# Time complexity : O(n)

def find\_student(name):

    for student in enrolled\_student\_names:

        if student == name:

            print(f"{name} is found in the list")

            return

    print(f"{name} is not found in the list")

def find\_student\_usingset():

    if ("Amit" in enrolled\_student\_names): # Time complexity to search is O(1)

        print("Amit is found in the list")

    else:

        print("Amit is not found in the list")

    print("Amit" in enrolled\_student\_names)

def student\_data\_analysis():

    # Students who have enrolled in the course and also paid the fees

    # They must be present in both the list - enrolled and paid

    paid\_enrolled = enrolled\_student\_names & paid\_course\_student\_names

    print("List of enrolled and paid students are ")

    print(paid\_enrolled)

    all\_the\_students = enrolled\_student\_names | paid\_course\_student\_names

    print("List of all the students")

    print(all\_the\_students)

    unpaid\_students = enrolled\_student\_names - paid\_course\_student\_names

    print("List of students who need to pay the fees")

    print(unpaid\_students)

    yet\_to\_enroll\_but\_paid\_fees = paid\_course\_student\_names - enrolled\_student\_names

    print("Students who paid fees but not yet enrolled ")

    print(yet\_to\_enroll\_but\_paid\_fees)

    call\_from\_customercare\_final\_list = paid\_course\_student\_names ^ enrolled\_student\_names

    print("Need to be followed-up by our customer care team")

    print(call\_from\_customercare\_final\_list)

if \_\_name\_\_ == "\_\_main\_\_":

    enrolled\_student\_ids = { 100, 200, 101, 205, 305}

    #print(enrolled\_student\_ids)

    enrolled\_student\_names = { "Mahesh", "Mahesh", "Mahesh", "Santosh", "Sushil", "Amit", "Mahesh Bisur", "Sangeeta"}

    #print(enrolled\_student\_names)

    #find\_student("Amit")

    #find\_student\_usingset()

    paid\_course\_student\_names = { "Santosh", "Sushil", "Vidya"}

    student\_data\_analysis()