Assignment 2.2: Distance Metrics

1. Eucledian Distance
2. City Block Distance
3. Chess Board Distance

In [11]:

```
import math
import numpy as np
```

In [12]:

```
# Calculate the Eucledian distance between two points
def eucledianDistance(image, center_x, center_y):
    h = len(image)
    w = len(image[0])
    euclidDist = np.zeros([image_height, image_width], dtype=float)

    for i in range(h):
        for j in range(w):
            euclidDist[i][j] = round(math.sqrt((i-center_x)**2 + (j-center_y)**2), 2
    return euclidDist
```

In [13]:

```
# Calculate the City Block distance between two points
def cityBlockDistance(image, center_x, center_y):
    h = len(image)
    w = len(image[0])
    cityBlockDist = np.zeros([image_height, image_width], dtype=int)

    for i in range(h):
        for j in range(w):
            cityBlockDist[i][j] = abs(i-center_x) + abs(j-center_y)
    return cityBlockDist
```

In [14]:

```
# Calculate the Chess Board distance between two points
def chessBoardDistance(image, center_x, center_y):
    h = len(image)
    w = len(image[0])
    chessBoardDist = np.zeros([image_height, image_width], dtype=int)

    for i in range(h):
        for j in range(w):
            chessBoardDist[i][j] = max(abs(i-center_x), abs(j-center_y))
    return chessBoardDist
```

In [15]:

```
image_height = int(input())
image_width = int(input())
center_x = int(image_height/2)
center_y = int(image_width/2)
```

In [16]:

```
image = np.zeros([image_height, image_width], dtype=int)
image
```

Out[16]:

```
array([[0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0]])
```

In [22]:

```
euclidDist = eucledianDistance(image, center_x, center_y)
print("Eucledian Distance Tranformation\n{}".format(euclidDist))
```

```
Eucledian Distance Tranformation
[[2.83 2.24 2.   2.24 2.83]
 [2.24 1.41 1.   1.41 2.24]
 [2.   1.   0.   1.   2.  ]
 [2.24 1.41 1.   1.41 2.24]
 [2.83 2.24 2.   2.24 2.83]]
```

In [23]:

```
cityBlockDist = cityBlockDistance(image, center_x, center_y)
print("City Block Distance Tranformation\n{}".format(cityBlockDist))
```

```
City Block Distance Tranformation
[[4 3 2 3 4]
 [3 2 1 2 3]
 [2 1 0 1 2]
 [3 2 1 2 3]
 [4 3 2 3 4]]
```

In [24]:

```
chessBoardDist = chessBoardDistance(image, center_x, center_y)
print("Chess Board Distance Transformation\n{}".format(chessBoardDist))
```

```
Chess Board Distance Transformation
[[2 2 2 2 2]
 [2 1 1 1 2]
 [2 1 0 1 2]
 [2 1 1 1 2]
 [2 2 2 2 2]]
```

In [ ]: