# STRING ALGORITHMS

*Mahsa Ziraksima*
*ACIBADEM UNIVERSITY*
*Fall 2025*

# SUMMARY

- Anagram detection
- Palindrome detection
- Last digit
- Caesar cipher

# ANAGRAM DETECTION

- Two strings are anagrams if they contain the same letters, but not necessarily in the same order, case does not matter.

car and arc

- One way to determine is to sort them.

```
def is_anagram(s1, s2):
    s1 = s1.replace(' ','').lower()
    print(s1)
    s2 = s2.replace(' ','').lower()
    print(s2)
    if sorted(s1) == sorted(s2):
        return True
    else:
        return False

s1 = 'Emperor Octavian'
s2 = 'Captain over Rome'
print(is_anagram(s1,s2))
```

# PALINDROME DETECTION

- A palindrome is a word that reads the same backward as forward.

  Hannah, mom, wow, and racecar

- One way of solving this problem is to copy the string, reverse it, and compare it to the original.

```
def is_palindrome(s1):
    if s1.lower() == s1[::-1].lower():
        return True
    return False

s1 = 'wow'
print(is_palindrome(s1))
```

# LAST DIGIT

- The question is to return the rightmost digit in a string.

"Buy 1 get 2 free“ -> 2

- One elegant way to solve this problem is to use Python’s list comprehension feature.

```
s = "Buy 1 get 2 free"
nl =[c for c in s if c.isdigit()][- 1]
print(nl)
```

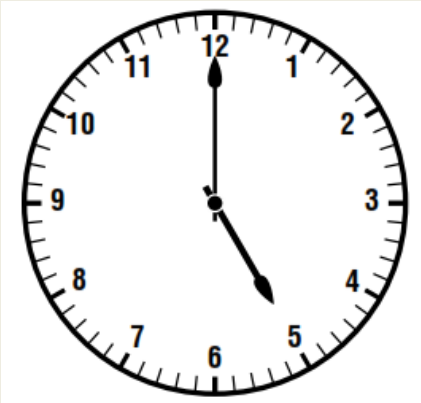
# CAESAR CIPHER

- A cipher is an algorithm for encryption or decryption.

  if the number is 3, the string abc would become def

  if the number is 2, the string z would become b

- The key in this problem is using modular arithmetic.



```
import string


def cipher(a_string, key):
    uppercase = string.ascii_uppercase
    lowercase = string.ascii_lowercase
    encrypt = ''
    for c in a_string:
        if c in uppercase:
            new = (uppercase.index(c) + key) % 26
            encrypt += uppercase[new]
        elif c in lowercase:
            new = (lowercase.index(c) + key) % 26
            encrypt += lowercase[new]
        else:
            encrypt += c
    return encrypt

s = 'Ab cD3Ef'
print(cipher(s,2))
```