a_transpose = a.T
b_transpose = b.T
print('a_transpose \n', a_transpose)
print('b_transpose \n',b_transpose)
```

```
a_transpose
 [[ 63  99 111  40  58  89 160 145 149 164 192 174 147 107  18  52 161 179
  118  91]
 [134  95 198   8  12  80  81  85 167 197  31 140  53  41 107 115  54 145
  130  82]
 [ 79  61  90 104  43  82 151 146  50 154  74 157  99   8 184 124 136 140
  175  58]
 [ 35  92  96 175 198 159 158 146  93  16 132 127  20  86 139  78 100  25
  171 133]
 [152 131  26 132 190 126 106 106 162 134 141 176  87 195 107   2 193  82
   52  16]
 [ 31 163  88  36  77 101  28  74 169 175  65   9   8  43 173 112 126 173
  144  59]
 [146 177  77 169 171  58 196  27 121 148 138  91 113 155 110  69  13  78
    3  53]
 [ 72  76 138  66  61 136 100 162  75 124  20  20 136  20  45  76 120 122
   64 122]
 [  9  10   5 191 130  97  38  41  66 157 167  83  65 160  60  53  99 151
  172  60]
 [ 81   1 136  64  67   7  34  42 136 191  30 103  40 112  89  74  33 187
  132 174]
 [105   9 185   7  52 106 174  53 117   5  90 179 196 144 183 190  13 183
  153 193]
 [ 26 173  92 121 165  47 160  45  28 151 160 104  29  97   8 142 134 150
  189  64]
 [ 79  61  90 104  43  82 151 146  50 154  74 157  99   8 184 124 136 140
  175  58]
 [ 35  92  96 175 198 159 158 146  93  16 132 127  20  86 139  78 100  25
  171 133]
 [152 131  26 132 190 126 106 106 162 134 141 176  87 195 107   2 193  82
   52  16]
 [ 31 163  88  36  77 101  28  74 169 175  65   9   8  43 173 112 126 173
  144  59]
 [146 177  77 169 171  58 196  27 121 148 138  91 113 155 110  69  13  78
    3  53]
 [ 72  76 138  66  61 136 100 162  75 124  20  20 136  20  45  76 120 122
   64 122]
 [  9  10   5 191 130  97  38  41  66 157 167  83  65 160  60  53  99 151
  172  60]
 [ 81   1 136  64  67   7  34  42 136 191  30 103  40 112  89  74  33 187
```

```
  132 174]]
b_transpose
 [[ 84  58 173 147 142 193  80  25 177  75 182   3  40  91  34 164  52 189
  135  62]
 [ 62  70   0 167 125 129  67   6   5 130 144  97 154 130  55  70  27 134
   93   8]
 [175 156  64  49  80  53  77 118  31  67  54  19 186 110 180 164  98  69
   94  13]
 [ 38  44  47 152 168  72 198 132 133 152 145  17  41 164 103 147 180 187
  182 146]
 [159  80 155 129 152 103 140  84   3 162   3 127  79 141 118  95  93  42
  139 115]
 [123 180 194 167   2  92  60  25 163  74  62  36 151 181  16 195  32 199
   46 135]
 [ 90  53   8  25 194 142 161 187 143 108  51 128  13 193  27   9 146  60
  122  29]
 [115  43  32   7 100  23  33  24 128  42  83  32 196  58  95 107  32  26
   52 138]
 [142 122 196 110   6 102 150 189 184  97  16   1  75  56 169 155 132  70
   74  85]
 [163   4 196 106  56  70  15 161 133 124  79  96 166  98  70  55 123  95
  112 133]
 [ 80  21  96 123  50 104 166 167 167 165  82 199 160 145 198  14  60 186
    9 104]
 [ 82  23  71  79 127  23 192 100  94 102 104 189 104 196   8 159 166 132
  188 118]
 [ 62  70   0 167 125 129  67   6   5 130 144  97 154 130  55  70  27 134
   93   8]
 [175 156  64  49  80  53  77 118  31  67  54  19 186 110 180 164  98  69
   94  13]
 [ 38  44  47 152 168  72 198 132 133 152 145  17  41 164 103 147 180 187
  182 146]
 [159  80 155 129 152 103 140  84   3 162   3 127  79 141 118  95  93  42
  139 115]
 [123 180 194 167   2  92  60  25 163  74  62  36 151 181  16 195  32 199
   46 135]
 [ 90  53   8  25 194 142 161 187 143 108  51 128  13 193  27   9 146  60
  122  29]
 [115  43  32   7 100  23  33  24 128  42  83  32 196  58  95 107  32  26
   52 138]
 [142 122 196 110   6 102 150 189 184  97  16   1  75  56 169 155 132  70
   74  85]]
```

3 Question 2.b : Find Inverse of Matrix

Matrix a is singular matrix hence it throws and error

[3]:
```
# a_inverse = np.linalg.inv(a)
b_inverse = np.linalg.inv(b)
print(b_inverse)
```

```
[[-5.18134715e-02 -2.07253886e-02  0.00000000e+00  1.55440415e-02
   4.04792746e-03 -5.18134715e-03  5.18134715e-03  2.07253886e-02
  -1.03626943e-02 -1.03626943e-02 -1.03626943e-02 -1.03626943e-02
   1.55440415e-02  0.00000000e+00  4.53367876e-03  4.14507772e-02
   1.55440415e-02 -2.07253886e-02 -2.07253886e-02  5.18134715e-03]
 [ 1.04455133e+14  1.09806481e+14 -7.90769883e+13 -2.51866852e+13
  -1.29628558e+12 -8.12628933e+13 -4.83975698e+13 -1.30149080e+14
   1.20127766e+14  1.15741166e+14 -9.82276844e+13  7.06862768e+13
  -5.07013988e+13 -1.67115373e+14  7.57629919e+12 -7.48069257e+13
   5.45966380e+13  1.36812895e+14  1.21147528e+14 -4.59678445e+13]
 [-4.75017357e+13 -1.17312923e+13  2.81053865e+13 -8.41094777e+12
   1.10004650e+13  1.39541220e+13  5.01097479e+12  2.99345365e+13
  -2.30765713e+13 -6.19529103e+12  1.24565739e+13 -1.30022057e+13
   2.10872777e+13  2.53805777e+13 -5.48898614e+12  1.90584907e+13
  -1.19659252e+13 -2.51772027e+13 -2.19470644e+13  3.43351338e+12]
 [-1.10216796e+13 -7.70656853e+12  2.16842391e+12  2.05079326e+13
   2.19724194e+13 -1.04151336e+13 -1.41891767e+13  4.74840511e+13
  -1.44161984e+13 -8.63596205e+12  2.09774710e+13  9.98991615e+12
  -1.33501264e+13  8.22490170e+12 -7.12984144e+12  3.83642410e+13
  -2.61858321e+13 -2.22578861e+13 -3.66202519e+13  6.78902192e+12]
 [-5.65342748e+13 -3.36062712e+13  1.66781736e+13  1.30880652e+13
  -7.03866127e+12  3.73018393e+13 -2.09092692e+13  6.56830086e+13
  -4.29961369e+13 -3.95585917e+13  4.60307161e+13 -1.54350121e+12
  -5.38010579e+12  7.05836737e+13  1.37979991e+13  5.63395871e+13
  -3.05655392e+13 -6.76640085e+13 -4.05639149e+13  3.00197588e+13]
 [-6.67036076e+13 -3.77503476e+13  3.47296129e+13  2.75880429e+13
   1.79505670e+13 -1.15393618e+13 -3.12729656e+13  1.23044564e+14
  -4.60255271e+13 -7.84421932e+12  9.64725554e+13  6.29845680e+11
  -2.94244584e+13  9.64906345e+13  1.37961592e+12  1.00964302e+14
  -9.05645712e+13 -1.13214746e+14 -8.16063908e+13  1.48842437e+13]
 [-8.05740345e+13 -4.38014786e+13  4.30085295e+13  1.85376322e+13
   1.23115901e+13  1.69236656e+13 -2.28141705e+13  9.35543405e+13
  -5.20813221e+13 -2.67268949e+13  7.92517707e+13 -7.92605879e+12
  -7.60075541e+12  1.00555785e+14  8.45819994e+12  8.33572197e+13
  -4.77707862e+13 -1.05038798e+14 -8.25630859e+13  1.79926989e+13]
 [ 1.27431431e+13  1.39533113e+13 -3.47893788e+12 -3.49890415e+12
  -3.37448314e+12 -4.31244974e+12  8.42991041e+12 -1.76108831e+13
   7.17467941e+12 -1.17414593e+12 -7.11085522e+12  8.54706303e+11
   2.30536052e+12 -1.74070059e+13 -8.29351537e+11 -2.32904604e+13
   9.36641203e+12  1.63372699e+13  1.65872358e+13 -8.19770117e+11]
 [ 1.56882131e+14  1.09807315e+14 -7.42809447e+13 -1.12934292e+14
  -7.63726296e+13 -1.66022348e+13  4.36142298e+13 -2.04228751e+14
   8.86738721e+13  1.41077666e+14 -9.32989862e+13  9.77525222e+12
```

```
 -7.00500207e+12 -1.79610294e+14 -1.95580872e+13 -1.60574724e+14
  1.00541872e+14  1.85560133e+14  1.55870548e+14 -2.37397464e+13]
[ 6.98371979e-03  4.93905856e-03 -1.88976587e-03 -1.08491273e-03
  6.56083048e-04 -7.21583041e-03 -4.05381120e-03 -3.09538685e-03
  5.39743924e-03  8.35343955e-03 -3.05332113e-03  2.07968367e-03
 -1.41915546e-03 -9.36265198e-03 -2.53933436e-03 -5.76398843e-03
  1.99148766e-03  7.61358165e-03  6.39829789e-03 -2.96435409e-03]
[-1.10085394e-02 -1.03207919e-02  5.68235431e-03  4.37011581e-03
  1.40410044e-03  5.52920696e-03 -1.13452176e-03  1.29607506e-02
 -8.63562038e-03 -1.21578449e-02  8.06754938e-03 -7.63248151e-04
  3.90773453e-04  1.60251873e-02  4.94963080e-03  9.45359103e-03
 -6.79874878e-03 -1.11780186e-02 -1.19164918e-02  4.10869544e-03]
[-6.75262076e-03 -6.25421000e-03  5.70379880e-03  2.21734351e-03
  6.94883927e-04  8.83946473e-04  4.70512300e-03  9.60101045e-03
 -6.91073383e-03 -7.70562238e-03  8.02361642e-03 -2.05094730e-04
  2.84861582e-03  7.90859337e-03 -2.80612281e-03  7.67444176e-03
 -4.73043804e-03 -9.37223549e-03 -7.09055022e-03  8.18043141e-04]
[-1.04455133e+14 -1.09806481e+14  7.90769883e+13  2.51866852e+13
  1.29628558e+12  8.12628933e+13  4.83975698e+13  1.30149080e+14
 -1.20127766e+14 -1.15741166e+14  9.82276844e+13 -7.06862768e+13
  5.07013988e+13  1.67115373e+14 -7.57629919e+12  7.48069257e+13
 -5.45966380e+13 -1.36812895e+14 -1.21147528e+14  4.59678445e+13]
[ 4.75017357e+13  1.17312923e+13 -2.81053865e+13  8.41094777e+12
 -1.10004650e+13 -1.39541220e+13 -5.01097479e+12 -2.99345365e+13
  2.30765713e+13  6.19529103e+12 -1.24565739e+13  1.30022057e+13
 -2.10872777e+13 -2.53805777e+13  5.48898614e+12 -1.90584907e+13
  1.19659252e+13  2.51772027e+13  2.19470644e+13 -3.43351338e+12]
[ 1.10216796e+13  7.70656853e+12 -2.16842391e+12 -2.05079326e+13
 -2.19724194e+13  1.04151336e+13  1.41891767e+13 -4.74840511e+13
  1.44161984e+13  8.63596205e+12 -2.09774710e+13 -9.98991615e+12
  1.33501264e+13 -8.22490170e+12  7.12984144e+12 -3.83642410e+13
  2.61858321e+13  2.22578861e+13  3.66202519e+13 -6.78902192e+12]
[ 5.65342748e+13  3.36062712e+13 -1.66781736e+13 -1.30880652e+13
  7.03866127e+12 -3.73018393e+13  2.09092692e+13 -6.56830086e+13
  4.29961369e+13  3.95585917e+13 -4.60307161e+13  1.54350121e+12
  5.38010579e+12 -7.05836737e+13 -1.37979991e+13 -5.63395871e+13
  3.05655392e+13  6.76640085e+13  4.05639149e+13 -3.00197588e+13]
[ 6.67036076e+13  3.77503476e+13 -3.47296129e+13 -2.75880429e+13
 -1.79505670e+13  1.15393618e+13  3.12729656e+13 -1.23044564e+14
  4.60255271e+13  7.84421932e+12 -9.64725554e+13 -6.29845680e+11
  2.94244584e+13 -9.64906345e+13 -1.37961592e+12 -1.00964302e+14
  9.05645712e+13  1.13214746e+14  8.16063908e+13 -1.48842437e+13]
[ 8.05740345e+13  4.38014786e+13 -4.30085295e+13 -1.85376322e+13
 -1.23115901e+13 -1.69236656e+13  2.28141705e+13 -9.35543405e+13
  5.20813221e+13  2.67268949e+13 -7.92517707e+13  7.92605879e+12
  7.60075541e+12 -1.00555785e+14 -8.45819994e+12 -8.33572197e+13
  4.77707862e+13  1.05038798e+14  8.25630859e+13 -1.79926989e+13]
[-1.27431431e+13 -1.39533113e+13  3.47893788e+12  3.49890415e+12
```

```
  3.37448314e+12  4.31244974e+12 -8.42991041e+12  1.76108831e+13
 -7.17467941e+12  1.17414593e+12  7.11085522e+12 -8.54706303e+11
 -2.30536052e+12  1.74070059e+13  8.29351537e+11  2.32904604e+13
 -9.36641203e+12 -1.63372699e+13 -1.65872358e+13  8.19770117e+11]
[-1.56882131e+14 -1.09807315e+14  7.42809447e+13  1.12934292e+14
  7.63726296e+13  1.66022348e+13 -4.36142298e+13  2.04228751e+14
 -8.86738721e+13 -1.41077666e+14  9.32989862e+13 -9.77525222e+12
  7.00500207e+12  1.79610294e+14  1.95580872e+13  1.60574724e+14
 -1.00541872e+14 -1.85560133e+14 -1.55870548e+14  2.37397464e+13]]
```

4 Question 2.c : Addition of matrix

This Follows commutative property

[4]:
```
add_1 = a + b
add_2 = b + a
print(add_1)
print()
print(add_2)
```

```
[[147 196 254  73 311 154 236 187 151 244 185 108 141 210 190 190 269 162
  124 223]
 [157 165 217 136 211 343 230 119 132   5  30 196 131 248 175 243 357 129
   53 123]
 [284 198 154 143 181 282  85 170 201 332 281 163  90 160  73 243 271 146
   37 332]
 [187 175 153 327 261 203 194  73 301 170 130 200 271 224 284 165 336  91
  198 174]
 [200 137 123 366 342  79 365 161 136 123 102 292 168 278 358 229 173 255
  230  73]
 [282 209 135 231 229 193 200 159 199  77 210  70 211 212 198 204 150 278
  120 109]
 [240 148 228 356 246  88 357 133 188  49 340 352 218 235 304 168 256 261
   71 184]
 [170  91 264 278 190  99 214 186 230 203 220 145 152 264 238 158  52 349
   65 231]
 [326 172  81 226 165 332 264 203 250 269 284 122  55 124 295 172 284 218
  194 320]
 [239 327 221 168 296 249 256 166 254 315 170 253 284  83 286 337 222 232
  199 288]
 [374 175 128 277 144 127 189 103 183 109 172 264 218 186 286  68 200  71
  250  46]
 [177 237 176 144 303  45 219  52  84 199 378 293 254 146 193 136 127 148
  115 104]
 [187 207 285  61 166 159 126 332 140 206 356 133 253 206 128  87 264 149
  261 115]
 [198 171 118 250 336 224 348  78 216 210 289 293 138 196 359 184 336 213
```

```
  218 168]
 [ 52 162 364 242 225 189 137 140 229 159 381  16 239 319 210 291 126  72
  155 258]
 [216 185 288 225  97 307  78 183 208 129 204 301 194 242 149 207 264  85
  160 229]
 [213  81 234 280 286 158 159 152 231 156  73 300 163 198 373 219  45 266
  131 165]
 [368 279 209 212 124 372 138 148 221 282 369 282 274  94 269 215 277 182
  177 257]
 [253 223 269 353 191 190 125 116 246 244 162 377 268 265 234 283  49 186
  224 206]
 [153  90  71 279 131 194  82 260 145 307 297 182  66 146 162 174 188 151
  198 259]]

[[147 196 254  73 311 154 236 187 151 244 185 108 141 210 190 190 269 162
  124 223]
 [157 165 217 136 211 343 230 119 132   5  30 196 131 248 175 243 357 129
   53 123]
 [284 198 154 143 181 282  85 170 201 332 281 163  90 160  73 243 271 146
   37 332]
 [187 175 153 327 261 203 194  73 301 170 130 200 271 224 284 165 336  91
  198 174]
 [200 137 123 366 342  79 365 161 136 123 102 292 168 278 358 229 173 255
  230  73]
 [282 209 135 231 229 193 200 159 199  77 210  70 211 212 198 204 150 278
  120 109]
 [240 148 228 356 246  88 357 133 188  49 340 352 218 235 304 168 256 261
   71 184]
 [170  91 264 278 190  99 214 186 230 203 220 145 152 264 238 158  52 349
   65 231]
 [326 172  81 226 165 332 264 203 250 269 284 122  55 124 295 172 284 218
  194 320]
 [239 327 221 168 296 249 256 166 254 315 170 253 284  83 286 337 222 232
  199 288]
 [374 175 128 277 144 127 189 103 183 109 172 264 218 186 286  68 200  71
  250  46]
 [177 237 176 144 303  45 219  52  84 199 378 293 254 146 193 136 127 148
  115 104]
 [187 207 285  61 166 159 126 332 140 206 356 133 253 206 128  87 264 149
  261 115]
 [198 171 118 250 336 224 348  78 216 210 289 293 138 196 359 184 336 213
  218 168]
 [ 52 162 364 242 225 189 137 140 229 159 381  16 239 319 210 291 126  72
  155 258]
 [216 185 288 225  97 307  78 183 208 129 204 301 194 242 149 207 264  85
  160 229]
 [213  81 234 280 286 158 159 152 231 156  73 300 163 198 373 219  45 266
  131 165]
```

```
[368 279 209 212 124 372 138 148 221 282 369 282 274  94 269 215 277 182
 177 257]
[253 223 269 353 191 190 125 116 246 244 162 377 268 265 234 283  49 186
 224 206]
[153  90  71 279 131 194  82 260 145 307 297 182  66 146 162 174 188 151
 198 259]]
```

5 Question 2.d :Subtraction

Doesnt follow commutative property

[5]:
```
sub_1 = a - b
sub_2 = b - a
print(sub_1,'\n \n',sub_2)
```

```
[[ -21   72  -96   -3   -7  -92   56  -43 -133  -82   25  -56   17 -140
   114 -128   23  -18 -106  -61]
 [  41   25  -95   48   51  -17  124   33 -112   -3  -12  150   -9  -64
    87   83   -3   23  -33 -121]
 [ -62  198   26   49 -129 -106   69  106 -191  -60   89   21   90   32
   -21  -67 -117  130  -27  -60]
 [-107 -159   55   23    3 -131  144   59   81  -42 -116   42  -63  126
   -20  -93    2   41  184  -46]
 [ -84 -113  -37   30   38   75  -23  -39  124   11    2   38  -82  118
    22  -75  169 -133   30   61]
 [-104  -49   29   87   23    9  -84  113   -5  -63    2   24  -47  106
    54   -2  -34   -6   74  -95]
 [  80   14   74  -40  -34  -32   35   67 -112   19    8  -32   84   81
   -92 -112  136  -61    5 -116]
 [ 120   79   28   14   22   49 -160  138 -148 -119 -114  -55  140   28
   -26  -10    2  -25   17 -147]
 [ -28  162   19  -40  159    6  -22  -53 -118    3  -50  -66   45   62
    29  166  -42  -68  -62  -48]
 [  89   67   87 -136  -28  101   40   82   60   67 -160   49   24  -51
   -18   13   74   16  115   94]
 [  10 -113   20  -13  138    3   87  -63  151  -49    8   56  -70   78
    -4   62   76  -31   84   14]
 [ 171   43  138  110   49  -27  -37  -12   82    7  -20  -85   60  108
   159 -118   55 -108   51  102]
 [ 107 -101  -87  -21    8 -143  100  -60  -10 -126   36  -75  -55 -166
    46  -71  -38  123 -131  -35]
 [  16  -89 -102  -78   54 -138  -38  -38  104   14   -1  -99 -122  -24
    31  -98  -26 -173  102   56]
 [ -16   52    4   36  -11  157   83  -50 -109   19  -15    0  129  -41
     4   55   94   18  -35  -80]
 [-112   45  -40  -69  -93  -83   60  -31 -102   19  176  -17   54  -86
  -145   17 -126   67  -54  -81]
```

```
[ 109   27   38  -80  100   94 -133   88  -33  -90  -47  -32  109    2
   13   33  -19  -26   67  -99]
[ -10   11   71 -162   40  -26   18   96   81   92   -3   18    6  -44
 -105  131 -121   62  125  117]
[ -17   37   81  -11  -87   98 -119   12   98   20  144    1   82   77
 -130    5  -43  -58  120   58]
[  29   74   45  -13  -99  -76   24  -16  -25   41   89  -54   50  120
 -130  -56  -82   93  -78   89]]

[[  21  -72   96    3    7   92  -56   43  133   82  -25   56  -17  140
 -114  128  -23   18  106   61]
[ -41  -25   95  -48  -51   17 -124  -33  112    3   12 -150    9   64
  -87  -83    3  -23   33  121]
[  62 -198  -26  -49  129  106  -69 -106  191   60  -89  -21  -90  -32
   21   67  117 -130   27   60]
[ 107  159  -55  -23   -3  131 -144  -59  -81   42  116  -42   63 -126
   20   93   -2  -41 -184   46]
[  84  113   37  -30  -38  -75   23   39 -124  -11   -2  -38   82 -118
  -22   75 -169  133  -30  -61]
[ 104   49  -29  -87  -23   -9   84 -113    5   63   -2  -24   47 -106
  -54    2   34    6  -74   95]
[ -80  -14  -74   40   34   32  -35  -67  112  -19   -8   32  -84  -81
   92  112 -136   61   -5  116]
[-120  -79  -28  -14  -22  -49  160 -138  148  119  114   55 -140  -28
   26   10   -2   25  -17  147]
[  28 -162  -19   40 -159   -6   22   53  118   -3   50   66  -45  -62
  -29 -166   42   68   62   48]
[ -89  -67  -87  136   28 -101  -40  -82  -60  -67  160  -49  -24   51
   18  -13  -74  -16 -115  -94]
[ -10  113  -20   13 -138   -3  -87   63 -151   49   -8  -56   70  -78
    4  -62  -76   31  -84  -14]
[-171  -43 -138 -110  -49   27   37   12  -82   -7   20   85  -60 -108
 -159  118  -55  108  -51 -102]
[-107  101   87   21   -8  143 -100   60   10  126  -36   75   55  166
  -46   71   38 -123  131   35]
[ -16   89  102   78  -54  138   38   38 -104  -14    1   99  122   24
  -31   98   26  173 -102  -56]
[  16  -52   -4  -36   11 -157  -83   50  109  -19   15    0 -129   41
   -4  -55  -94  -18   35   80]
[ 112  -45   40   69   93   83  -60   31  102  -19 -176   17  -54   86
  145  -17  126  -67   54   81]
[-109  -27  -38   80 -100  -94  133  -88   33   90   47   32 -109   -2
  -13  -33   19   26  -67   99]
[  10  -11  -71  162  -40   26  -18  -96  -81  -92    3  -18   -6   44
  105 -131  121  -62 -125 -117]
[  17  -37  -81   11   87  -98  119  -12  -98  -20 -144   -1  -82  -77
  130   -5   43   58 -120  -58]
[ -29  -74  -45   13   99   76  -24   16   25  -41  -89   54  -50 -120
```

```
   130   56   82  -93   78  -89]]
```

6 Question 2.e : Matrix Multiplication

Doesnt follow commutative property

[6]:
```
mul_1 = np.matmul(a, b)
mul_2 = np.matmul(b, a)
print(mul_1)
print()
print(mul_2)
```

```
[[143813 122882 156274 195079 166273 138753 145105 105801 165670 141963
  174381 163024 122882 156274 195079 166273 138753 145105 105801 165670]
 [175004 151996 181744 212424 204611 173870 187347 105661 194944 158156
  205933 211300 151996 181744 212424 204611 173870 187347 105661 194944]
 [192652 166686 176457 221124 190863 221722 161229 120133 193960 190903
  215696 202872 166686 176457 221124 190863 221722 161229 120133 193960]
 [210139 167376 177794 277225 229285 204265 222379 135089 222153 216662
  250853 261514 167376 177794 277225 229285 204265 222379 135089 222153]
 [218104 189480 189248 292795 245909 198285 238505 136141 218012 207256
  263351 270034 189480 189248 292795 245909 198285 238505 136141 218012]
 [213073 168535 183397 234745 189337 206024 181215 119998 198644 186718
  213686 203722 168535 183397 234745 189337 206024 181215 119998 198644]
 [212491 191838 210050 263758 238002 223311 212916 141153 225605 229956
  266631 264164 191838 210050 263758 238002 223311 212916 141153 225605]
 [194589 161621 189188 208931 196744 218484 160434 124032 201017 203761
  220012 189392 161621 189188 208931 196744 218484 160434 124032 201017]
 [236455 185714 219387 276133 239657 227832 206251 153903 239530 202515
  230456 229722 185714 219387 276133 239657 227832 206251 153903 239530]
 [270457 199754 260423 313312 297523 286837 251381 196070 309488 279902
  301541 296374 199754 260423 313312 297523 286837 251381 196070 309488]
 [208695 167747 187754 245189 224192 191544 210970 142830 208634 206286
  227899 243910 167747 187754 245189 224192 191544 210970 142830 208634]
 [209781 185522 209368 241727 236981 217282 185742 164151 212500 223574
  241113 226842 185522 209368 241727 236981 217282 185742 164151 212500]
 [164976 123851 153946 195685 149299 148599 147609 111302 169075 170936
  183686 174300 123851 153946 195685 149299 148599 147609 111302 169075]
 [199712 158586 170270 267287 211679 157088 210336 140188 194771 183172
  219440 232117 158586 170270 267287 211679 157088 210336 140188 194771]
 [248978 193842 205623 260420 221190 247878 174562 152678 229023 212931
  226837 227157 193842 205623 260420 221190 247878 174562 152678 229023]
 [183402 149850 147056 186703 162541 196007 139741 110676 166566 173611
  188429 190871 149850 147056 186703 162541 196007 139741 110676 166566]
 [213947 171368 206999 220335 224206 212173 185889 148332 215699 215300
  237693 209081 171368 206999 220335 224206 212173 185889 148332 215699]
 [278300 205338 234812 294289 262351 278136 224677 190844 273673 273401
```

```
  283084 279276 205338 234812 294289 262351 278136 224677 190844 273673]
 [270238 216848 213043 265732 253047 294499 204087 175897 242994 265633
  271893 268643 216848 213043 265732 253047 294499 204087 175897 242994]
 [191267 160390 146474 230232 179659 204848 159371 116382 175529 192035
  211396 203617 160390 146474 230232 179659 204848 159371 116382 175529]]

[[260333 229796 220354 238006 251726 212044 223783 201727 202178 211798
  261973 237576 220354 238006 251726 212044 223783 201727 202178 211798]
 [178480 151016 144525 177437 187867 149495 151839 152482 133068 132246
  192530 152020 144525 177437 187867 149495 151839 152482 133068 132246]
 [214225 198584 183855 206216 237063 191354 190166 173948 176024 182416
  201284 185696 183855 206216 237063 191354 190166 173948 176024 182416]
 [217243 182666 204091 218971 254785 192478 232536 177013 180240 151947
  215970 199665 204091 218971 254785 192478 232536 177013 180240 151947]
 [218488 192740 234727 220935 226734 176906 242981 166682 180634 171094
  255028 226516 234727 220935 226734 176906 242981 166682 180634 171094]
 [208070 173979 180882 179811 212081 166413 213375 163309 149966 149148
  214580 187042 180882 179811 212081 166413 213375 163309 149966 149148]
 [247190 203441 236452 261030 258515 194384 255880 170418 207899 196201
  286266 229774 236452 261030 258515 194384 255880 170418 207899 196201]
 [242854 207015 207378 215993 222148 192239 232347 163751 199284 214729
  255088 213910 207378 215993 222148 192239 232347 163751 199284 214729]
 [264466 219577 242331 241643 256978 209343 210206 194238 207010 200532
  242182 206841 242331 241643 256978 209343 210206 194238 207010 200532]
 [237149 194687 222959 232703 247624 198839 248870 174849 200851 177465
  244464 231573 222959 232703 247624 198839 248870 174849 200851 177465]
 [172656 149990 176261 159298 192001 134370 178157 131290 134507 118060
  168706 147443 176261 159298 192001 134370 178157 131290 134507 118060]
 [191035 129805 158383 149742 169217 119265 165830 107793 140127 108819
  169518 184213 158383 149742 169217 119265 165830 107793 140127 108819]
 [279765 223563 232478 244732 263764 212544 213376 206767 212013 186579
  243954 242451 232478 244732 263764 212544 213376 206767 212013 186579]
 [304102 240018 286429 287017 313978 234017 285669 221307 242918 205555
  306049 283556 286429 287017 313978 234017 285669 221307 242918 205555]
 [203452 174347 165276 210686 198189 171361 198933 147800 176066 177707
  225296 187653 165276 210686 198189 171361 198933 147800 176066 177707]
 [238796 223795 229114 259828 275526 203922 217137 198807 194480 196471
  262952 201992 229114 259828 275526 203922 217137 198807 194480 196471]
 [210118 193260 208837 210820 216690 174865 218492 151285 181285 192701
  239114 193220 208837 210820 216690 174865 218492 151285 181285 192701]
 [243367 194695 229869 233910 279463 193597 239984 181499 199394 157213
  228710 207016 229869 233910 279463 193597 239984 181499 199394 157213]
 [214823 204908 228030 222065 240117 181760 238682 166551 181984 187321
  250086 209239 228030 222065 240117 181760 238682 166551 181984 187321]
 [192413 165418 206756 217086 209720 174277 163957 150038 181082 149975
  180340 182356 206756 217086 209720 174277 163957 150038 181082 149975]]
```

7 Question 2.f : Multiply with a scalar to both matrices A and B.

7.1 Question 2.f : Multiply with $C > 1$

[7]:
```
c1 = 2
a_scalar = a * c1
b_scalar = b * c1
print(a_scalar)
print()
print(b_scalar)
```

```
[[126 268 158  70 304  62 292 144  18 162 210  52 158  70 304  62 292 144
   18 162]
 [198 190 122 184 262 326 354 152  20   2  18 346 122 184 262 326 354 152
   20   2]
 [222 396 180 192  52 176 154 276  10 272 370 184 180 192  52 176 154 276
   10 272]
 [ 80  16 208 350 264  72 338 132 382 128  14 242 208 350 264  72 338 132
  382 128]
 [116  24  86 396 380 154 342 122 260 134 104 330  86 396 380 154 342 122
  260 134]
 [178 160 164 318 252 202 116 272 194  14 212  94 164 318 252 202 116 272
  194  14]
 [320 162 302 316 212  56 392 200  76  68 348 320 302 316 212  56 392 200
   76  68]
 [290 170 292 292 212 148  54 324  82  84 106  90 292 292 212 148  54 324
   82  84]
 [298 334 100 186 324 338 242 150 132 272 234  56 100 186 324 338 242 150
  132 272]
 [328 394 308  32 268 350 296 248 314 382  10 302 308  32 268 350 296 248
  314 382]
 [384  62 148 264 282 130 276  40 334  60 180 320 148 264 282 130 276  40
  334  60]
 [348 280 314 254 352  18 182  40 166 206 358 208 314 254 352  18 182  40
  166 206]
 [294 106 198  40 174  16 226 272 130  80 392  58 198  40 174  16 226 272
  130  80]
 [214  82  16 172 390  86 310  40 320 224 288 194  16 172 390  86 310  40
  320 224]
 [ 36 214 368 278 214 346 220  90 120 178 366  16 368 278 214 346 220  90
  120 178]
 [104 230 248 156   4 224 138 152 106 148 380 284 248 156   4 224 138 152
  106 148]
 [322 108 272 200 386 252  26 240 198  66  26 268 272 200 386 252  26 240
  198  66]
 [358 290 280  50 164 346 156 244 302 374 366 300 280  50 164 346 156 244
  302 374]
 [236 260 350 342 104 288   6 128 344 264 306 378 350 342 104 288   6 128
```

```
  344 264]
 [182 164 116 266  32 118 106 244 120 348 386 128 116 266  32 118 106 244
  120 348]]

[[168 124 350  76 318 246 180 230 284 326 160 164 124 350  76 318 246 180
  230 284]
 [116 140 312  88 160 360 106  86 244   8  42  46 140 312  88 160 360 106
   86 244]
 [346   0 128  94 310 388  16  64 392 392 192 142   0 128  94 310 388  16
   64 392]
 [294 334  98 304 258 334  50  14 220 212 246 158 334  98 304 258 334  50
   14 220]
 [284 250 160 336 304   4 388 200  12 112 100 254 250 160 336 304   4 388
  200  12]
 [386 258 106 144 206 184 284  46 204 140 208  46 258 106 144 206 184 284
   46 204]
 [160 134 154 396 280 120 322  66 300  30 332 384 134 154 396 280 120 322
   66 300]
 [ 50  12 236 264 168  50 374  48 378 322 334 200  12 236 264 168  50 374
   48 378]
 [354  10  62 266   6 326 286 256 368 266 334 188  10  62 266   6 326 286
  256 368]
 [150 260 134 304 324 148 216  84 194 248 330 204 260 134 304 324 148 216
   84 194]
 [364 288 108 290   6 124 102 166  32 158 164 208 288 108 290   6 124 102
  166  32]
 [  6 194  38  34 254  72 256  64   2 192 398 378 194  38  34 254  72 256
   64   2]
 [ 80 308 372  82 158 302  26 392 150 332 320 208 308 372  82 158 302  26
  392 150]
 [182 260 220 328 282 362 386 116 112 196 290 392 260 220 328 282 362 386
  116 112]
 [ 68 110 360 206 236  32  54 190 338 140 396  16 110 360 206 236  32  54
  190 338]
 [328 140 328 294 190 390  18 214 310 110  28 318 140 328 294 190 390  18
  214 310]
 [104  54 196 360 186  64 292  64 264 246 120 332  54 196 360 186  64 292
   64 264]
 [378 268 138 374  84 398 120  52 140 190 372 264 268 138 374  84 398 120
   52 140]
 [270 186 188 364 278  92 244 104 148 224  18 376 186 188 364 278  92 244
  104 148]
 [124  16  26 292 230 270  58 276 170 266 208 236  16  26 292 230 270  58
  276 170]]
```

7.2 Question 2.f : Multiply with $C < 1$

[8]:
```
c2 = 0.4
a_scalar = a * c2
b_scalar = b * c2
print(a_scalar)
print()
print(b_scalar)
```

```
[[25.2 53.6 31.6 14.  60.8 12.4 58.4 28.8  3.6 32.4 42.  10.4 31.6 14.
  60.8 12.4 58.4 28.8  3.6 32.4]
 [39.6 38.  24.4 36.8 52.4 65.2 70.8 30.4  4.   0.4  3.6 69.2 24.4 36.8
  52.4 65.2 70.8 30.4  4.   0.4]
 [44.4 79.2 36.  38.4 10.4 35.2 30.8 55.2  2.  54.4 74.  36.8 36.  38.4
  10.4 35.2 30.8 55.2  2.  54.4]
 [16.   3.2 41.6 70.  52.8 14.4 67.6 26.4 76.4 25.6  2.8 48.4 41.6 70.
  52.8 14.4 67.6 26.4 76.4 25.6]
 [23.2  4.8 17.2 79.2 76.  30.8 68.4 24.4 52.  26.8 20.8 66.  17.2 79.2
  76.  30.8 68.4 24.4 52.  26.8]
 [35.6 32.  32.8 63.6 50.4 40.4 23.2 54.4 38.8  2.8 42.4 18.8 32.8 63.6
  50.4 40.4 23.2 54.4 38.8  2.8]
 [64.  32.4 60.4 63.2 42.4 11.2 78.4 40.  15.2 13.6 69.6 64.  60.4 63.2
  42.4 11.2 78.4 40.  15.2 13.6]
 [58.  34.  58.4 58.4 42.4 29.6 10.8 64.8 16.4 16.8 21.2 18.  58.4 58.4
  42.4 29.6 10.8 64.8 16.4 16.8]
 [59.6 66.8 20.  37.2 64.8 67.6 48.4 30.  26.4 54.4 46.8 11.2 20.  37.2
  64.8 67.6 48.4 30.  26.4 54.4]
 [65.6 78.8 61.6  6.4 53.6 70.  59.2 49.6 62.8 76.4  2.  60.4 61.6  6.4
  53.6 70.  59.2 49.6 62.8 76.4]
 [76.8 12.4 29.6 52.8 56.4 26.  55.2  8.  66.8 12.  36.  64.  29.6 52.8
  56.4 26.  55.2  8.  66.8 12. ]
 [69.6 56.  62.8 50.8 70.4  3.6 36.4  8.  33.2 41.2 71.6 41.6 62.8 50.8
  70.4  3.6 36.4  8.  33.2 41.2]
 [58.8 21.2 39.6  8.  34.8  3.2 45.2 54.4 26.  16.  78.4 11.6 39.6  8.
  34.8  3.2 45.2 54.4 26.  16. ]
 [42.8 16.4  3.2 34.4 78.  17.2 62.   8.  64.  44.8 57.6 38.8  3.2 34.4
  78.  17.2 62.   8.  64.  44.8]
 [ 7.2 42.8 73.6 55.6 42.8 69.2 44.  18.  24.  35.6 73.2  3.2 73.6 55.6
  42.8 69.2 44.  18.  24.  35.6]
 [20.8 46.  49.6 31.2  0.8 44.8 27.6 30.4 21.2 29.6 76.  56.8 49.6 31.2
   0.8 44.8 27.6 30.4 21.2 29.6]
 [64.4 21.6 54.4 40.  77.2 50.4  5.2 48.  39.6 13.2  5.2 53.6 54.4 40.
  77.2 50.4  5.2 48.  39.6 13.2]
 [71.6 58.  56.  10.  32.8 69.2 31.2 48.8 60.4 74.8 73.2 60.  56.  10.
  32.8 69.2 31.2 48.8 60.4 74.8]
 [47.2 52.  70.  68.4 20.8 57.6  1.2 25.6 68.8 52.8 61.2 75.6 70.  68.4
  20.8 57.6  1.2 25.6 68.8 52.8]
 [36.4 32.8 23.2 53.2  6.4 23.6 21.2 48.8 24.  69.6 77.2 25.6 23.2 53.2
```

```
   6.4 23.6 21.2 48.8 24.  69.6]]

[[33.6 24.8 70.  15.2 63.6 49.2 36.  46.  56.8 65.2 32.  32.8 24.8 70.
  15.2 63.6 49.2 36.  46.  56.8]
 [23.2 28.  62.4 17.6 32.  72.  21.2 17.2 48.8  1.6  8.4  9.2 28.  62.4
  17.6 32.  72.  21.2 17.2 48.8]
 [69.2  0.  25.6 18.8 62.  77.6  3.2 12.8 78.4 78.4 38.4 28.4  0.  25.6
  18.8 62.  77.6  3.2 12.8 78.4]
 [58.8 66.8 19.6 60.8 51.6 66.8 10.   2.8 44.  42.4 49.2 31.6 66.8 19.6
  60.8 51.6 66.8 10.   2.8 44. ]
 [56.8 50.  32.  67.2 60.8  0.8 77.6 40.   2.4 22.4 20.  50.8 50.  32.
  67.2 60.8  0.8 77.6 40.   2.4]
 [77.2 51.6 21.2 28.8 41.2 36.8 56.8  9.2 40.8 28.  41.6  9.2 51.6 21.2
  28.8 41.2 36.8 56.8  9.2 40.8]
 [32.  26.8 30.8 79.2 56.  24.  64.4 13.2 60.   6.  66.4 76.8 26.8 30.8
  79.2 56.  24.  64.4 13.2 60. ]
 [10.   2.4 47.2 52.8 33.6 10.  74.8  9.6 75.6 64.4 66.8 40.   2.4 47.2
  52.8 33.6 10.  74.8  9.6 75.6]
 [70.8  2.  12.4 53.2  1.2 65.2 57.2 51.2 73.6 53.2 66.8 37.6  2.  12.4
  53.2  1.2 65.2 57.2 51.2 73.6]
 [30.  52.  26.8 60.8 64.8 29.6 43.2 16.8 38.8 49.6 66.  40.8 52.  26.8
  60.8 64.8 29.6 43.2 16.8 38.8]
 [72.8 57.6 21.6 58.   1.2 24.8 20.4 33.2  6.4 31.6 32.8 41.6 57.6 21.6
  58.   1.2 24.8 20.4 33.2  6.4]
 [ 1.2 38.8  7.6  6.8 50.8 14.4 51.2 12.8  0.4 38.4 79.6 75.6 38.8  7.6
   6.8 50.8 14.4 51.2 12.8  0.4]
 [16.  61.6 74.4 16.4 31.6 60.4  5.2 78.4 30.  66.4 64.  41.6 61.6 74.4
  16.4 31.6 60.4  5.2 78.4 30. ]
 [36.4 52.  44.  65.6 56.4 72.4 77.2 23.2 22.4 39.2 58.  78.4 52.  44.
  65.6 56.4 72.4 77.2 23.2 22.4]
 [13.6 22.  72.  41.2 47.2  6.4 10.8 38.  67.6 28.  79.2  3.2 22.  72.
  41.2 47.2  6.4 10.8 38.  67.6]
 [65.6 28.  65.6 58.8 38.  78.   3.6 42.8 62.  22.   5.6 63.6 28.  65.6
  58.8 38.  78.   3.6 42.8 62. ]
 [20.8 10.8 39.2 72.  37.2 12.8 58.4 12.8 52.8 49.2 24.  66.4 10.8 39.2
  72.  37.2 12.8 58.4 12.8 52.8]
 [75.6 53.6 27.6 74.8 16.8 79.6 24.  10.4 28.  38.  74.4 52.8 53.6 27.6
  74.8 16.8 79.6 24.  10.4 28. ]
 [54.  37.2 37.6 72.8 55.6 18.4 48.8 20.8 29.6 44.8  3.6 75.2 37.2 37.6
  72.8 55.6 18.4 48.8 20.8 29.6]
 [24.8  3.2  5.2 58.4 46.  54.  11.6 55.2 34.  53.2 41.6 47.2  3.2  5.2
  58.4 46.  54.  11.6 55.2 34. ]]
```

8 Question 2.g : Multiply with a scalar to both matrices A and B.

8.1 Question 2.g : Divide with $C < 1$

[9]:
```
c1 = 2
a_scalar = a / c1
b_scalar = b / c1
print(a_scalar)
print()
print(b_scalar)
```

```
[[31.5 67.  39.5 17.5 76.  15.5 73.  36.   4.5 40.5 52.5 13.  39.5 17.5
  76.  15.5 73.  36.   4.5 40.5]
 [49.5 47.5 30.5 46.  65.5 81.5 88.5 38.   5.   0.5  4.5 86.5 30.5 46.
  65.5 81.5 88.5 38.   5.   0.5]
 [55.5 99.  45.  48.  13.  44.  38.5 69.   2.5 68.  92.5 46.  45.  48.
  13.  44.  38.5 69.   2.5 68. ]
 [20.   4.  52.  87.5 66.  18.  84.5 33.  95.5 32.   3.5 60.5 52.  87.5
  66.  18.  84.5 33.  95.5 32. ]
 [29.   6.  21.5 99.  95.  38.5 85.5 30.5 65.  33.5 26.  82.5 21.5 99.
  95.  38.5 85.5 30.5 65.  33.5]
 [44.5 40.  41.  79.5 63.  50.5 29.  68.  48.5  3.5 53.  23.5 41.  79.5
  63.  50.5 29.  68.  48.5  3.5]
 [80.  40.5 75.5 79.  53.  14.  98.  50.  19.  17.  87.  80.  75.5 79.
  53.  14.  98.  50.  19.  17. ]
 [72.5 42.5 73.  73.  53.  37.  13.5 81.  20.5 21.  26.5 22.5 73.  73.
  53.  37.  13.5 81.  20.5 21. ]
 [74.5 83.5 25.  46.5 81.  84.5 60.5 37.5 33.  68.  58.5 14.  25.  46.5
  81.  84.5 60.5 37.5 33.  68. ]
 [82.  98.5 77.   8.  67.  87.5 74.  62.  78.5 95.5  2.5 75.5 77.   8.
  67.  87.5 74.  62.  78.5 95.5]
 [96.  15.5 37.  66.  70.5 32.5 69.  10.  83.5 15.  45.  80.  37.  66.
  70.5 32.5 69.  10.  83.5 15. ]
 [87.  70.  78.5 63.5 88.   4.5 45.5 10.  41.5 51.5 89.5 52.  78.5 63.5
  88.   4.5 45.5 10.  41.5 51.5]
 [73.5 26.5 49.5 10.  43.5  4.  56.5 68.  32.5 20.  98.  14.5 49.5 10.
  43.5  4.  56.5 68.  32.5 20. ]
 [53.5 20.5  4.  43.  97.5 21.5 77.5 10.  80.  56.  72.  48.5  4.  43.
  97.5 21.5 77.5 10.  80.  56. ]
 [ 9.  53.5 92.  69.5 53.5 86.5 55.  22.5 30.  44.5 91.5  4.  92.  69.5
  53.5 86.5 55.  22.5 30.  44.5]
 [26.  57.5 62.  39.   1.  56.  34.5 38.  26.5 37.  95.  71.  62.  39.
   1.  56.  34.5 38.  26.5 37. ]
 [80.5 27.  68.  50.  96.5 63.   6.5 60.  49.5 16.5  6.5 67.  68.  50.
  96.5 63.   6.5 60.  49.5 16.5]
 [89.5 72.5 70.  12.5 41.  86.5 39.  61.  75.5 93.5 91.5 75.  70.  12.5
  41.  86.5 39.  61.  75.5 93.5]
 [59.  65.  87.5 85.5 26.  72.   1.5 32.  86.  66.  76.5 94.5 87.5 85.5
```

```
  26.  72.   1.5 32.  86.  66. ]
 [45.5 41.  29.  66.5  8.  29.5 26.5 61.  30.  87.  96.5 32.  29.  66.5
   8.  29.5 26.5 61.  30.  87. ]]

[[42.  31.  87.5 19.  79.5 61.5 45.  57.5 71.  81.5 40.  41.  31.  87.5
  19.  79.5 61.5 45.  57.5 71. ]
 [29.  35.  78.  22.  40.  90.  26.5 21.5 61.   2.  10.5 11.5 35.  78.
  22.  40.  90.  26.5 21.5 61. ]
 [86.5  0.  32.  23.5 77.5 97.   4.  16.  98.  98.  48.  35.5  0.  32.
  23.5 77.5 97.   4.  16.  98. ]
 [73.5 83.5 24.5 76.  64.5 83.5 12.5  3.5 55.  53.  61.5 39.5 83.5 24.5
  76.  64.5 83.5 12.5  3.5 55. ]
 [71.  62.5 40.  84.  76.   1.  97.  50.   3.  28.  25.  63.5 62.5 40.
  84.  76.   1.  97.  50.   3. ]
 [96.5 64.5 26.5 36.  51.5 46.  71.  11.5 51.  35.  52.  11.5 64.5 26.5
  36.  51.5 46.  71.  11.5 51. ]
 [40.  33.5 38.5 99.  70.  30.  80.5 16.5 75.   7.5 83.  96.  33.5 38.5
  99.  70.  30.  80.5 16.5 75. ]
 [12.5  3.  59.  66.  42.  12.5 93.5 12.  94.5 80.5 83.5 50.   3.  59.
  66.  42.  12.5 93.5 12.  94.5]
 [88.5  2.5 15.5 66.5  1.5 81.5 71.5 64.  92.  66.5 83.5 47.   2.5 15.5
  66.5  1.5 81.5 71.5 64.  92. ]
 [37.5 65.  33.5 76.  81.  37.  54.  21.  48.5 62.  82.5 51.  65.  33.5
  76.  81.  37.  54.  21.  48.5]
 [91.  72.  27.  72.5  1.5 31.  25.5 41.5  8.  39.5 41.  52.  72.  27.
  72.5  1.5 31.  25.5 41.5  8. ]
 [ 1.5 48.5  9.5  8.5 63.5 18.  64.  16.   0.5 48.  99.5 94.5 48.5  9.5
   8.5 63.5 18.  64.  16.   0.5]
 [20.  77.  93.  20.5 39.5 75.5  6.5 98.  37.5 83.  80.  52.  77.  93.
  20.5 39.5 75.5  6.5 98.  37.5]
 [45.5 65.  55.  82.  70.5 90.5 96.5 29.  28.  49.  72.5 98.  65.  55.
  82.  70.5 90.5 96.5 29.  28. ]
 [17.  27.5 90.  51.5 59.   8.  13.5 47.5 84.5 35.  99.   4.  27.5 90.
  51.5 59.   8.  13.5 47.5 84.5]
 [82.  35.  82.  73.5 47.5 97.5  4.5 53.5 77.5 27.5  7.  79.5 35.  82.
  73.5 47.5 97.5  4.5 53.5 77.5]
 [26.  13.5 49.  90.  46.5 16.  73.  16.  66.  61.5 30.  83.  13.5 49.
  90.  46.5 16.  73.  16.  66. ]
 [94.5 67.  34.5 93.5 21.  99.5 30.  13.  35.  47.5 93.  66.  67.  34.5
  93.5 21.  99.5 30.  13.  35. ]
 [67.5 46.5 47.  91.  69.5 23.  61.  26.  37.  56.   4.5 94.  46.5 47.
  91.  69.5 23.  61.  26.  37. ]
 [31.   4.   6.5 73.  57.5 67.5 14.5 69.  42.5 66.5 52.  59.   4.   6.5
  73.  57.5 67.5 14.5 69.  42.5]]
```

8.2 Question 2.g : Divide with $C < 1$

[10]:
```
c2 = 0.4
a_scalar = a / c2
b_scalar = b / c2
print(a_scalar)
print()
print(b_scalar)
```

```
[[157.5 335.  197.5  87.5 380.   77.5 365.  180.   22.5 202.5 262.5  65.
  197.5  87.5 380.   77.5 365.  180.   22.5 202.5]
 [247.5 237.5 152.5 230.  327.5 407.5 442.5 190.   25.    2.5  22.5 432.5
  152.5 230.  327.5 407.5 442.5 190.   25.    2.5]
 [277.5 495.  225.  240.   65.  220.  192.5 345.   12.5 340.  462.5 230.
  225.  240.   65.  220.  192.5 345.   12.5 340. ]
 [100.   20.  260.  437.5 330.   90.  422.5 165.  477.5 160.   17.5 302.5
  260.  437.5 330.   90.  422.5 165.  477.5 160. ]
 [145.   30.  107.5 495.  475.  192.5 427.5 152.5 325.  167.5 130.  412.5
  107.5 495.  475.  192.5 427.5 152.5 325.  167.5]
 [222.5 200.  205.  397.5 315.  252.5 145.  340.  242.5  17.5 265.  117.5
  205.  397.5 315.  252.5 145.  340.  242.5  17.5]
 [400.  202.5 377.5 395.  265.   70.  490.  250.   95.   85.  435.  400.
  377.5 395.  265.   70.  490.  250.   95.   85. ]
 [362.5 212.5 365.  365.  265.  185.   67.5 405.  102.5 105.  132.5 112.5
  365.  365.  265.  185.   67.5 405.  102.5 105. ]
 [372.5 417.5 125.  232.5 405.  422.5 302.5 187.5 165.  340.  292.5  70.
  125.  232.5 405.  422.5 302.5 187.5 165.  340. ]
 [410.  492.5 385.   40.  335.  437.5 370.  310.  392.5 477.5  12.5 377.5
  385.   40.  335.  437.5 370.  310.  392.5 477.5]
 [480.   77.5 185.  330.  352.5 162.5 345.   50.  417.5  75.  225.  400.
  185.  330.  352.5 162.5 345.   50.  417.5  75. ]
 [435.  350.  392.5 317.5 440.   22.5 227.5  50.  207.5 257.5 447.5 260.
  392.5 317.5 440.   22.5 227.5  50.  207.5 257.5]
 [367.5 132.5 247.5  50.  217.5  20.  282.5 340.  162.5 100.  490.   72.5
  247.5  50.  217.5  20.  282.5 340.  162.5 100. ]
 [267.5 102.5  20.  215.  487.5 107.5 387.5  50.  400.  280.  360.  242.5
   20.  215.  487.5 107.5 387.5  50.  400.  280. ]
 [ 45.  267.5 460.  347.5 267.5 432.5 275.  112.5 150.  222.5 457.5  20.
  460.  347.5 267.5 432.5 275.  112.5 150.  222.5]
 [130.  287.5 310.  195.    5.  280.  172.5 190.  132.5 185.  475.  355.
  310.  195.    5.  280.  172.5 190.  132.5 185. ]
 [402.5 135.  340.  250.  482.5 315.   32.5 300.  247.5  82.5  32.5 335.
  340.  250.  482.5 315.   32.5 300.  247.5  82.5]
 [447.5 362.5 350.   62.5 205.  432.5 195.  305.  377.5 467.5 457.5 375.
  350.   62.5 205.  432.5 195.  305.  377.5 467.5]
 [295.  325.  437.5 427.5 130.  360.    7.5 160.  430.  330.  382.5 472.5
  437.5 427.5 130.  360.    7.5 160.  430.  330. ]
 [227.5 205.  145.  332.5  40.  147.5 132.5 305.  150.  435.  482.5 160.
```

```
  145.  332.5  40.  147.5 132.5 305.  150.  435. ]]

[[210.  155.  437.5  95.  397.5 307.5 225.  287.5 355.  407.5 200.  205.
  155.  437.5  95.  397.5 307.5 225.  287.5 355. ]
 [145.  175.  390.  110.  200.  450.  132.5 107.5 305.   10.   52.5  57.5
  175.  390.  110.  200.  450.  132.5 107.5 305. ]
 [432.5   0.  160.  117.5 387.5 485.   20.   80.  490.  490.  240.  177.5
    0.  160.  117.5 387.5 485.   20.   80.  490. ]
 [367.5 417.5 122.5 380.  322.5 417.5  62.5  17.5 275.  265.  307.5 197.5
  417.5 122.5 380.  322.5 417.5  62.5  17.5 275. ]
 [355.  312.5 200.  420.  380.    5.  485.  250.   15.  140.  125.  317.5
  312.5 200.  420.  380.    5.  485.  250.   15. ]
 [482.5 322.5 132.5 180.  257.5 230.  355.   57.5 255.  175.  260.   57.5
  322.5 132.5 180.  257.5 230.  355.   57.5 255. ]
 [200.  167.5 192.5 495.  350.  150.  402.5  82.5 375.   37.5 415.  480.
  167.5 192.5 495.  350.  150.  402.5  82.5 375. ]
 [ 62.5  15.  295.  330.  210.   62.5 467.5  60.  472.5 402.5 417.5 250.
   15.  295.  330.  210.   62.5 467.5  60.  472.5]
 [442.5  12.5  77.5 332.5   7.5 407.5 357.5 320.  460.  332.5 417.5 235.
   12.5  77.5 332.5   7.5 407.5 357.5 320.  460. ]
 [187.5 325.  167.5 380.  405.  185.  270.  105.  242.5 310.  412.5 255.
  325.  167.5 380.  405.  185.  270.  105.  242.5]
 [455.  360.  135.  362.5   7.5 155.  127.5 207.5  40.  197.5 205.  260.
  360.  135.  362.5   7.5 155.  127.5 207.5  40. ]
 [  7.5 242.5  47.5  42.5 317.5  90.  320.   80.    2.5 240.  497.5 472.5
  242.5  47.5  42.5 317.5  90.  320.   80.    2.5]
 [100.  385.  465.  102.5 197.5 377.5  32.5 490.  187.5 415.  400.  260.
  385.  465.  102.5 197.5 377.5  32.5 490.  187.5]
 [227.5 325.  275.  410.  352.5 452.5 482.5 145.  140.  245.  362.5 490.
  325.  275.  410.  352.5 452.5 482.5 145.  140. ]
 [ 85.  137.5 450.  257.5 295.   40.   67.5 237.5 422.5 175.  495.   20.
  137.5 450.  257.5 295.   40.   67.5 237.5 422.5]
 [410.  175.  410.  367.5 237.5 487.5  22.5 267.5 387.5 137.5  35.  397.5
  175.  410.  367.5 237.5 487.5  22.5 267.5 387.5]
 [130.   67.5 245.  450.  232.5  80.  365.   80.  330.  307.5 150.  415.
   67.5 245.  450.  232.5  80.  365.   80.  330. ]
 [472.5 335.  172.5 467.5 105.  497.5 150.   65.  175.  237.5 465.  330.
  335.  172.5 467.5 105.  497.5 150.   65.  175. ]
 [337.5 232.5 235.  455.  347.5 115.  305.  130.  185.  280.   22.5 470.
  232.5 235.  455.  347.5 115.  305.  130.  185. ]
 [155.   20.   32.5 365.  287.5 337.5  72.5 345.  212.5 332.5 260.  295.
   20.   32.5 365.  287.5 337.5  72.5 345.  212.5]]
```

9 Question 2.h : Element by element multiplication

Elementwise multiplication is commutative

[11]:
```
elementwise_1 = np.multiply(a,b)
elementwise_2 = np.multiply(b,a)
print(elementwise_1)
print()
print(elementwise_2)
```

```
[[ 5292  8308 13825  1330 24168  3813 13140  8280  1278 13203  8400  2132
   4898  6125  5776  4929 17958  6480  1035 11502]
 [ 5742  6650  9516  4048 10480 29340  9381  3268  1220     4   189  3979
   4270 14352  5764 13040 31860  4028   430   122]
 [19203     0  5760  4512  4030 17072   616  4416   980 26656 17760  6532
      0  6144  1222 13640 14938  1104   160 26656]
 [ 5880  1336  5096 26600 17028  6012  4225   462 21010  6784   861  9559
  17368  8575 20064  4644 28223  1650  1337  7040]
 [ 8236  1500  3440 33264 28880   154 33174  6100   780  3752  2600 20955
   5375 15840 31920 11704   342 11834 13000   402]
 [17177 10320  4346 11448 12978  9292  8236  3128  9894   490 11024  1081
  10578  8427  9072 10403  5336 19312  2231   714]
 [12800  5427 11627 31284 14840  1680 31556  3300  5700   510 28884 30720
  10117 12166 20988  3920 11760 16100  1254  5100]
 [ 3625   510 17228 19272  8904  1850  5049  3888  7749  6762  8851  4500
    876 17228 13992  6216   675 30294   984  7938]
 [26373   835  1550 12369   486 27547 17303  9600 12144 18088 19539  2632
    250  2883 21546   507 19723 10725  8448 25024]
 [12300 25610 10318  2432 21708 12950 15984  5208 15229 23684   825 15402
  20020  1072 20368 28350 10952 13392  6594 18527]
 [34944  4464  3996 19140   423  4030  7038  1660  2672  2370  7380 16640
  10656  7128 20445   195  8556  1020 13861   480]
 [  522 13580  2983  2159 22352   324 11648   640    83  9888 35621 19656
  15229  2413  2992  1143  3276  2560  2656   103]
 [ 5880  8162 18414   820  6873  1208  1469 26656  4875  6640 31360  3016
  15246  3720  3567   632 17063  1768 12740  3000]
 [ 9737  5330   880 14104 27495  7783 29915  1160  8960 10976 20880 19012
   1040  9460 31980  6063 28055  3860  9280  6272]
 [  612  5885 33120 14317 12626  2768  2970  4275 10140  6230 36234    64
  10120 25020 11021 20414  1760  1215  5700 15041]
 [ 8528  8050 20336 11466   190 21840   621  8132  8215  4070  2660 22578
   8680 12792   294 10640 13455   684  5671 11470]
 [ 8372  1458 13328 18000 17949  4032  1898  3840 13068  4059   780 22244
   3672  9800 34740 11718   416 17520  3168  4356]
 [33831 19430  9660  4675  3444 34427  4680  3172 10570 17765 34038 19800
  18760  1725 15334  7266 15522  7320  3926 13090]
 [15930 12090 16450 31122  7228  6624   366  3328 12728 14784  1377 35532
  16275 16074  9464 20016   138  7808  8944  9768]
 [ 5642   656   754 19418  1840  7965  1537 16836  5100 23142 20072  7552
    464  1729  2336  6785  7155  3538  8280 14790]]
```

```
[[ 5292  8308 13825  1330 24168  3813 13140  8280  1278 13203  8400  2132
   4898  6125  5776  4929 17958  6480  1035 11502]
 [ 5742  6650  9516  4048 10480 29340  9381  3268  1220     4   189  3979
   4270 14352  5764 13040 31860  4028   430   122]
 [19203     0  5760  4512  4030 17072   616  4416   980 26656 17760  6532
      0  6144  1222 13640 14938  1104   160 26656]
 [ 5880  1336  5096 26600 17028  6012  4225   462 21010  6784   861  9559
  17368  8575 20064  4644 28223  1650  1337  7040]
 [ 8236  1500  3440 33264 28880   154 33174  6100   780  3752  2600 20955
   5375 15840 31920 11704   342 11834 13000   402]
 [17177 10320  4346 11448 12978  9292  8236  3128  9894   490 11024  1081
  10578  8427  9072 10403  5336 19312  2231   714]
 [12800  5427 11627 31284 14840  1680 31556  3300  5700   510 28884 30720
  10117 12166 20988  3920 11760 16100  1254  5100]
 [ 3625   510 17228 19272  8904  1850  5049  3888  7749  6762  8851  4500
    876 17228 13992  6216   675 30294   984  7938]
 [26373   835  1550 12369   486 27547 17303  9600 12144 18088 19539  2632
    250  2883 21546   507 19723 10725  8448 25024]
 [12300 25610 10318  2432 21708 12950 15984  5208 15229 23684   825 15402
  20020  1072 20368 28350 10952 13392  6594 18527]
 [34944  4464  3996 19140   423  4030  7038  1660  2672  2370  7380 16640
  10656  7128 20445   195  8556  1020 13861   480]
 [  522 13580  2983  2159 22352   324 11648   640    83  9888 35621 19656
  15229  2413  2992  1143  3276  2560  2656   103]
 [ 5880  8162 18414   820  6873  1208  1469 26656  4875  6640 31360  3016
  15246  3720  3567   632 17063  1768 12740  3000]
 [ 9737  5330   880 14104 27495  7783 29915  1160  8960 10976 20880 19012
   1040  9460 31980  6063 28055  3860  9280  6272]
 [  612  5885 33120 14317 12626  2768  2970  4275 10140  6230 36234    64
  10120 25020 11021 20414  1760  1215  5700 15041]
 [ 8528  8050 20336 11466   190 21840   621  8132  8215  4070  2660 22578
   8680 12792   294 10640 13455   684  5671 11470]
 [ 8372  1458 13328 18000 17949  4032  1898  3840 13068  4059   780 22244
   3672  9800 34740 11718   416 17520  3168  4356]
 [33831 19430  9660  4675  3444 34427  4680  3172 10570 17765 34038 19800
  18760  1725 15334  7266 15522  7320  3926 13090]
 [15930 12090 16450 31122  7228  6624   366  3328 12728 14784  1377 35532
  16275 16074  9464 20016   138  7808  8944  9768]
 [ 5642   656   754 19418  1840  7965  1537 16836  5100 23142 20072  7552
    464  1729  2336  6785  7155  3538  8280 14790]]
```

10 Question 2.i : Find out the location(s) of a specific values

[12]:
```
key = 134

x1,y1 = np.where(a == key)
```

```
a_copy = a.copy()
b_copy = b.copy()
for c,(i,j )in enumerate(zip(x1,y1)):
    print(f'In matrix a, found {key} at : [{i+1},{j+1}]' )

print()
x2,y2 = np.where(b == key)

for c,(i,j) in enumerate(zip(x2,y2)):
    b_copy[i][j] = replace_by
    print(f'In matrix b, found {key} at : [{i+1},{j+1}]' )
```

```
In matrix a, found 134 at : [1,2]
In matrix a, found 134 at : [10,5]
In matrix a, found 134 at : [10,15]
In matrix a, found 134 at : [17,12]
```

```
---------------------------------------------------------------------------

NameError                                 Traceback (most recent call last)

<ipython-input-12-1984f6607297> in <module>
 11
 12 for c,(i,j) in enumerate(zip(x2,y2)):
---> 13     b_copy[i][j] = replace_by
 14     print(f'In matrix b, found {key} at : [{i+1},{j+1}]' )


NameError: name 'replace_by' is not defined
```

11 Question 2.j : Find the specific value of X (only first occurrence) using the scan and search mechanism and amplify the value by a factor of 2

[ ]:
```
key = 134
x1,y1 = np.where(a == key)
a_copy = a.copy()
b_copy = b.copy()
for c,(i,j )in enumerate(zip(x1,y1)):
    a_copy[i][j] = a_copy[i][j] * 2
    print(f'In matrix a, replaced {key} at : [{i+1},{j+1}]' )
print(f'In matrix a, {key} found {c} times \n' )
```

```
x2,y2 = np.where(b == key)

for c,(i,j) in enumerate(zip(x2,y2)):
    b_copy[i][j] = b_copy[i][j] * 2
    print(f'In matrix b, replaced {key} at : [{i+1},{j+1}]' )
print(f'In matrix b, {key} found {c} times \n' )

print(a_copy)
print()
print(b_copy)
```

12 Question 2.k : Find the specific value of X and replace it with birthday. Count number of occurence as well

[ ]:
```
key = 134
replace_by = 709
x1,y1 = np.where(a == key)
a_copy = a.copy()
b_copy = b.copy()
for c,(i,j )in enumerate(zip(x1,y1)):
    a_copy[i][j] = replace_by
    print(f'In matrix a, replaced {key} at : [{i+1},{j+1}]' )
print(f'In matrix a, {key} found {c} times \n' )

x2,y2 = np.where(b == key)

for c,(i,j) in enumerate(zip(x2,y2)):
    b_copy[i][j] = replace_by
    print(f'In matrix b, replaced {key} at : [{i+1},{j+1}]' )
print(f'In matrix b, {key} found {c} times \n' )

print(a_copy)
print()
print(b_copy)
```