# Assignment - 1: EE4708 ('exercise_basic.ipynb')

Raj Jain | CH17B066

1) Given the array split the array at the middle (if it is odd length consider next higher integer), add reverse the array.

Ex: if input is [12,10,5,6,52,36] output should be [6,52,36,12,10,5]

if input is [12,10,5,6,52,36,34] output should be [6,52,36,34,12,10,5]

In [3]:

```
import math
# Complete this function to get the desired result
def reverseatCenter(arr):
    arr2 = arr[math.floor(len(arr)/2):] + arr[:math.floor(len(arr)/2)]
    return arr2
```

In [4]:

```
# Print wroking examples
print(reverseatCenter([12,10,5,6,52,36]))
arr = [12,10,5,6,52,36,34]
reverseatCenter(arr)
```

```
[6, 52, 36, 12, 10, 5]
```

Out[4]:

```
[6, 52, 36, 34, 12, 10, 5]
```

2) Given a list of numbers, return a list where all adjacent duplicate elements have been removed.

Ex:

2, 2, 2, 3, 2 returns 2, 3, 2.

In [5]:

```
# Complete the function to get the desired result

def remove_adjacent(b):
    b2 = []
    for i in range(1,len(b)):
        if b[i]!=b[i-1]:
            b2.append(b[i])
    return [b[0]]+b2
```

In [6]:

```
print(remove_adjacent([2, 2, 2, 3, 2]))
print(remove_adjacent([2, 2, 2, 3, 2,2,2,2,5,5,5,5,6,6,2,3,3,3]))
```

```
[2, 3, 2]
[2, 3, 2, 5, 6, 2, 3]
```

3) given matrix of 7x7 full of ones, create a square with given side length (center same as original square(7x7)) replace ones with zeors at the edges for example,

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

After modification

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

In [7]:

```
import numpy as np
def matSquare(l):
    #l: given side length
    #A: matrix of ones of size (7,7)
    #A2: ouptut matrix with centre at (4,4), of length 'l'
    A=np.ones((7,7))
    A2=A
    A2[3-int((l-1)/2):3+int((l-1)/2),3-int((l-1)/2)]=0
    A2[3-int((l-1)/2):3+int((l-1)/2),3+int((l-1)/2)]=0
    A2[3-int((l-1)/2),3-int((l-1)/2):3+int((l-1)/2)]=0
    A2[3+int((l-1)/2),3-int((l-1)/2):3+int((l-1)/2)+1]=0
    return A2

# write programm to get the desired result
```

In [9]:

```
#Print working examples
print("Square with edge length of: 1\n",matSquare(1))
print("\nSquare with edge length of: 3\n",matSquare(3))
print("\nSquare with edge length of: 5\n",matSquare(5))
print("\nSquare with edge length of: 7\n",matSquare(7))
```

```
Square with edge length of: 1
 [[1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 0. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]]

Square with edge length of: 3
 [[1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 0. 0. 0. 1. 1.]
 [1. 1. 0. 1. 0. 1. 1.]
 [1. 1. 0. 0. 0. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]]

Square with edge length of: 5
 [[1. 1. 1. 1. 1. 1. 1.]
 [1. 0. 0. 0. 0. 0. 1.]
 [1. 0. 1. 1. 1. 0. 1.]
 [1. 0. 1. 1. 1. 0. 1.]
 [1. 0. 1. 1. 1. 0. 1.]
 [1. 0. 0. 0. 0. 0. 1.]
 [1. 1. 1. 1. 1. 1. 1.]]

Square with edge length of: 7
 [[0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 1. 1. 1. 1. 0.]
 [0. 1. 1. 1. 1. 1. 0.]
 [0. 1. 1. 1. 1. 1. 0.]
 [0. 1. 1. 1. 1. 1. 0.]
 [0. 1. 1. 1. 1. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0.]]
```

4) Paragraph present in data.txt is encoded such that each alphabet in word is incremented to next ascii value. Decode the paragraph present in the data_encoded.txt (Hint: decrease the ascii value of each character in the word)

In [10]:

```
def sentence_decode(str):
    # write the code here
    dec_words = "".join([chr(i) for i in [ord(c)-1 for c in str]]).replace(chr(ord(" ")
-1)," ")
    return dec_words
```

In [11]:

```
with open ("data_encoded.txt", "r") as myfile:
    data=myfile.read()
    print(data)
    print(" ")
    print(sentence_decode(data))
```

```
ebub bobmztjt jt b qspdftt pg jotqfdujoh- dmfbotjoh- usbotgpsnjoh boe npef
mjoh ebub xjui uif hpbm pg ejtdpwfsjoh vtfgvm jogpsnbujpo- jogpsnjoh dpodm
vtjpot boe tvqqpsujoh efdjtjpo.nbljoh/ ebub bobmztjt ibt nvmujqmf gbdfut b
oe bqqspbdift- fodpnqbttjoh ejwfstf ufdiojrvft voefs b wbsjfuz pg obnft- b
oe jt vtfe jo ejggfsfou cvtjoftt- tdjfodf- boe tpdjbm tdjfodf epnbjot/ jo
upebz(t cvtjoftt xpsme- ebub bobmztjt qmbzt b spmf jo nbljoh efdjtjpot nps
f tdjfoujgjd boe ifmqjoh cvtjofttft pqfsbuf npsf fggfdujwfmz/

data analysis is a process of inspecting, cleansing, transforming and mode
ling data with the goal of discovering useful information, informing concl
usions and supporting decision-making. data analysis has multiple facets a
nd approaches, encompassing diverse techniques under a variety of names, a
nd is used in different business, science, and social science domains. in
today's business world, data analysis plays a role in making decisions mor
e scientific and helping businesses operate more effectively.
```

In [12]:

```
# Has to print the decoded paragraph.
```