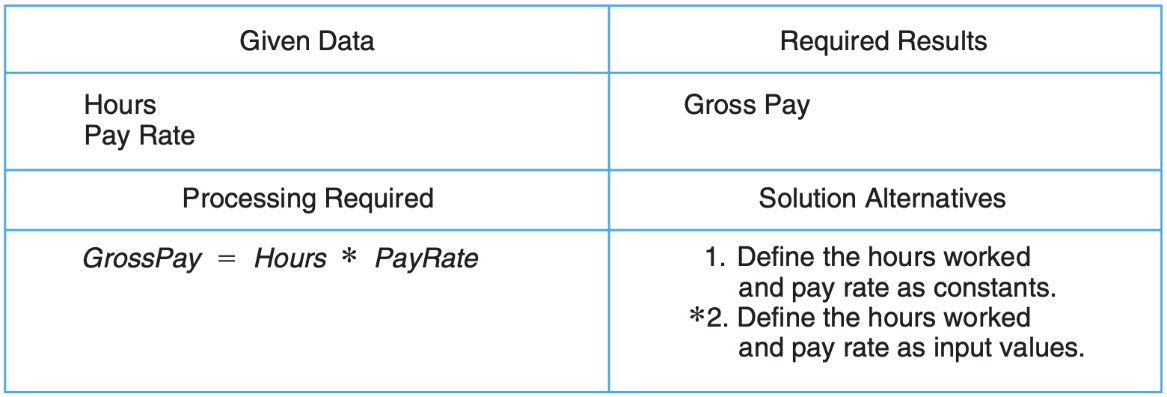
Problem Solving

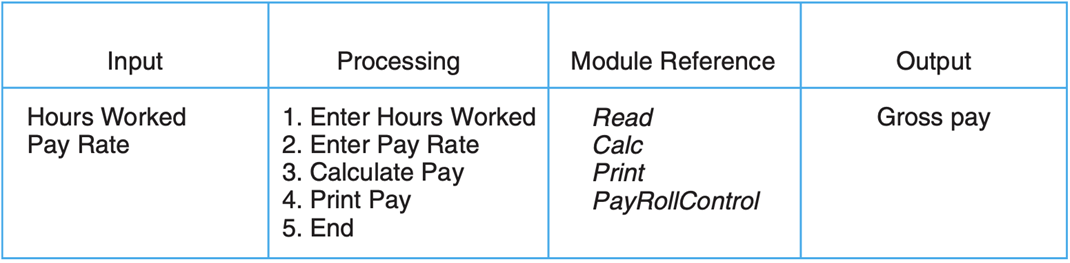
**Analysis Chart**

**Given data** : ข้อมูลที่โปรแกรมต้องได้รับ ค่า Input ต่าง ๆ ทั้งจากระบบ และจาก คีย์บอร์ด

**Required Results** : การกําหนดผลลัพธ์ที่ต้องการได้ดจากโปรแกรม

**Processing Required** : รายการของการประมวลผล สิ่งที่โปรแกรมต้องการใช้ในการประมวณผล เช่น สูตรคำนวณ สมการต่างๆ เงื่อนไข และการประมวลอย่างอื่น เช่น การเรียงลำดับ การค้นหา การแก้ไขข้อมูล

**Solution Alternatives** : วิธีการขั้นตอนในการแก้ปัญหา \*อธิบายวิธีในการแก้ปัญหา

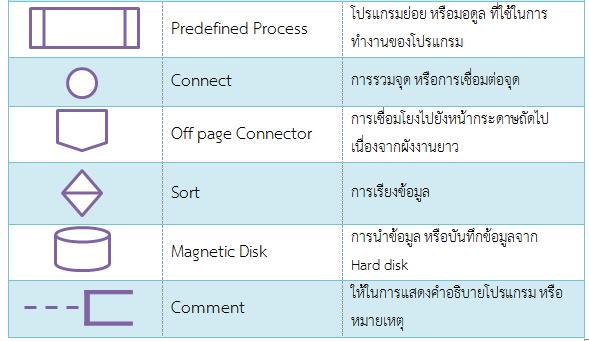
**IPO Chart**

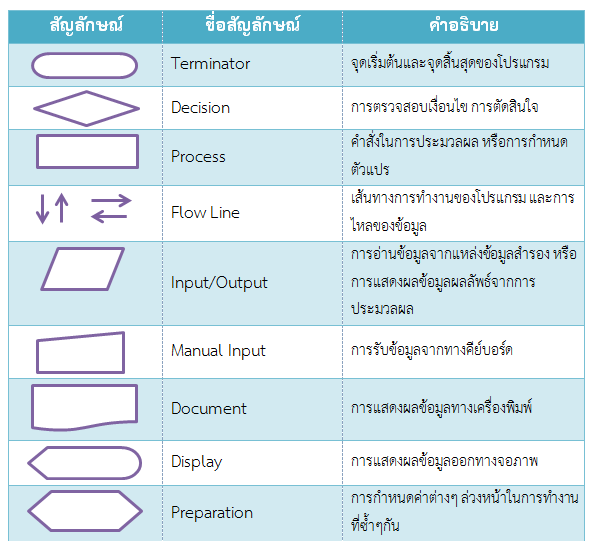
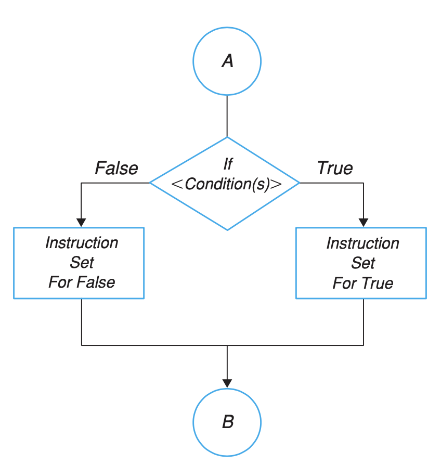
**การนําเข้าข้อมูล (Input)** : ข้อมูลที่โปรแกรมต้องได้รับ ที่จากผู้ใช้งาน และจากระบบ

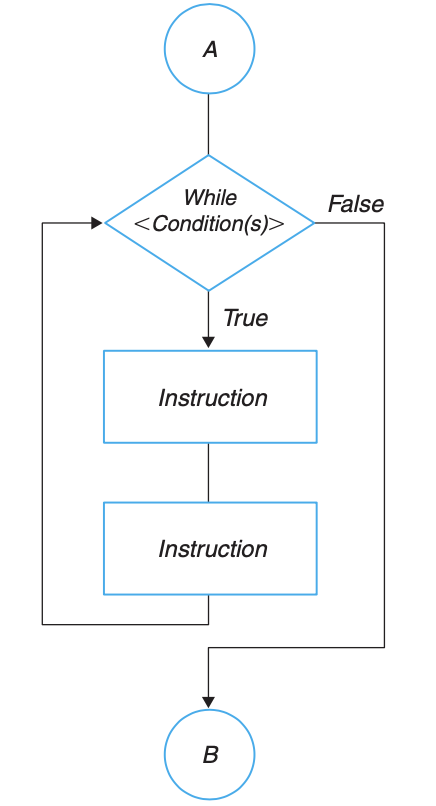
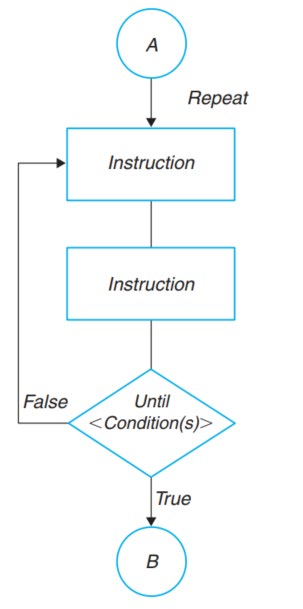
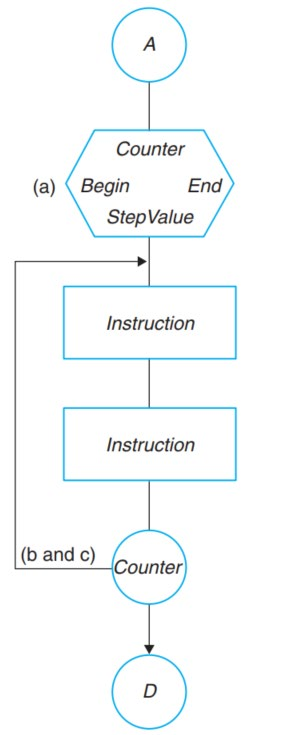
**การประมวลผล(Process)** : ขั้นตอนวิธีการทํางานในโปรแกรม ตั้งแต่ต้นจนจบ

**Module Reference** : เป็นการเขียนอ้างอิงโมดูลที่เกี่ยวข้อง เช่น Read Calc Print \*โมดูลคือการทำงานต่างๆ ของโปรแกรม

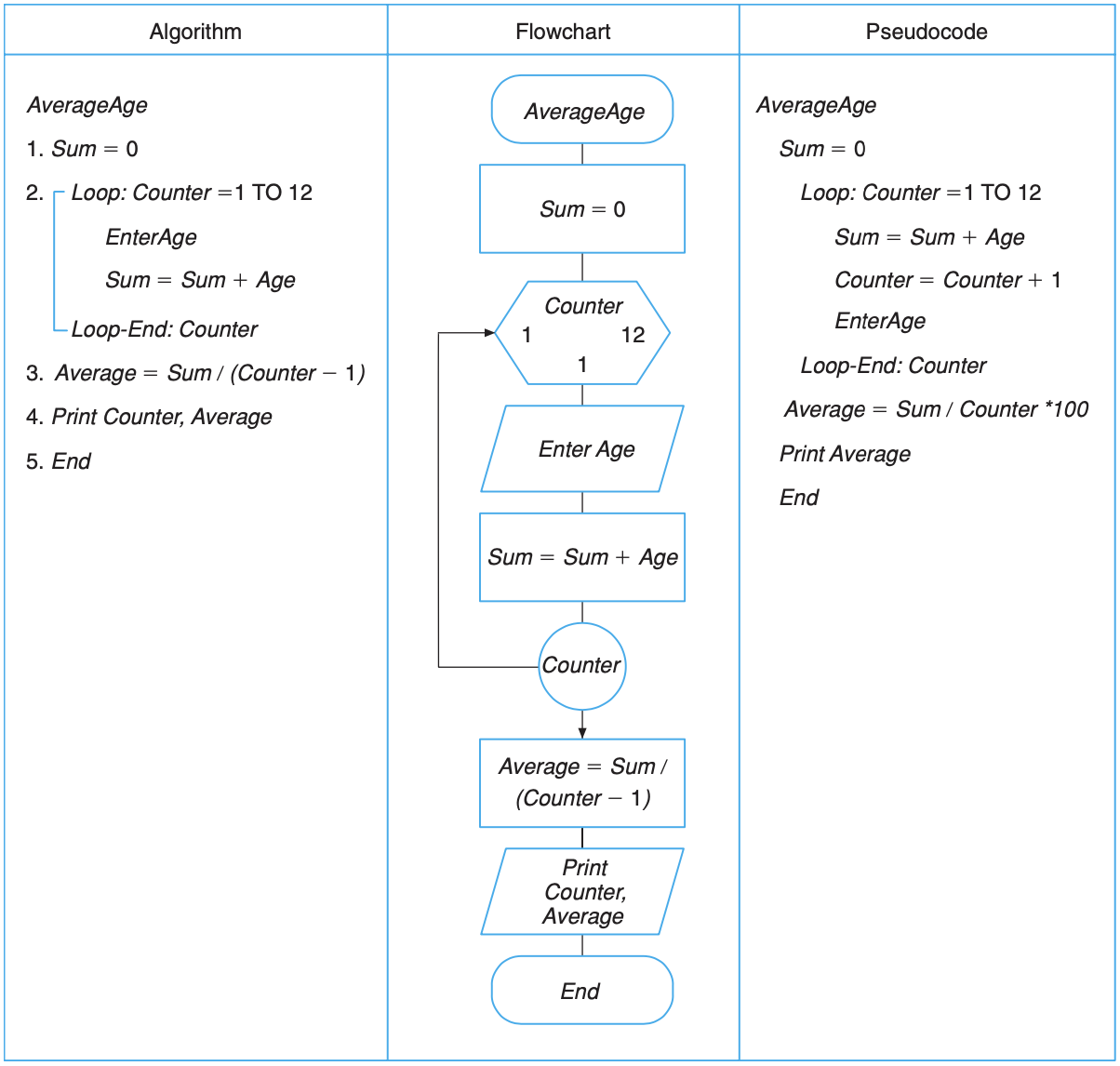
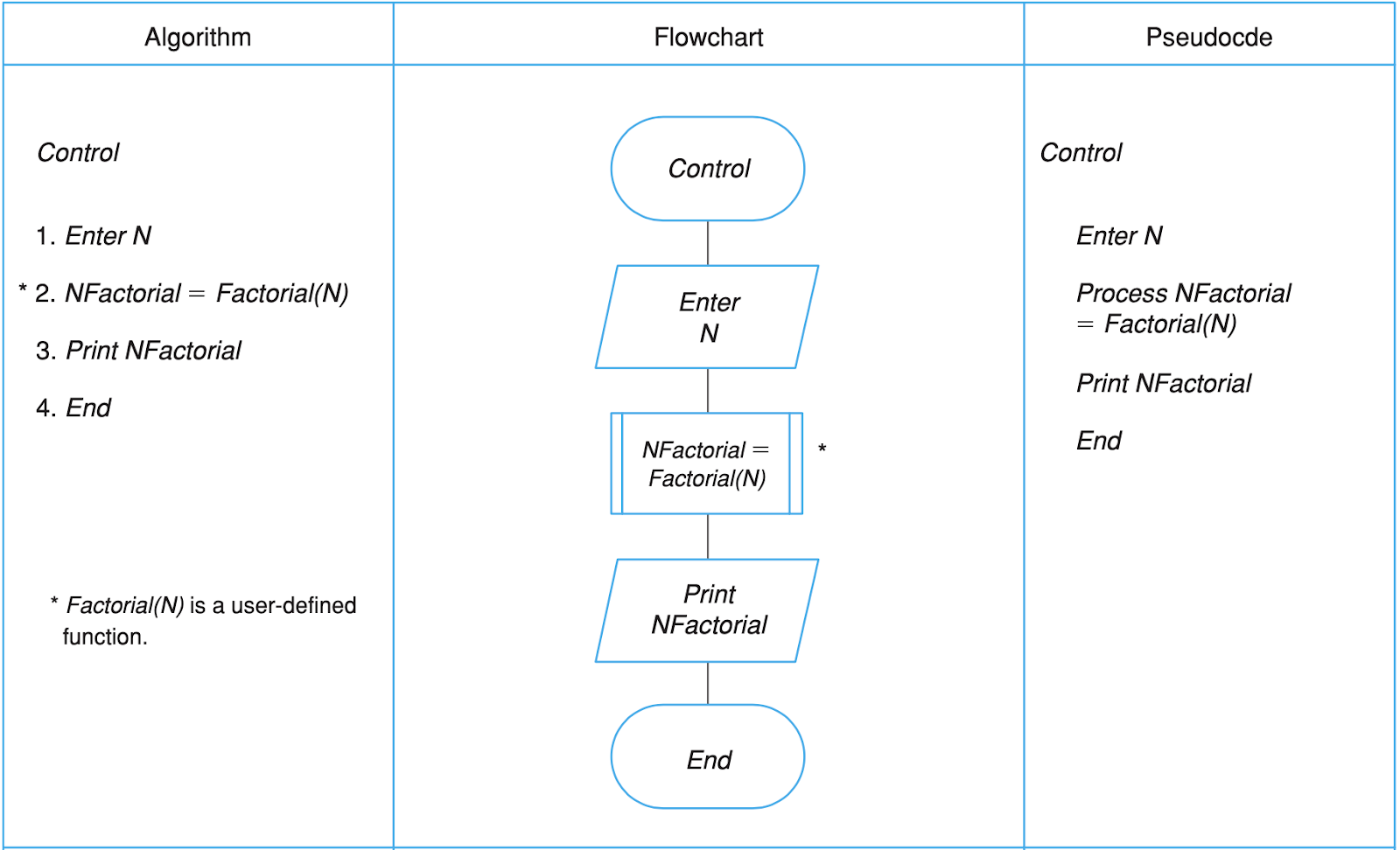
**การแสดงผล (Output)** : เป็นการกําหนดผลลัพธ์ที่ต้องการให้แสดงผลออกมา

**FlowChart**

แบบมีเงื่อนไข

Loop While Do While For Loop

ตัวอย่าง For Loop ตัวอย่างใช้ ฟังก์ชัน



numbers = [21, 34, 51, 23, 37, 44, 60, 11, 94, 99]

result = []

result2 = {}

for i in range(0,len(numbers)):

    group =[]

    for j in range(0,len(numbers)):

        if numbers[j]%10 == i:

            group.append(numbers[j])

    result.append(group)

    result2[i] = group ไม่เกี่ยวกับสไลด์เอาไปเฉยๆ

group\_most = [[]]

result3 = {}

for i in result2:

    if len(result2[i]) > len(group\_most[0]):

        group\_most.clear()

        group\_most.append(result2[i])

        result3.clear()

        result3[i] = result2[i]

    elif len(result2[i]) == len(group\_most[0]):

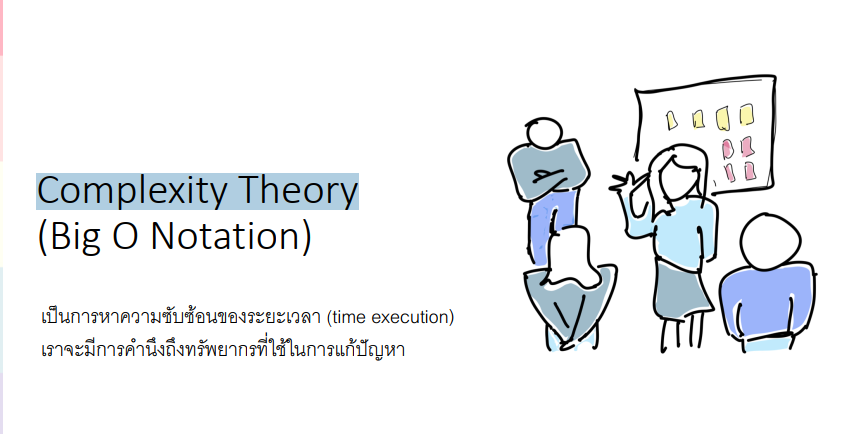
        group\_most.append(result2[i])

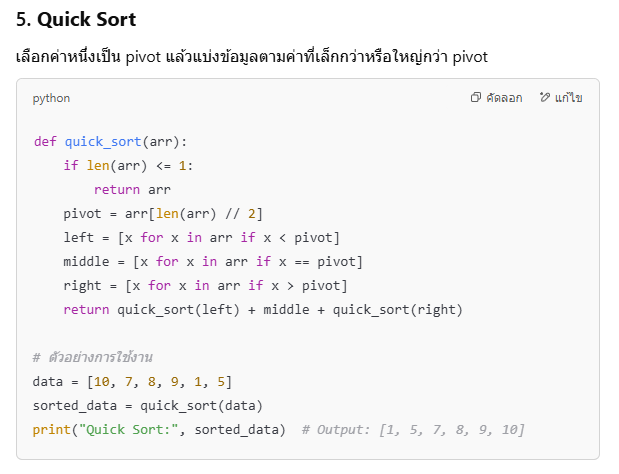
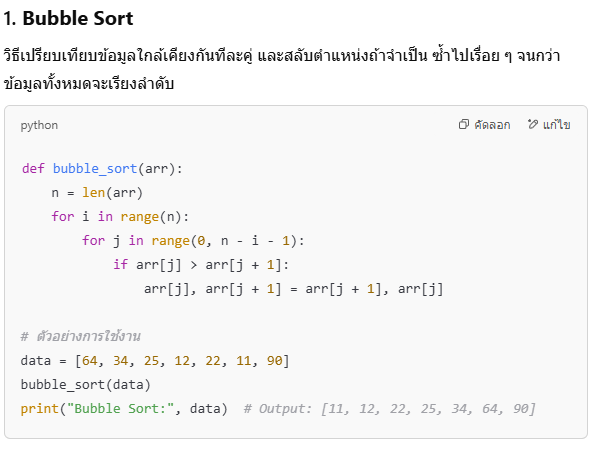
        result3[i] = result2[i]

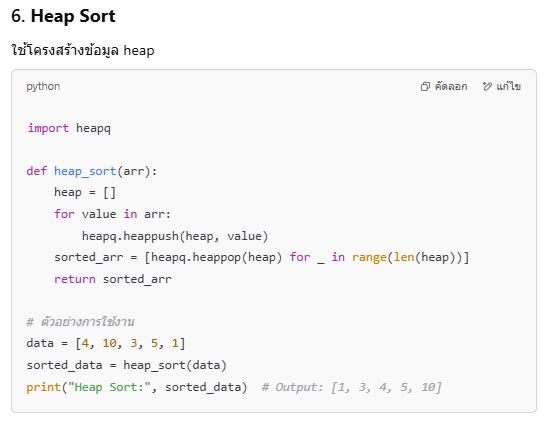
print(result2)

print(result3)

**Complexcity *\*\*อะไรบ้างที่ใช้ในการตรวจสอบว่า โค้ดนี้ดีแค่ไหน*** ให้ตอบตามสไลด์นี้

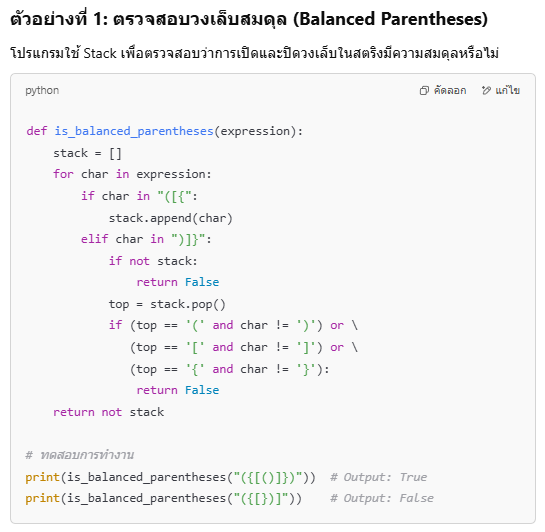


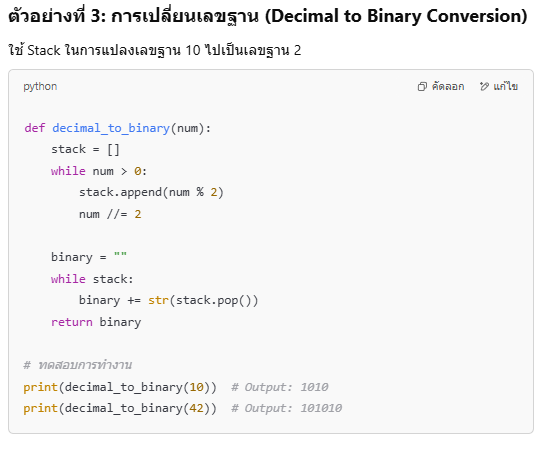
**Sort**การเรียงข้อมูลแบบต่างๆ

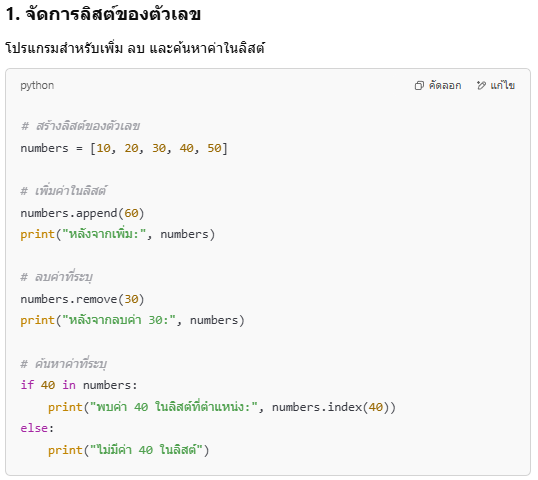
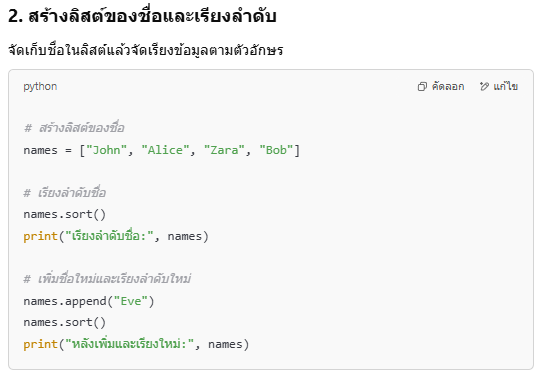
****

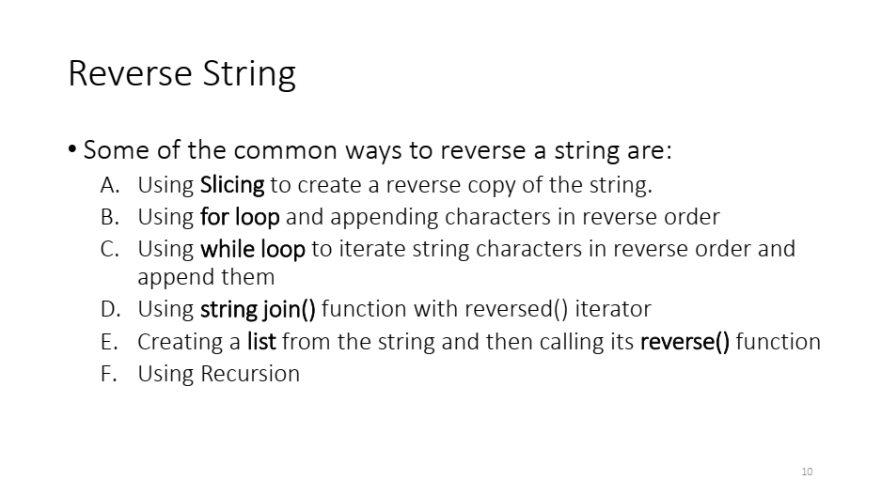
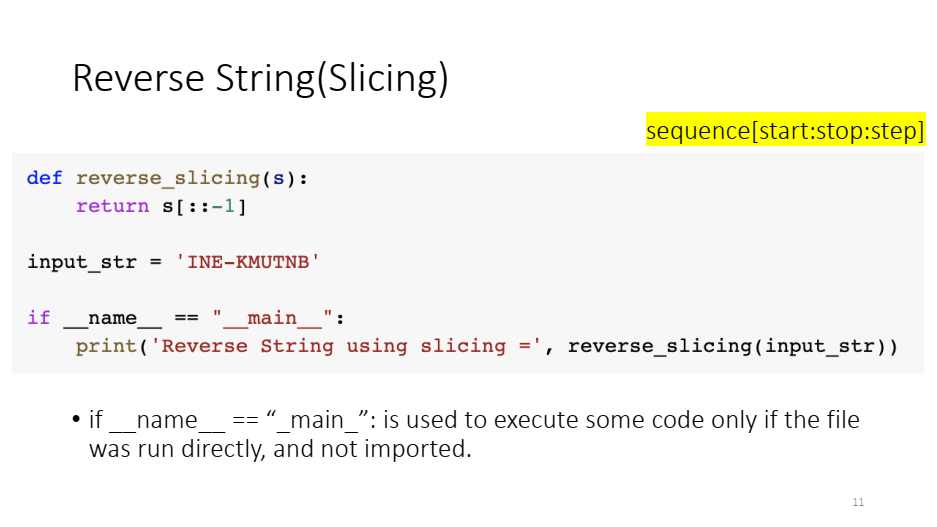
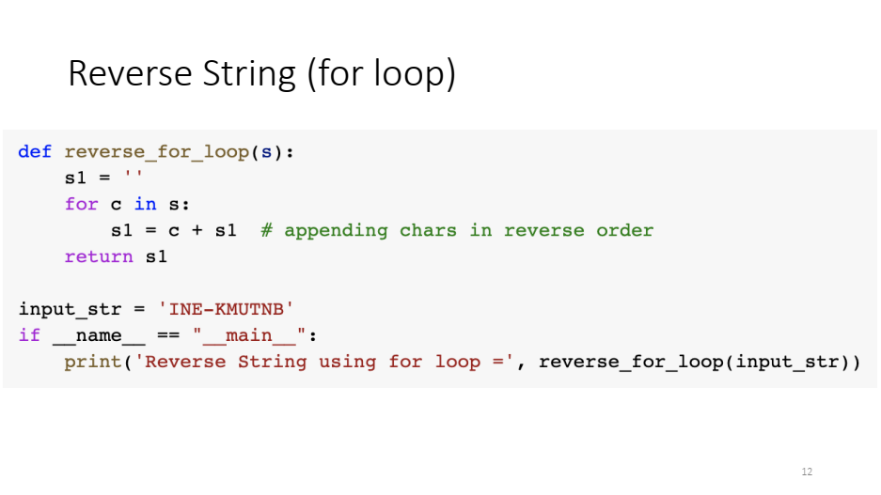
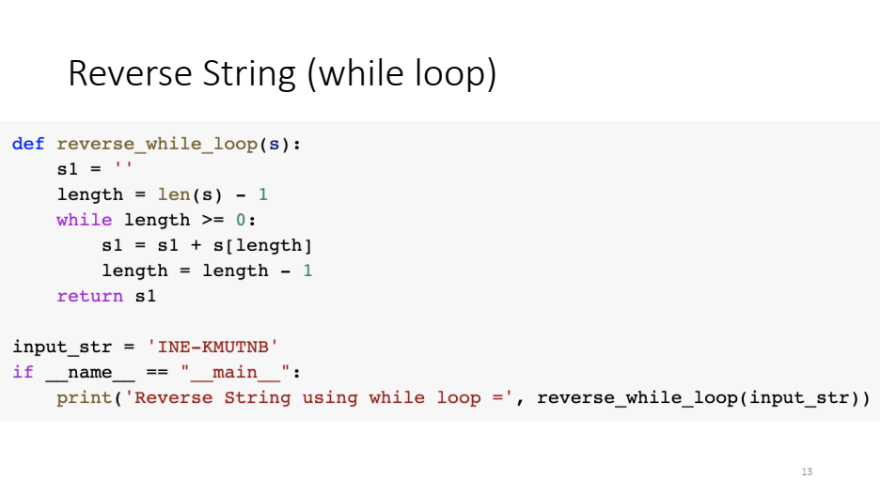
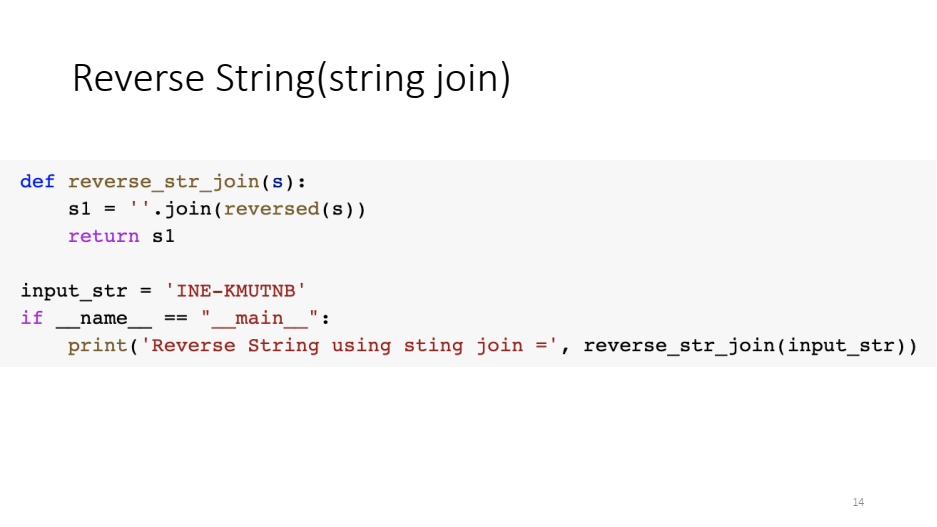
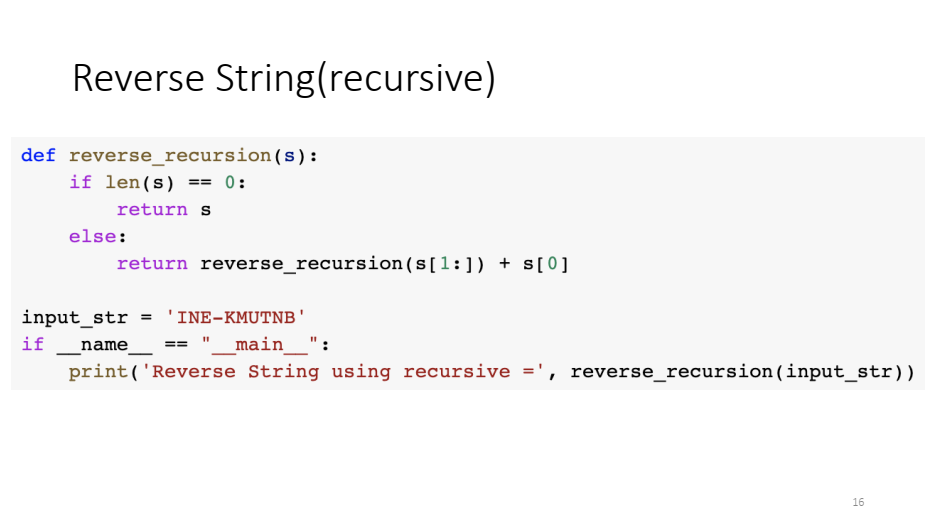
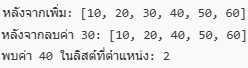
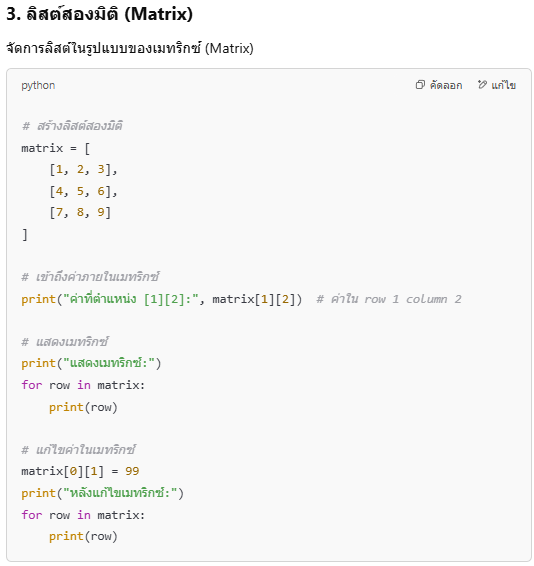
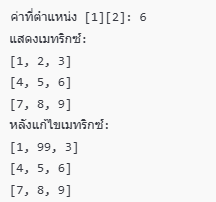
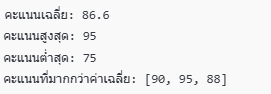
**Queue** ตัวอย่างโปรแกรมที่เก็บข้อมูลแบบคิว



**Stack** ตัวอย่างโปรแกรมที่เก็บข้อมูลแบบ stack



**List** ตัวอย่างโปรแกรมที่เก็บข้อมูลแบบ stack



พกไว้อุ่นใจนิดนึง

result = dict(zip(data,amount))

highest\_data = max(dict\_data, key=dict\_data.get)

highest\_value = dict\_data[highest\_data]