INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT010-3-1-PYP

PYTHON PROGRAMMING

CHANG SHIAU HUEI

APD1F2106CS(CYB)

HAND OUT DATE: 19TH JULY 2021

HAND IN DATE: 27TH SEPTEMBER 2021

WEIGHTAGE: 100%

INSTRUCTIONS TO CANDIDATES:

1. Submit your assignment online in MS Teams unless advised otherwise
2. Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld
3. Cases of plagiarism will be penalized
4. You must obtain at least 50% in each component to pass this module

TABLE OF CONTENTS

INTRODUCTION

Before the COVID-19 pandemic, Malaysia's health sector has always faced a huge problem —— staff shortages in hospitals or even clinics (Kanyakumari, 2019). This situation has been more apparent during the pandemic itself, and we need to find solutions to automate mundane tasks that don't require human interaction. Hence, this program written in Python acts as a menu-driven vaccination record management system to automate certain processes in administering for a vaccine. It consists of 5 main functions, namely for new patient registration, vaccine administration, searching patient records and vaccination status, showing statistical information on patients vaccinated and printing all patient records and vaccination status. By entering information into Section 1 and 2, the input will be written onto 2 files, which are "patients.txt" and "vaccination.txt". "patients.txt" will consist of relevant details about the registered user such as a unique vaccination ID, name, chosen vaccination center and vaccine, telephone number, email, and relevant medical information. "Vaccination.txt" will then consist of information related to the vaccine administration. For instance, vaccination ID, name, chosen vaccination center, vaccine and vaccination status.

ASSUMPTIONS

1. This program is being accessed by both vaccination centers (VC1 and VC2) on the cloud.
2. Users cannot access Section 2 to 5 if “patients.txt” and “vaccination.txt” are empty, in this case, only Section 1 can be chosen.
3. Name length is within 23 characters.
4. Patient ID is referred to as Vaccination ID.
5. All patients should go through the first dosage before the second dose unless they have opted for EC vaccine.
6. Users can administer their second dose on the same day as the first dose.
7. Users can change to any vaccination status by reinserting the information in Section 2 - Vaccine Administration. This is implemented so that patients can undo their errors.
8. Users can quit the program at any part of the process.
9. Capitalization in inputs does not influence the output.
10. The arrangement of data in “vaccination.txt” follows an order by the sequence of vaccination ID.
11. The arrangement of data in “vaccination.txt” follows an order by the latest vaccination administration.

DESIGN OF THE PROGRAM

PSEUDOCODE

```
PROGRAM VaccinationRecordManagementSystem
BEGIN
    IMPORT os, sys
    FROM os IMPORT path

    FUNCTION fileCheck()
        IF (os.path.isfile("patients.txt") == False) THEN
            fh = OPENFILE “patients.txt” for WRITE
            CLOSEFILE fh
        ENDIF
        IF (os.path.isfile("vaccination.txt") == False) THEN
            fh = OPENFILE “vaccination.txt” for WRITE
            CLOSEFILE fh
        ENDIF
        RETURN
    ENDFUNCTION

    FUNCTION listCheck(string)
        separate = list(string.split("|"))
        RETURN separate
    ENDFUNCTION

    FUNCTION characterCheck(string)
        separate = list(string.split("-"))
        RETURN separate
    ENDFUNCTION

    FUNCTION lineCount(fileName)
        fh = OPENFILE fileName for READ
```

```
        lineCount = 0
        FOR EACH line IN fh
                IF (line != newline) THEN
                        lineCount = lineCount + 1
                ENDIF
                NEXT line
        ENDFOR
        CLOSEFILE fh
        RETURN lineCount
ENDFUNCTION

FUNCTION integerValidation(lineShown)
        DOWHILE True
                TRY
                        Print lineShown
                        Read item
                EXCEPT
                        Print "Your input is invalid. Please try again by entering integers.”
                        Print newline
                        CONTINUE
                ENDTRY
                IF item == -1 THEN
                        sys.exit()
                ELSE
                        RETURN item
                ENDIF
        ENDDO
ENDFUNCTION

FUNCTION floatValidation(lineShown)
        DOWHILE True
                TRY
                        Print lineShown
```

```
                Read item
                decimal = round(item,  2)
            EXCEPT
                Print "Your input is invalid. Please try again by entering integers or floats.”
                Print newline
                CONTINUE
            ENDTRY
            IF (item == -1) THEN
                sys.exit()
            ELSE
                RETURN decimal
            ENDIF
        ENDDO
ENDFUNCTION

FUNCTION vacIDValid(fileUsed)
        Print “Please insert your vaccination ID (Enter 'X' to quit): “
        Read vacID.upper()
        IF vacID == “X” THEN
            sys.exit()
        ELSE
            WITH OPEN fileName with READ mode AS fh
                FOR EACH row IN fh
                    vacIDLine = call listCheck(row)
                    IF vacIDLine[0] == vacID THEN
                        BREAK
                    ENDIF
                    NEXT row
                ENDFOR
            ENDWITH
        ENDIF
        RETURN vacIDLine, vacID
ENDFUNCTION
```

```
FUNCTION vaccinationCenterSelection()
    DOWHILE True
        Print “Which vaccination centre are you going to? [ VC1 / VC2 ]"
        Print “Enter 'X' to quit: "
        Read vc
        IF (vc.upper() == “X”) THEN
            sys.exit()
        ELSE
            IF (vc.upper() != "VC1") AND (vc.upper() != "VC2") AND (vc.upper() !=
            "X") THEN
                Print “Please choose either VC1 or VC2 only.”
                Print newline
                CONTINUE
        ELSE
            IF (vc.upper() == "VC1") THEN
                vc = “VC1”
        ELSE
            vc = “VC2”
        ENDIF
        RETURN vc
    ENDDO
ENDFUNCTION

FUNCTION nameCheck(fileUsed)
    DOWHILE True
        valid = True
        Print “Please insert your name (Enter 'X' to quit): “
        Read name.upper()
        IF (name == "X") THEN
            sys.exit()
        ELSE
            IF (len(name) > 23) THEN
```

```
                    Print “Please input a name within 23 characters."
                    Print newline
            ELSE
                WITH OPEN fileUsed with READ mode AS fh
                    info = fh.readlines()
                    IF (call lineCount(fileUsed) > 0) THEN
                        FOR EACH row IN info
                            nameLine = call listCheck(row)
                            IF (nameLine[1].upper() == name.upper()) THEN
                                Print "A record with this name already
                                exists.”
                                Print newline
                                valid = False
                            ENDIF
                            NEXT row
                        ENDFOR
                        IF (valid == True) THEN
                            RETURN name
                        ENDIF
                    ELSE
                        RETURN name
                    ENDIF
                ENDWITH
            ENDIF
        ENDDO
ENDFUNCTION

FUNCTION vaccineSelection(listUsed)
    Print "You are elligble for these vaccines:", listUsed
    DOWHILE True
        Print “Please choose only one vaccine from the list above (Enter 'X' to
        exit): "
        Read vaccine.upper()
```

```
            IF (vaccine == "X") THEN
                sys.exit()
            ENDIF
            FOR EACH vac IN listUsed
                IF (vac.upper() == vaccine.upper()) THEN
                    Print "You have chosen " + vaccine + "."
                    RETURN vaccine
                ELSE
                    CONTINUE
                ENDIF
                NEXT vac
            ENDFOR
        ENDDO
ENDFUNCTION

FUNCTION uniqueIDGenerator(vc)
        vacID = 0
        WITH OPEN “patients.txt” with READ mode AS fh
            IF (call lineCount("patients.txt") > 0) THEN
                lines = fh.readlines()
                lastLine = lines[-1:]
                vacIDLine = call listCheck(lastLine)
                vacIDSection = call characterCheck(vacIDLine[0])
                increment = vacIDSection[1] + 1
                vacID = vc + "-" + increment
            ELSE
                vacID = vc + "-1"
            ENDIF
        ENDWITH
        RETURN vacID
ENDFUNCTION

FUNCTION extraInformation()
```

```
        Print “Please input your height in cm (Enter '-1' to quit): "
        Read call floatValidation(height)
        h = "Height:" + height + "cm"
        Print "Please input your weight in kg (Enter '-1' to quit): "
        Read call floatValidation(weight)
        w = "Weight:" + str(weight) + "kg”
        RETURN h, w
ENDFUNCTION

FUNCTION printResultsSectionOne(name, vacID)
        Print newline
        Print " Registration Details ".center(100, "=")
        Print newline
        Print "You have successfully registered.".center(100, " ")
        Print newline
        Print “\tUser:", name
        Print “\tYour vaccination ID is " + vacID + "."
        Print newline
        Print newline
        Print “Do remember to head to 'Section 2 - Vaccine Administration' after taking your
        first vaccination dose!"
        RETURN
ENDFUNCTION

FUNCTION newPatientRegistration()
        DOWHILE True
                vc = call vaccinationCenterSelection()
                name = (call nameCheck("patients.txt")).upper()
                Print "Please insert your age (Enter '-1' to quit): "
                Read call integerValidation(age)
                elligibleVac = []
                IF (age >= 12) THEN
                        elligibleVac.append("AF")
```

```
            elligibleVac.append("DM")
            IF (age >= 18) THEN
                elligibleVac.append("BV")
                elligibleVac.append("EC")
            ENDIF
            IF (age <= 45) THEN
                elligibleVac.append("CZ")
            ENDIF
    ELSE
            Print "You are not elligible for any vaccine."
            BREAK
    ENDIF
    vaccine = call vaccineSelection(elligibleVac)
    Print “Please insert your phone number (Without '-', Enter '-1' to exit): "
    Read call integerValidation(phoneNum)
    phoneNum = "0" + phoneNum
    Print "Please insert your email (Enter 'X' to quit): "
    Read email
    IF (email.upper() == "X") THEN
            sys.exit()
    ENDIF
    vacID = call uniqueIDGenerator(vc)
    height, weight = call extraInformation()
    Call printResultsSectionOne(name, vacID)
    WITH OPEN “patients.txt” with APPEND mode AS fh
            fh.write(f"{vacID}|{name}|{vc}|{age}|{vaccine}|{phoneNum}|{email}|{he
            ight}|{weight}")
            Print newline
    ENDWITH
    WITH OPEN “vaccination.txt” with APPEND mode AS fh
            fh.write(f"{vacID}|{name}|{vc}|{vaccine}|NEW”)
            Print newline
    ENDWITH
```

```
            BREAK
        ENDDO
        RETURN
ENDFUNCTION

FUNCTION vaccinationStatusAndIntervalBetweenDoses(vaccine)
        DOWHILE True
            valid = True
            IF (vaccine == "EC") THEN
                Print "Have you received your first vaccination dose? [ Y / N ]"
                Print "Enter 'X' to quit: "
                Read choice.upper()
                IF (choice == "X") THEN
                    sys.exit()
                ELSE
                    IF (choice == "Y") THEN
                        status = "COMPLETED"
                        Print "You have finished your vaccination."
                ELSE
                    IF (choice == "N") THEN
                        status = "NEW"
                        Print "Come back when you have received your first
                        vaccination dose."
                ELSE
                    Print "Invalid input, please try again."
                    Print newline
                    valid = False
                ENDIF
            ELSE
                Print "Have you received your first vaccination dose? [ Y / N ]"
                Print "Enter 'X' to quit: "
                Read choice.upper()
                IF (choice == "X") THEN
```

```
            sys.exit()
        ELSE
            IF (choice == "Y") THEN
                Print "Have you received your second vaccination dose? [ Y
                / N ]"
                Print "Enter 'X' to quit: "
                Read choice2.upper()
                IF (choice2 == "X") THEN
                    sys.exit()
                ELSE
                    IF (choice2 == "N") THEN
                        IF (vaccine == "AF") THEN
                            intervalBetweenDoses = 2
                        ELSE
                            IF (vaccine == "BV") OR (vaccine ==
                            "CZ") THEN
                                intervalBetweenDoses = 3
                        ELSE
                            intervalBetweenDoses = 4
                        ENDIF
                        status = "COMPLETED-D1"
                        Print "Please come after",
                        intervalBetweenDoses, "weeks for your
                        second vaccination dose."
                    ELSE
                        IF (choice2 == "Y") THEN
                            status = "COMPLETED"
                            Print "You have finished your
                            vaccination."
                    ELSE
                        Print "Invalid input, please try again."
                        Print newline
                        valid = False
```

```
                        ENDIF
                    ENDIF
                ELSE
                    IF (choice == "N") THEN
                        status = "NEW"
                        Print "Come back when you have received your first
                        vaccination dose."
                ELSE
                    Print "Invalid input, please try again."
                    Print newline
                    valid = False
                ENDIF
            ENDIF
        ENDIF
        IF (valid == True) THEN
            RETURN status
        ENDIF
    ENDDO
ENDFUNCTION

FUNCTION deleteDuplicate(vacID)
    WITH OPEN "vaccination.txt" with READ mode AS fh
        lines = fh.readlines()
        WITH OPEN "vaccination.txt" with WRITE mode AS fh
            FOR EACH line IN lines
                IF (line.find(vacID) != -1) THEN
                    PASS
                ELSE
                    fh.write(line)
                ENDIF
                NEXT line
            ENDFOR
        ENDWITH
```

```
        ENDWITH
        RETURN
ENDFUNCTION

FUNCTION printResultsSectionTwo(name, vacID, vc, vaccine, status)
        Print newline
        Print " Vaccine Administration ".center(100, "=")
        Print newline
        Print "\t1. Name\t\t\t:\t\t", name
        Print "\t2. Vaccination ID\t\t:\t\t", vacID
        Print "\t3. Vaccination Center\t\t:\t\t", vc
        Print "\t4. Vaccine\t\t\t:\t\t", vaccine
        Print "\t5. Vaccination Status\t\t:\t\t", status
        RETURN
ENDFUNCTION

FUNCTION vaccineAdministration()
        vacIDLine, vacID = call vacIDValid("patients.txt")
        IF (vacIDLine[0] == vacID) THEN
                name = (vacIDLine[1]).upper()
                vc = vacIDLine[2]
                vaccine = vacIDLine[4]
                Print "The selected vaccine for user " + name + " is " + vaccine + "."
                status = call vaccinationStatusAndIntervalBetweenDoses(vaccine)
                Call printResultsSectionTwo(name, vacID, vc, vaccine, status)
                Call deleteDuplicate(vacID)
                WITH OPEN "vaccination.txt" with APPEND mode AS fh
                        fh.write(f"{vacID}|{name}|{vc}|{vaccine}|{status}")
                        Print newline
                ENDWITH
        ELSE
                Print "This vaccination ID doesn't exist. Please register in 'Section 1 - New Patient
                Registration' before proceeding to this section."
```

```
        ENDIF
        RETURN
ENDFUNCTION

FUNCTION totalDosageRequired(vaccine)
        IF (vaccine == "EC") THEN
                totalDosage = 1
        ELSE
                totalDosage = 2
        ENDIF
        RETURN totalDosage
ENDFUNCTION

FUNCTION dosageRequired(status, vaccine)
        IF (status == "COMPLETED\n") THEN
                dosage = 0
        ELSE
                IF (vaccine == "EC") THEN
                        dosage = 1
                ELSE
                        IF (status == "NEW\n") THEN
                                dosage = 2
                        ELSE
                                dosage = 1
                        ENDIF
                ENDIF
        ENDIF
        RETURN dosage
ENDFUNCTION

FUNCTION printResultsSectionThree(name, vacID, vc, vaccine, totalDosage, dosage, status)
        Print newline
        Print " Search Patient Record & Vaccination Status ".center(100, "=")
```

```
        Print newline
        Print "\t1. Name\t\t\t:\t\t", name
        Print "\t2. Vaccination ID\t\t:\t\t", vacID
        Print "\t3. Vaccination Center\t\t:\t\t", vc
        Print "\t4. Vaccine\t\t\t:\t\t", vaccine
        Print "\t5. Total dosage Required\t:\t\t", totalDosage
        Print "\t6. Dosage Required\t\t:\t\t", dosage
        Print "\t7. Vaccination Status\t\t:\t\t", status
        RETURN
ENDFUNCTION

FUNCTION patientRecordAndVaccineStatus()
        vacIDLine, vacID = call vacIDValid("vaccination.txt")
        IF (vacIDLine[0] == vacID.upper()) THEN
                name = vacIDLine[1]
                vc = vacIDLine[2]
                vaccine = vacIDLine[3]
                status = vacIDLine[4]
                totalDosage = call totalDosageRequired(vaccine)
                dosage = call dosageRequired(status, vaccine)
                Call printResultsSectionThree(name, vacID, vc, vaccine, totalDosage, dosage,
                status)
        ELSE
                Print "This vaccination ID does not exist. Please register in 'Section 1 - New Patient
                Registration' to get a vaccination ID."
        ENDIF
        RETURN
ENDFUNCTION

FUNCTION printResultsSectionFour(totalVC1, totalWaitingD2VC1, totalVaccinatedVC1,
totalVC2, totalWaitingD2VC2, totalVaccinatedVC2, totalWaitingD2, totalVaccinated)
        Print newline
        Print " Statistical Information on Patients Vaccinated ".center(100, "=")
```

```
        Print newline
        Print “\tFor VC1:”
        Print newline
        Print “\t\tNumber of people receiving vaccine in VC1\t:\t", totalVC1
        Print "\t\tPeople who are waiting for dose 2\t\t:\t", totalWaitingD2VC1
        Print "\t\tPeople who have completed vaccination\t\t:\t", totalVaccinatedVC1
        Print newline
        Print “\tFor VC2:”
        Print “\t\tNumber of people receiving vaccine in VC2\t:\t", totalVC2
        Print "\t\tPeople who are waiting for dose 2\t\t:\t", totalWaitingD2VC2
        Print "\t\tPeople who have completed vaccination\t\t:\t", totalVaccinatedVC2
        Print newline
        Print "-" * 100
        Print newline
        Print “\t\tTotal people who are waiting for dose 2\t\t:\t", totalWaitingD2
        Print "\t\tTotal people that have completed vaccination\t:\t", totalVaccinated
        RETURN
ENDFUNCTION


FUNCTION statisticalInfoOnPatientsVaccinated()
        totalVC1 = 0
        totalVC2 = 0
        totalWaitingD2VC1 = 0
        totalVaccinatedVC1 = 0
        totalWaitingD2VC2 = 0
        totalVaccinatedVC2 = 0
        WITH OPEN "vaccination.txt" with READ mode AS fh
                FOR EACH row IN fh
                        lineInfo = call listCheck(row)
                        IF (lineInfo[2] == "VC1") THEN
                                totalVC1 = totalVC1 + 1
                                IF (lineInfo[4] == "COMPLETED-D1\n") THEN
```

```
                totalWaitingD2VC1 = totalWaitingD2VC1 + 1
            ENDIF
            IF (lineInfo[4] == "COMPLETED\n") THEN
                totalVaccinatedVC1 = totalVaccinatedVC1 + 1
            ENDIF
        ELSE
            totalVC2 = totalVC2 + 1
            IF (lineInfo[4] == "COMPLETED-D1\n") THEN
                totalWaitingD2VC2 = totalWaitingD2VC2 + 1
            ENDIF
            IF (lineInfo[4] == "COMPLETED\n") THEN
                totalVaccinatedVC2 = totalVaccinatedVC2 + 1
            ENDIF
        ENDIF
        NEXT row
    ENDFOR
ENDWITH
totalWaitingD2 = totalWaitingD2VC1 + totalWaitingD2VC2
totalVaccinated = totalVaccinatedVC1 + totalVaccinatedVC2
Call printResultsSectionFour(totalVC1, totalWaitingD2VC1, totalVaccinatedVC1,
totalVC2, totalWaitingD2VC2, totalVaccinatedVC2, totalWaitingD2, totalVaccinated)
RETURN
ENDFUNCTION

FUNCTION readAllPatientRecords()
    allRecords = []
    WITH OPEN "vaccination.txt" with READ mode AS fh
        FOR EACH row IN fh
            patientInformation = call listCheck(row)
            allRecords.append(patientInformation)
            NEXT row
        ENDFOR
    ENDWITH
```

```
        RETURN allRecords
ENDFUNCTION

FUNCTION printAllPatientRecords(records)
        Print newline
        Print "=" * 126
        Print "|" + "Vaccination ID".center(25) + "|" + "Name".center(25) + "|" + "Vaccination
        Center".center(25) + "|" + "Vaccine".center(20) + "|" + "Status".center(25) + "|"
        Print "=" * 126
        LOOP counter FROM 0 TO len(records) STEP 1
                items = records[counter]
                Print("| " + items[0].center(25) + "| " + items[1].ljust(24) + "|" + items[2].center(25)
                + "|" + items[3].center(20) + "|" + (items[4].rstrip()).center(25) + "|")
                NEXT counter
        ENDLOOP
        Print "=" * 126
        RETURN
ENDFUNCTION

FUNCTION printAllPatientRecordsAndVaccineStatus()
        allRecords = call readAllPatientRecords()
        Call printAllPatientRecords(allRecords)
        RETURN
ENDFUNCTION

FUNCTION menu()
        DOWHILE True
                Call fileCheck()
                Print newline
                Print " Welcome to the Vaccination Record Management System ".center(100,
                "=")
                Print newline
                Print “\t1. New Patient Registration
```

```
Print newline
Print ”\t2. Vaccine Administration”
Print newline
Print “\t3. Search Patient Record & Vaccination Status”
Print newline
Print “\t4. Statistical Information on Patients Vaccinated”
Print newline
Print “\t5. Print All Patient Record & Vaccination Status”
Print newline
Print “\t6. Exit"
TRY
        Print newline
        Print "Please choose any operation from the given options: "
        Read choice
EXCEPT
        Print "Your input is invalid. Please try again."
        CONTINUE
ENDTRY
IF (choice <= 0) OR (choice > 6) THEN
        Print "Please choose a number from 1 to 6 only."
        CONTINUE
ELSE
        IF (choice == 1) THEN
                Call newPatientRegistration()
                CONTINUE
ELSE
        IF (choice == 2) THEN
                IF (call lineCount("patients.txt") > 0) THEN
                        Call vaccineAdministration()
                ELSE
                        Print "No record exists. Please register in 'Section 1 - New
                        Patient Registration' before proceeding to this section."
                ENDIF
```

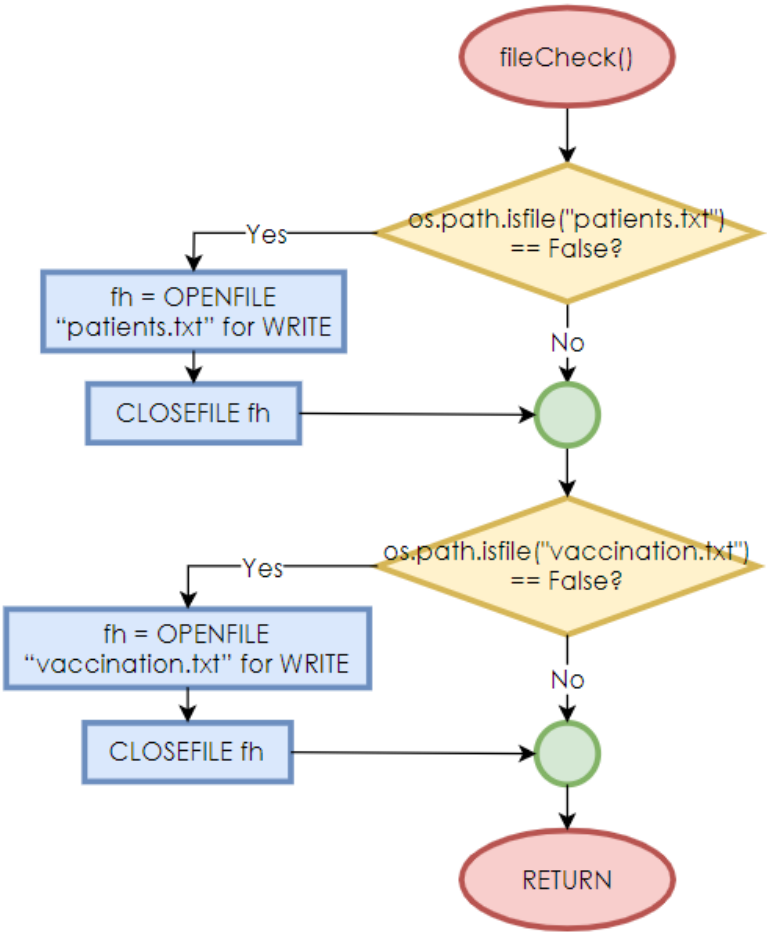
```
                CONTINUE
        ELSE
            IF (choice == 3) THEN
                IF (call lineCount("patients.txt") > 0) THEN
                    Call patientRecordAndVaccineStatus()
                ELSE
                    Print "No record exists. Please register in 'Section 1 - New
                    Patient Registration' before proceeding to this section."
                ENDIF
                CONTINUE
        ELSE
            IF (choice == 4) THEN
                IF (call lineCount("patients.txt") > 0) THEN
                    Call statisticalInfoOnPatientsVaccinated()
                ELSE
                    Print "No record exists. Please register in 'Section 1 - New
                    Patient Registration' before proceeding to this section."
                ENDIF
                CONTINUE
        ELSE
            IF (choice == 5) THEN
                IF (call lineCount("patients.txt") > 0) THEN
                    Call printAllPatientRecordsAndVaccineStatus()
                ELSE
                    Print "No record exists. Please register in 'Section 1 - New
                    Patient Registration' before proceeding to this section."
                ENDIF
                CONTINUE
        ELSE
            BREAK
        ENDIF
    ENDDO
    RETURN
```

```
    ENDFUNCTION

    Call menu()
END
```

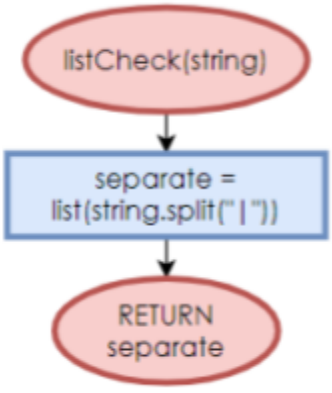
FLOWCHART

GENERAL FUNCTIONS

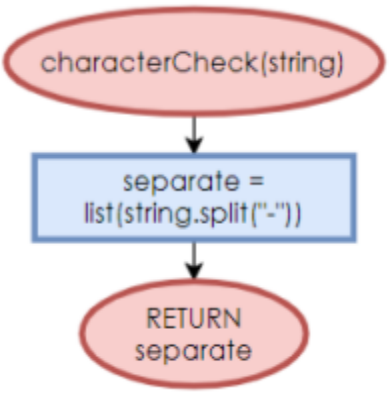
fileCheck()



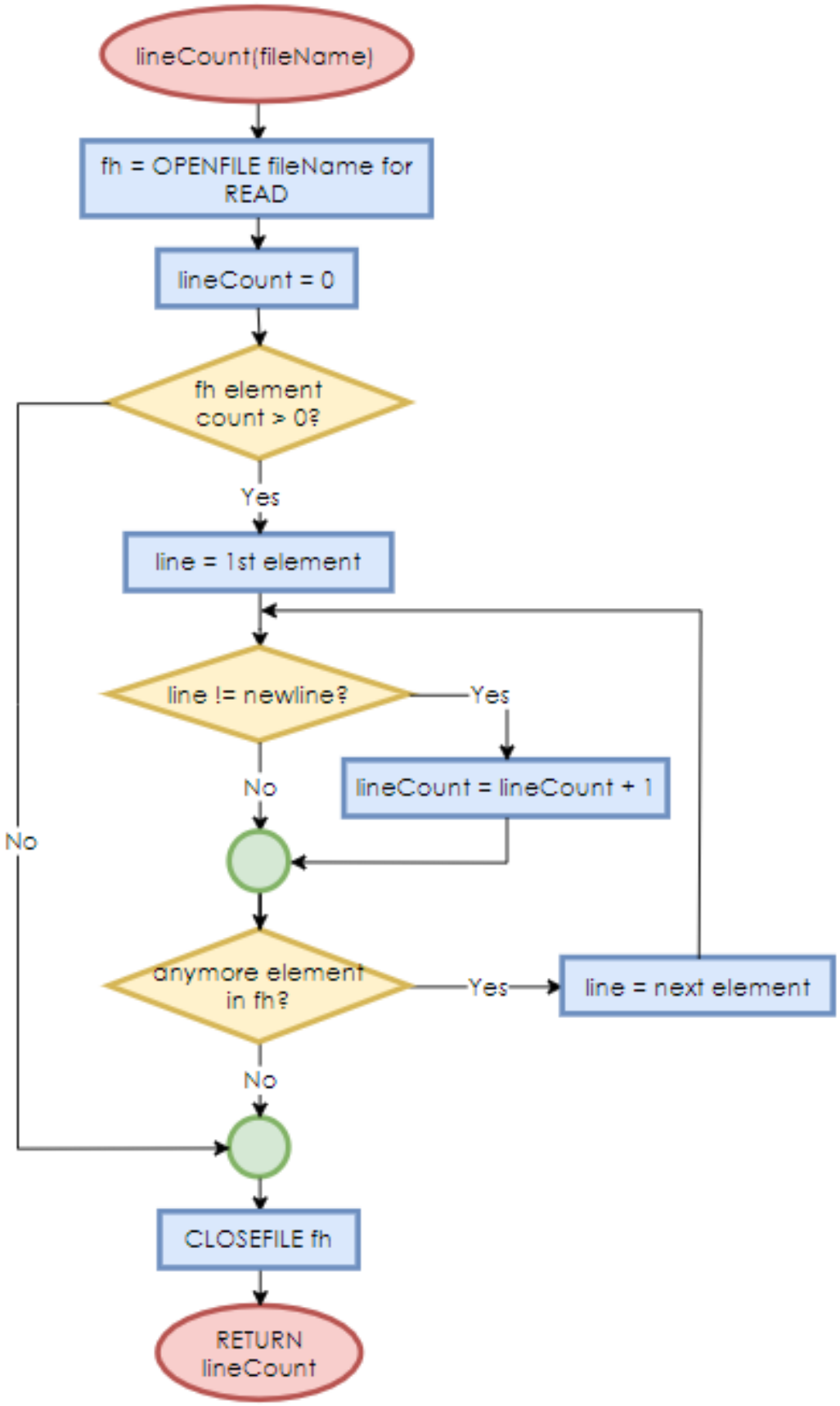
listCheck(string)



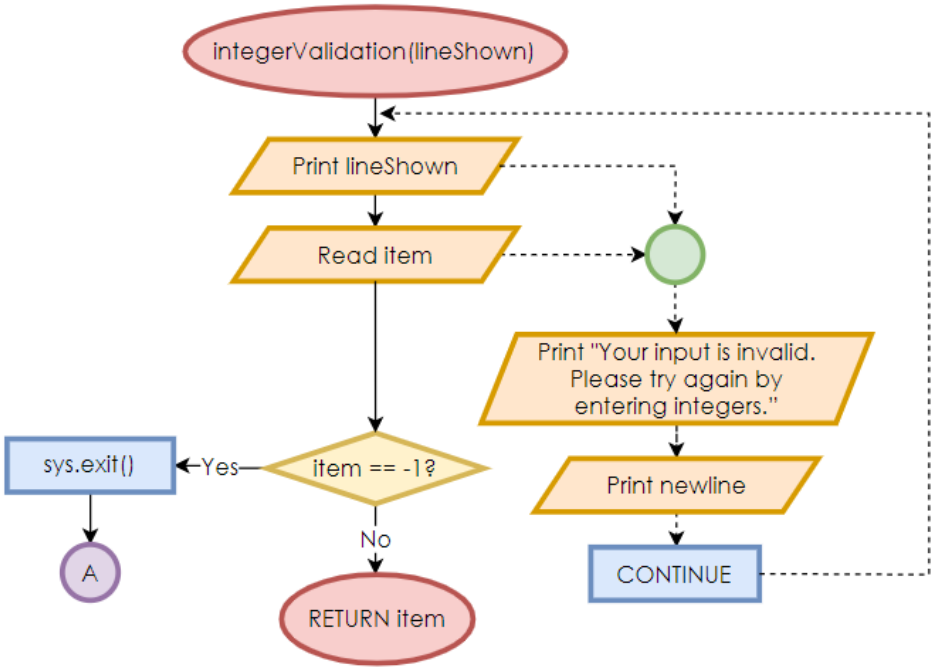
characterCheck(string)



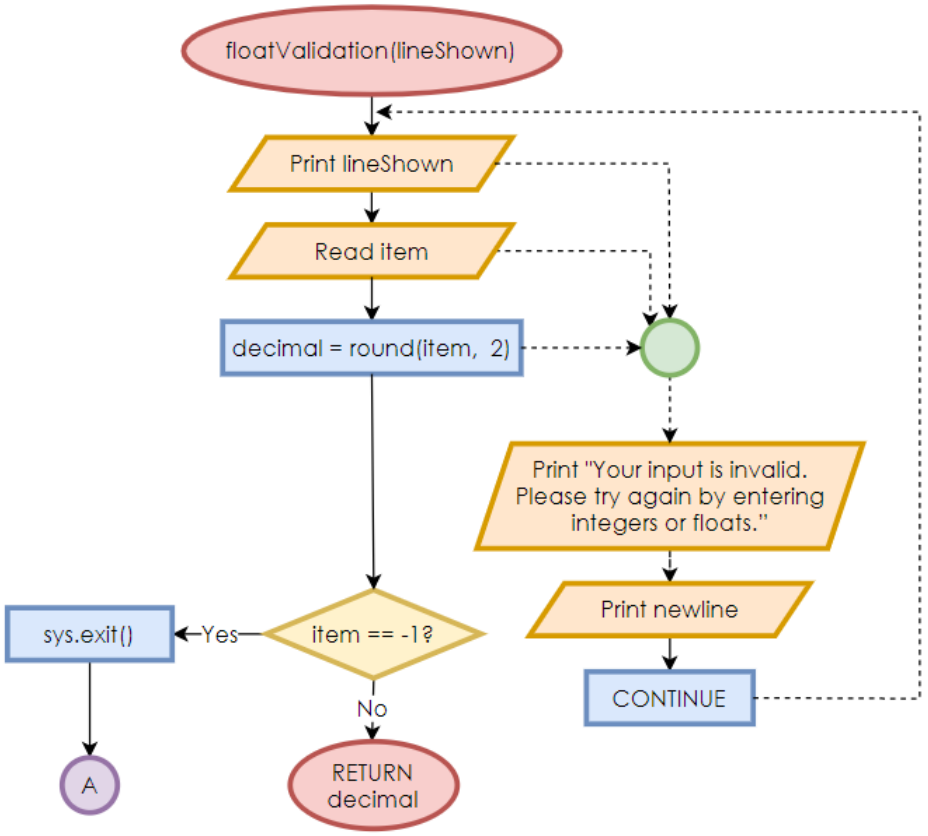
lineCount(fileName)



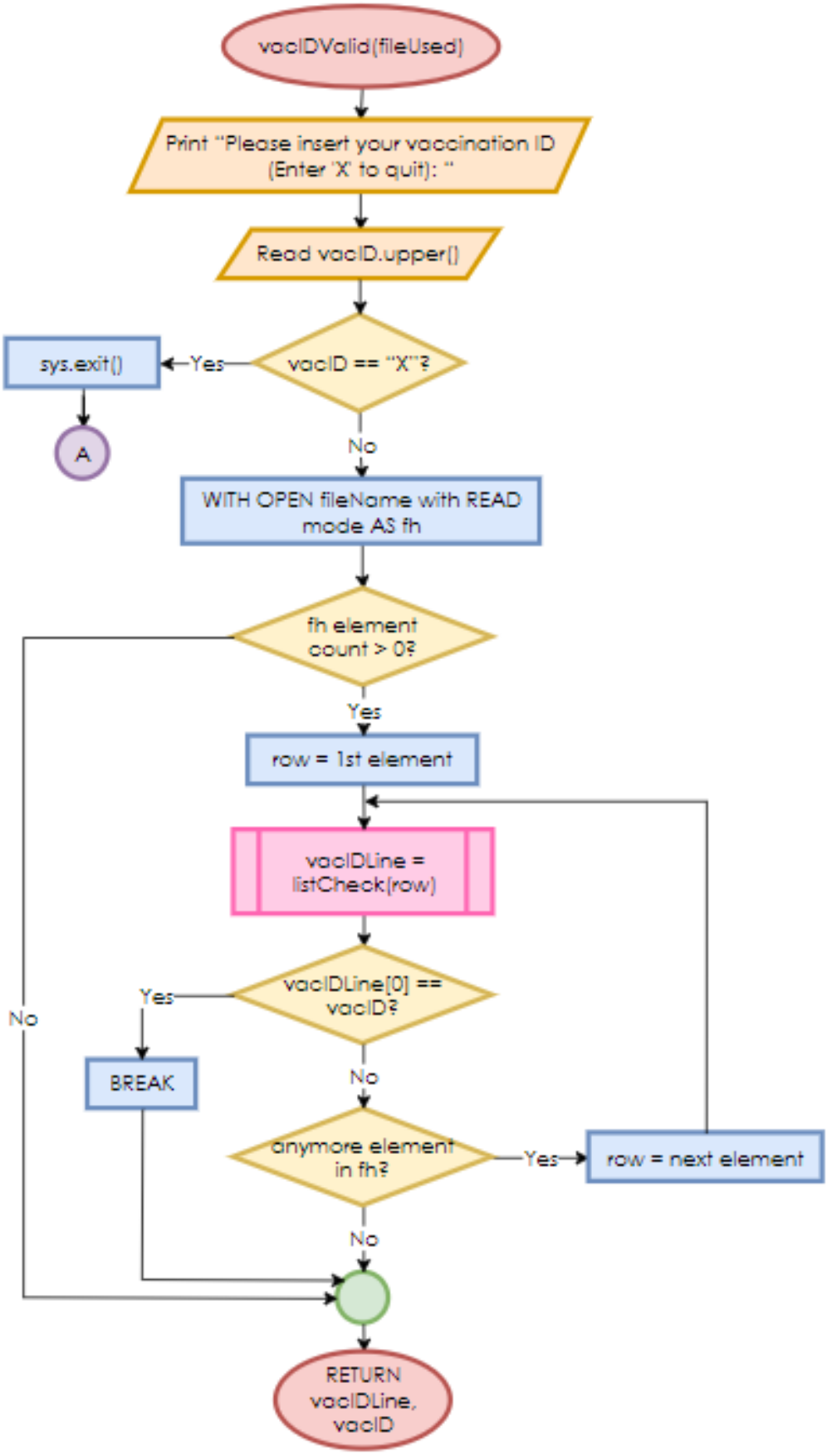
integerValidation(lineShown)



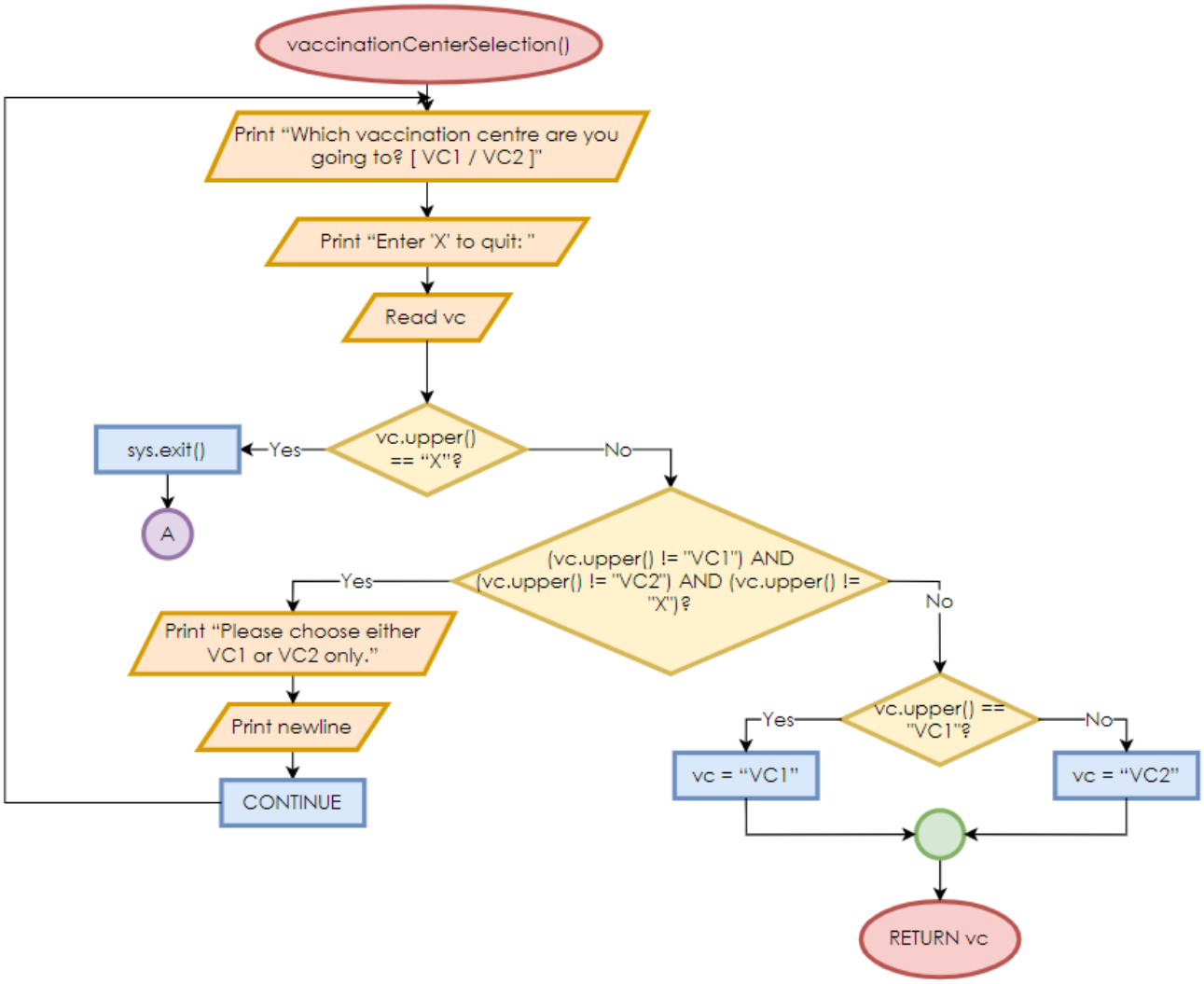
floatValidation(lineShown)



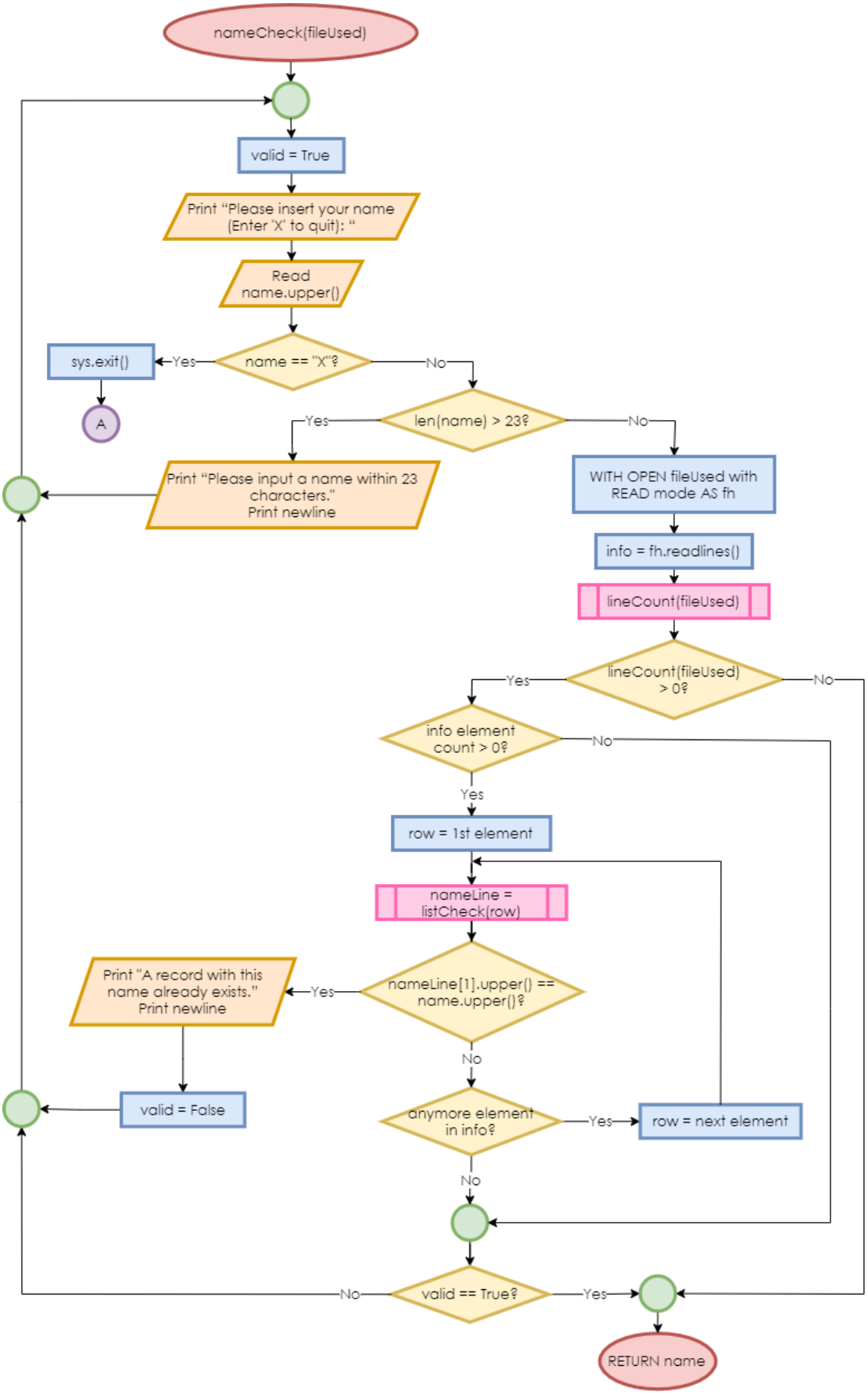
vacIDValid(fileUsed)



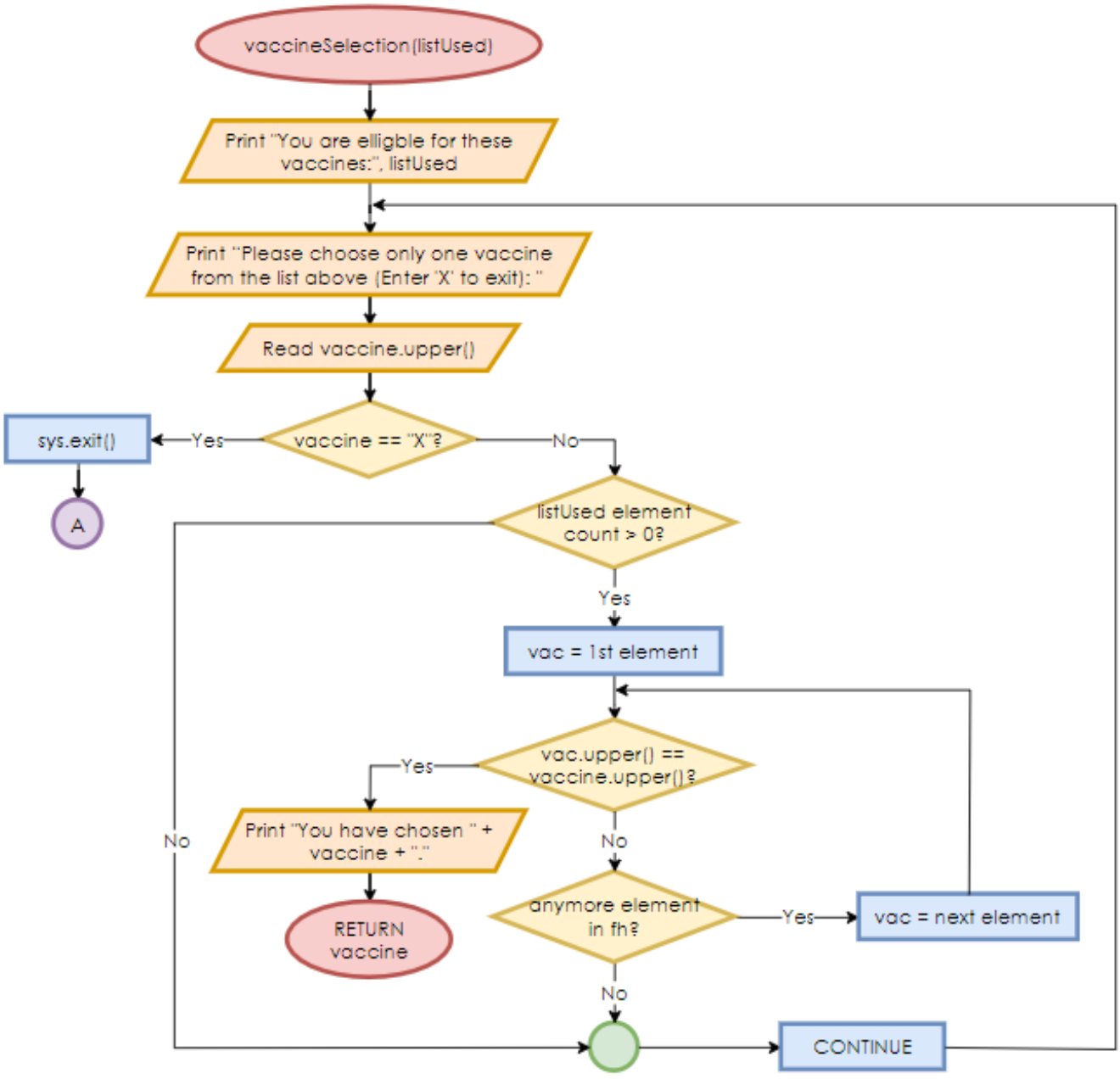
vaccinationCenterSelection()



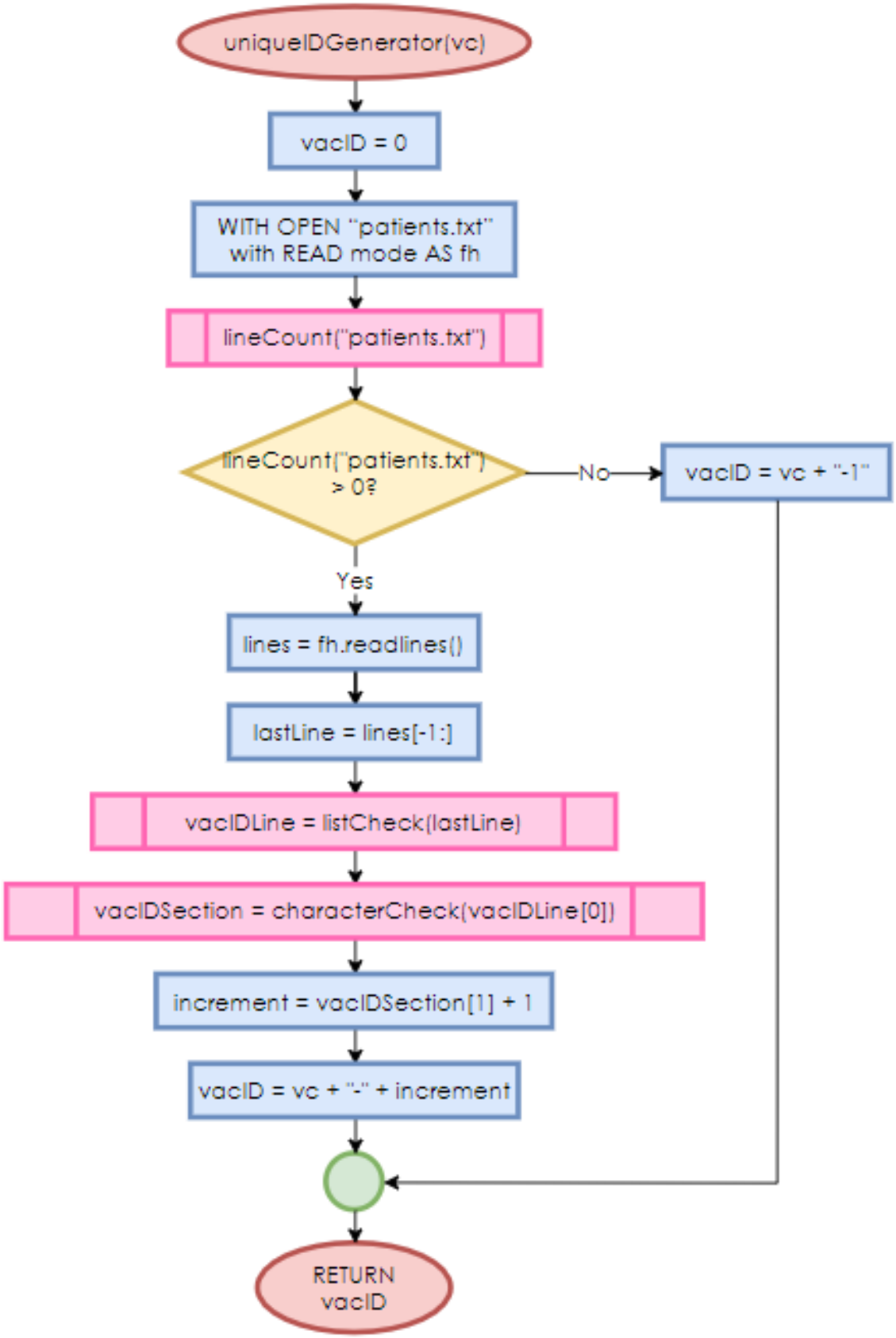
nameCheck(fileUsed)



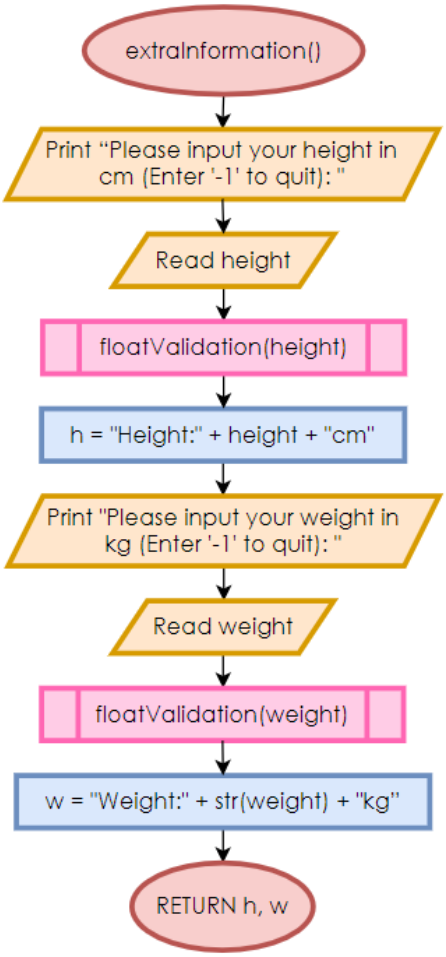
vaccineSelection(listUsed)



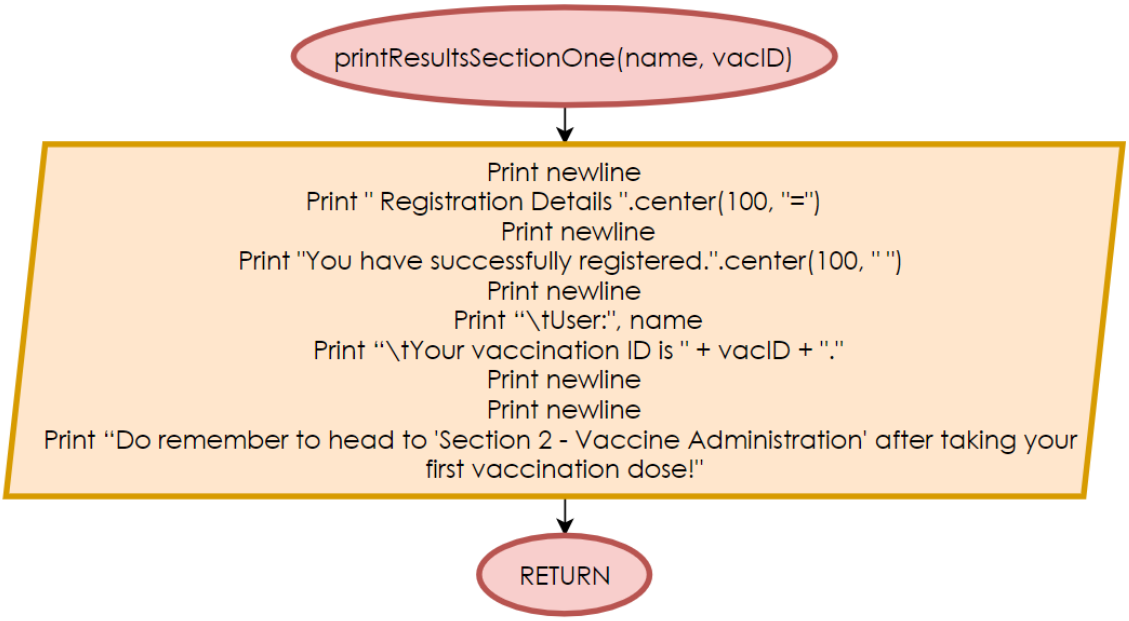
uniqueIDGenerator(vc)



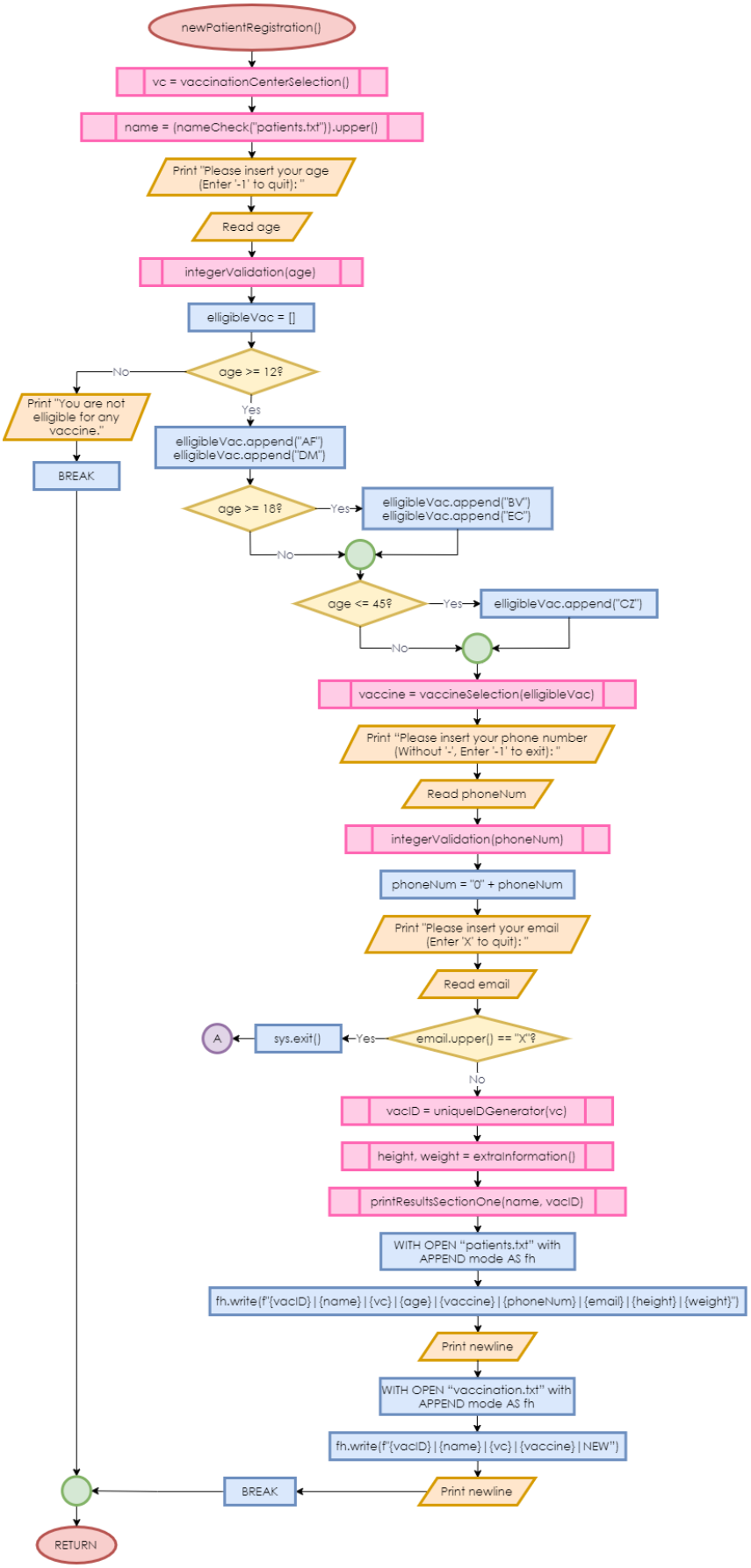
extraInformation()



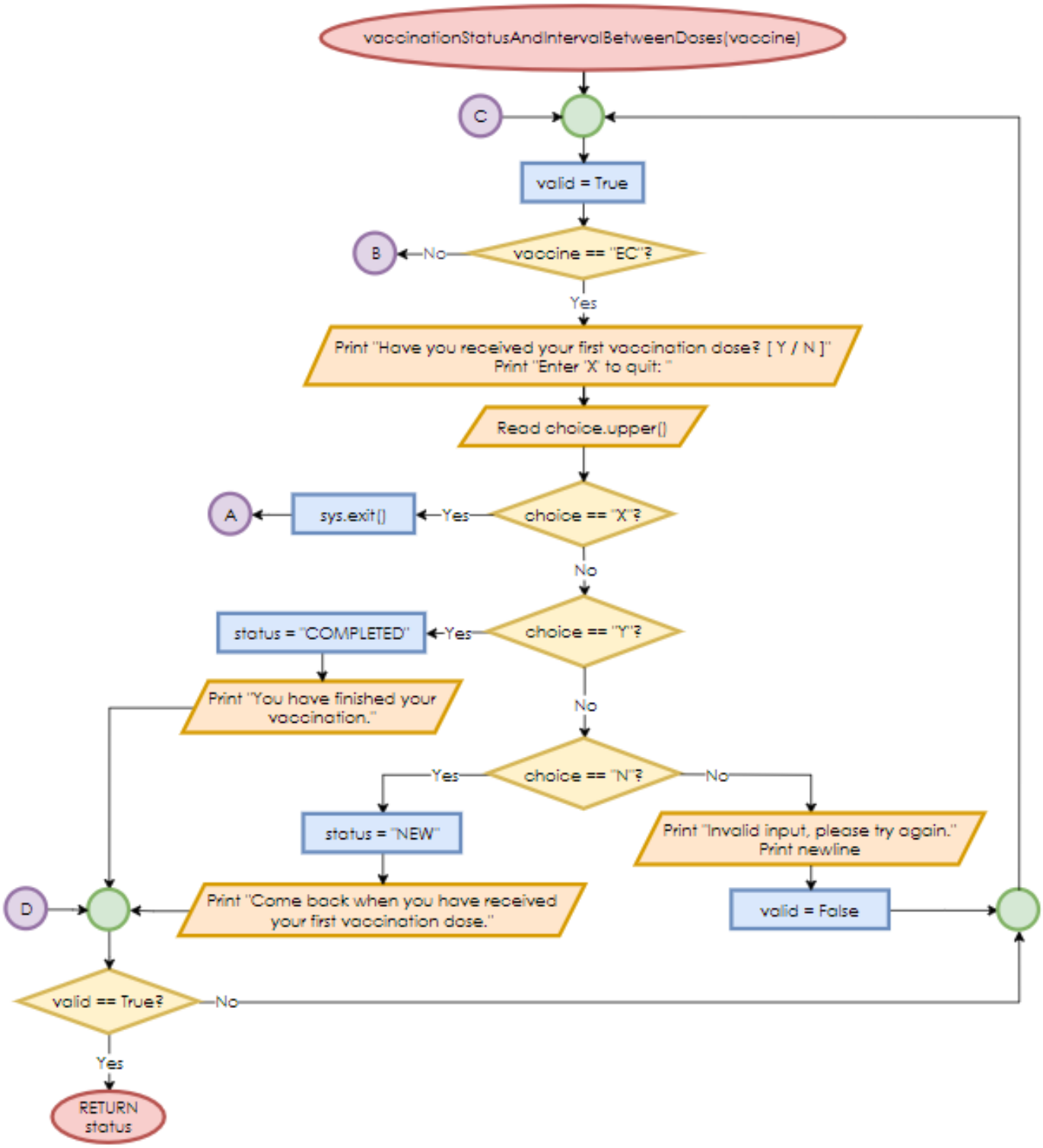
printResultsSectionOne(name, vacID)

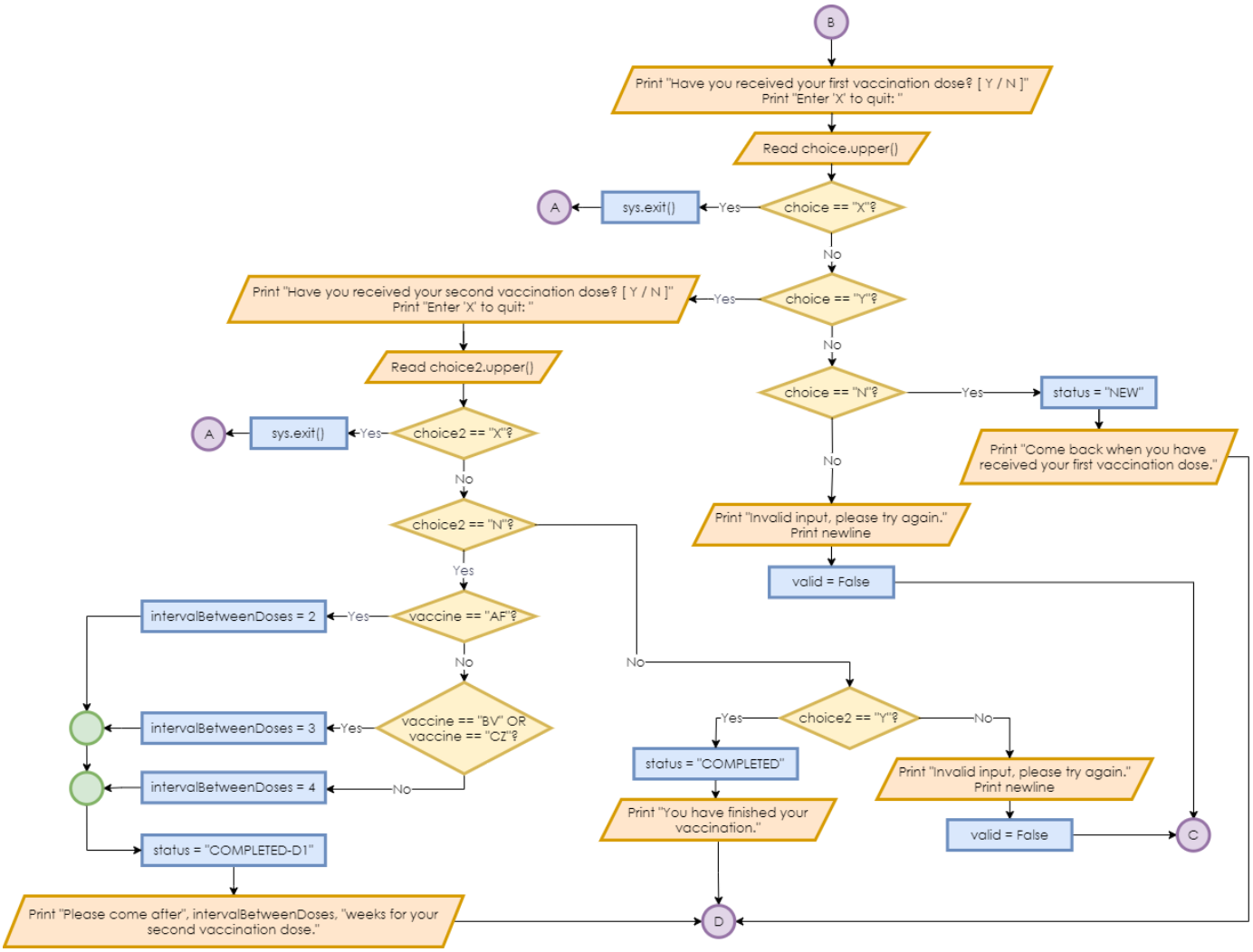


newPatientRegistration()

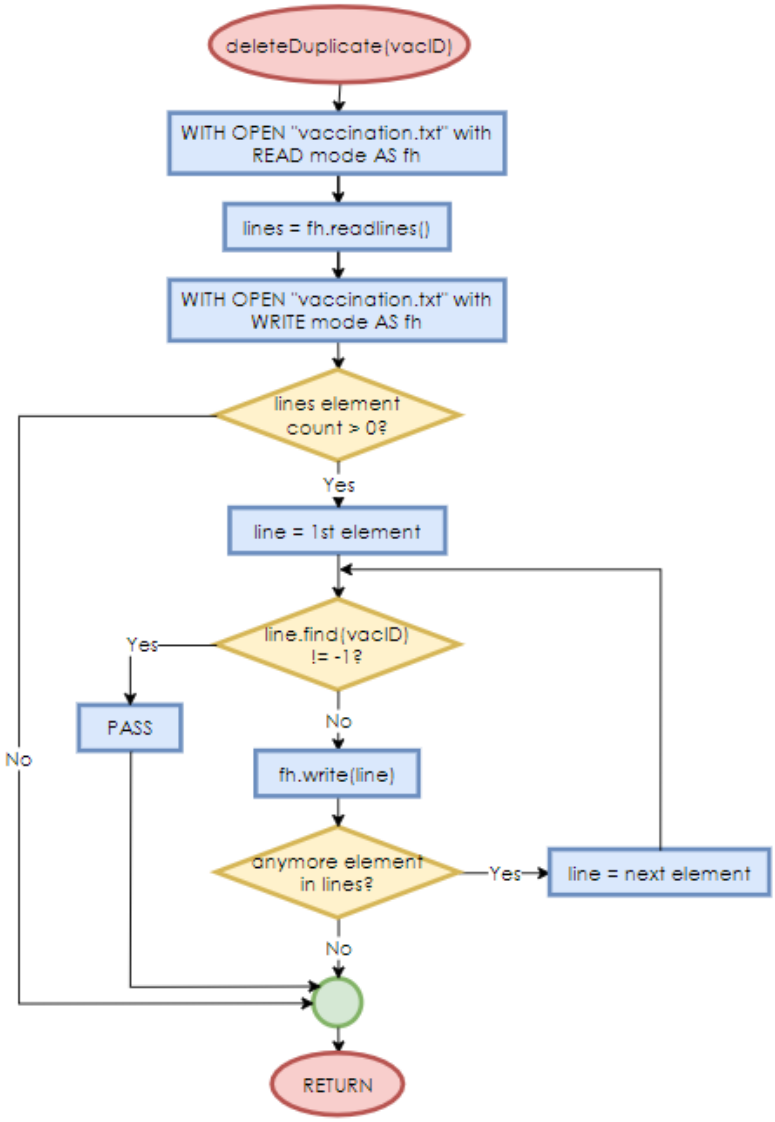


vaccinationStatusAndIntervalBetweenDoses(vaccine)

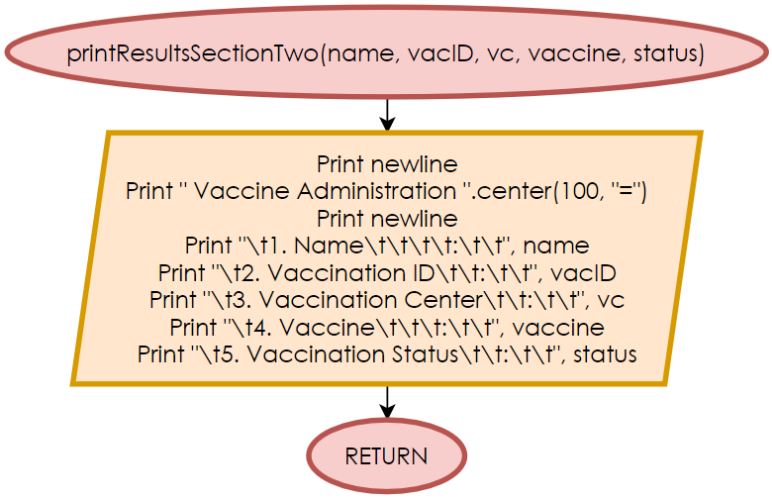



B
Print "Have you received your first vaccination dose? [Y / N]"
Print "Enter 'X' to quit: "
Read choice.upper()
choice == "X"?
Yes
sys.exit()
A
No
choice == "Y"?
Yes
Print "Have you received your second vaccination dose? [Y / N]"
Print "Enter 'X' to quit: "
Read choice2.upper()
choice2 == "X"?
Yes
sys.exit()
A
No
choice2 == "N"?
Yes
vaccine == "AF"?
Yes
intervalBetweenDoses = 2
No
vaccine == "BV" OR vaccine == "CZ"?
Yes
intervalBetweenDoses = 3
intervalBetweenDoses = 4
No
status = "COMPLETED-D1"
Print "Please come after", intervalBetweenDoses, "weeks for your second vaccination dose."
No
choice == "N"?
Yes
status = "NEW"
Print "Come back when you have received your first vaccination dose."
No
Print "Invalid input, please try again."
Print newline
valid = False
No
choice2 == "Y"?
Yes
status = "COMPLETED"
Print "You have finished your vaccination."
No
Print "Invalid input, please try again."
Print newline
valid = False
C
D

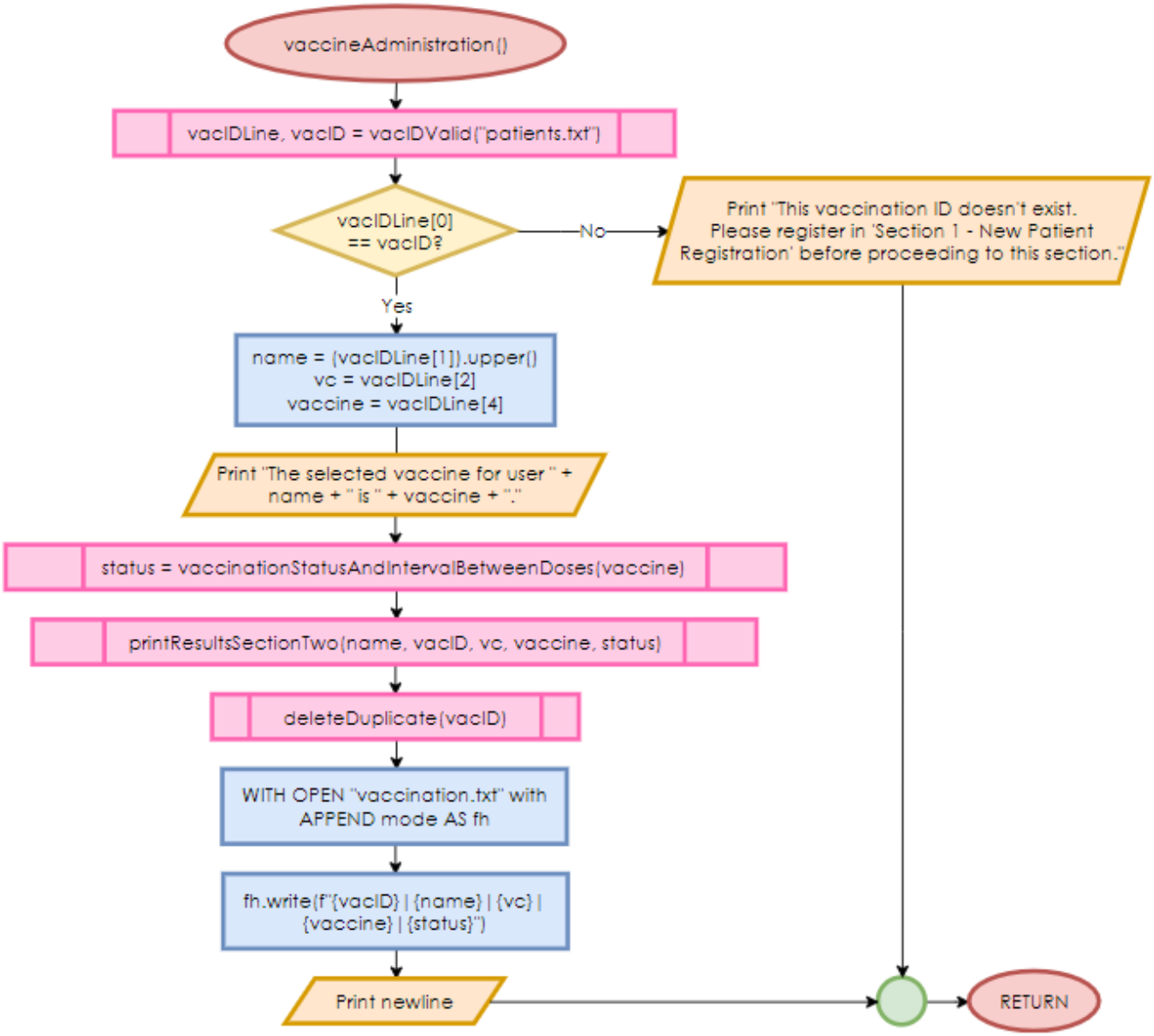
deleteDuplicate(vacID)



printResultsSectionTwo(name, vacID, vc, vaccine, status)

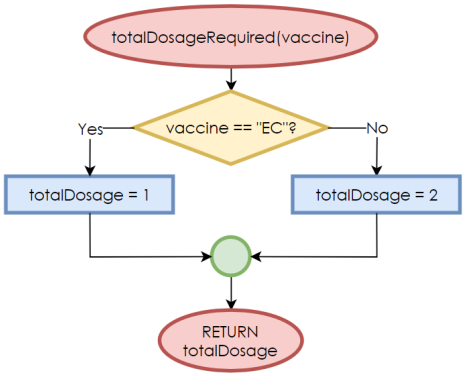


vaccineAdministration()

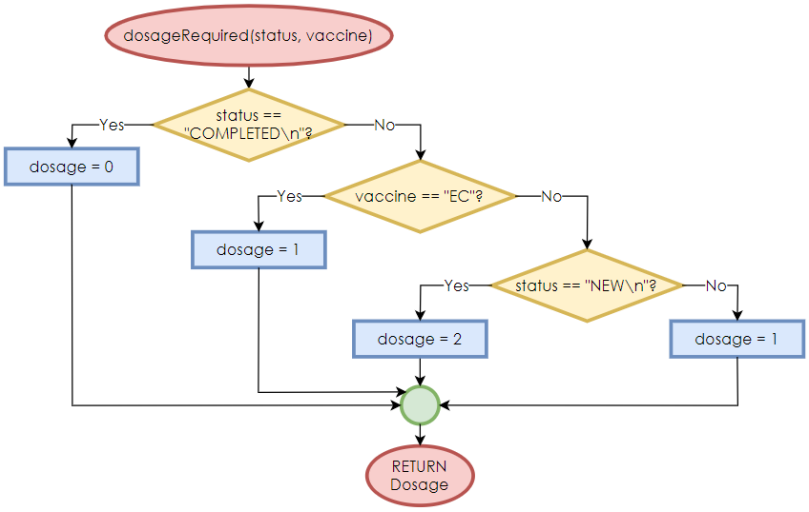


SECTION 3 FUNCTIONS

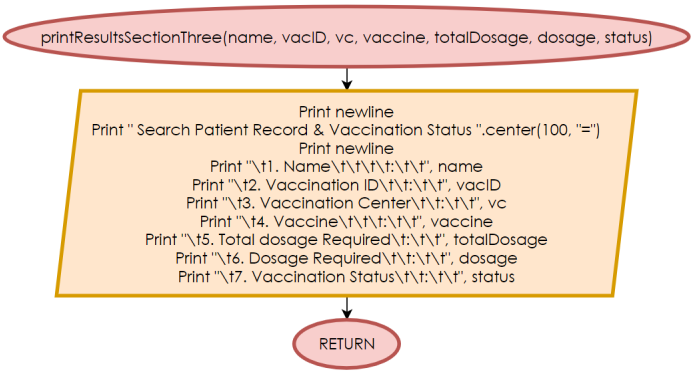
totalDosageRequired(vaccine)



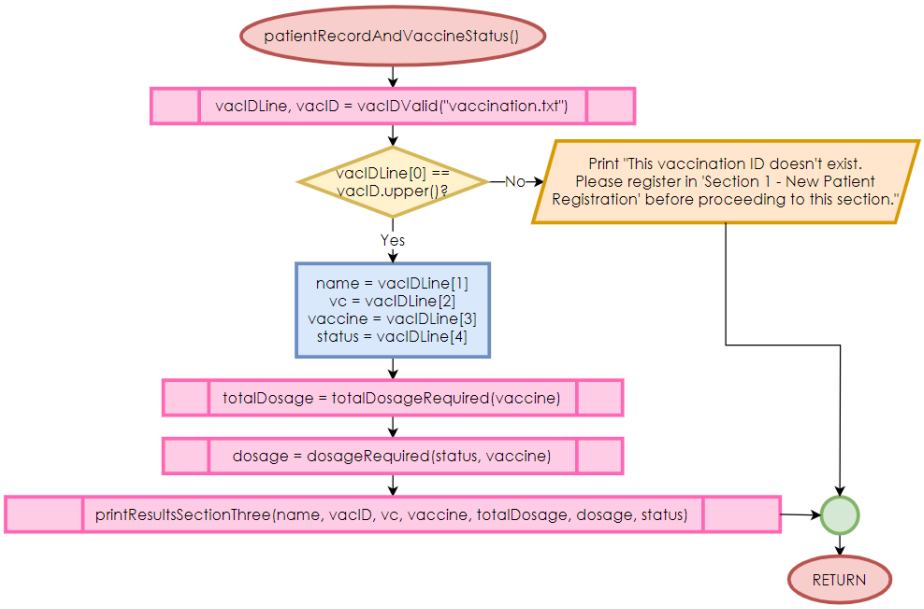
dosageRequired(status, vaccine)



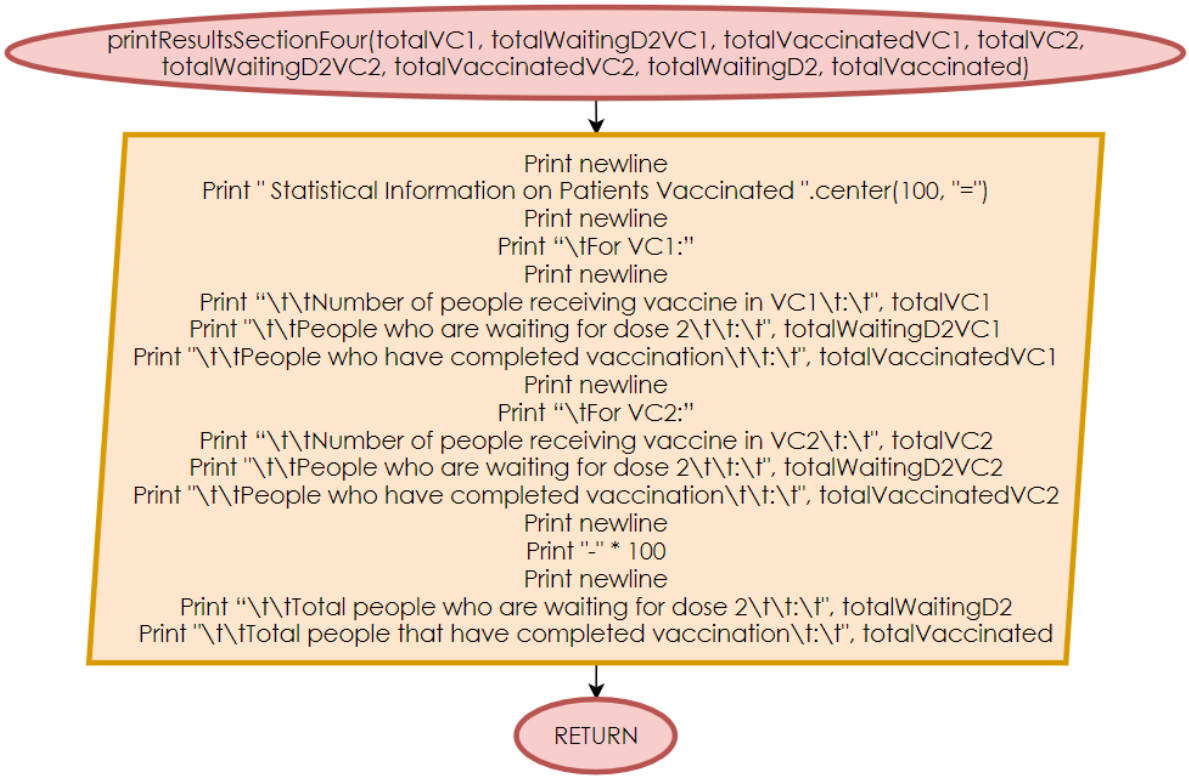
printResultsSectionThree(name, vacID, vc, vaccine, totalDosage, dosage, status)



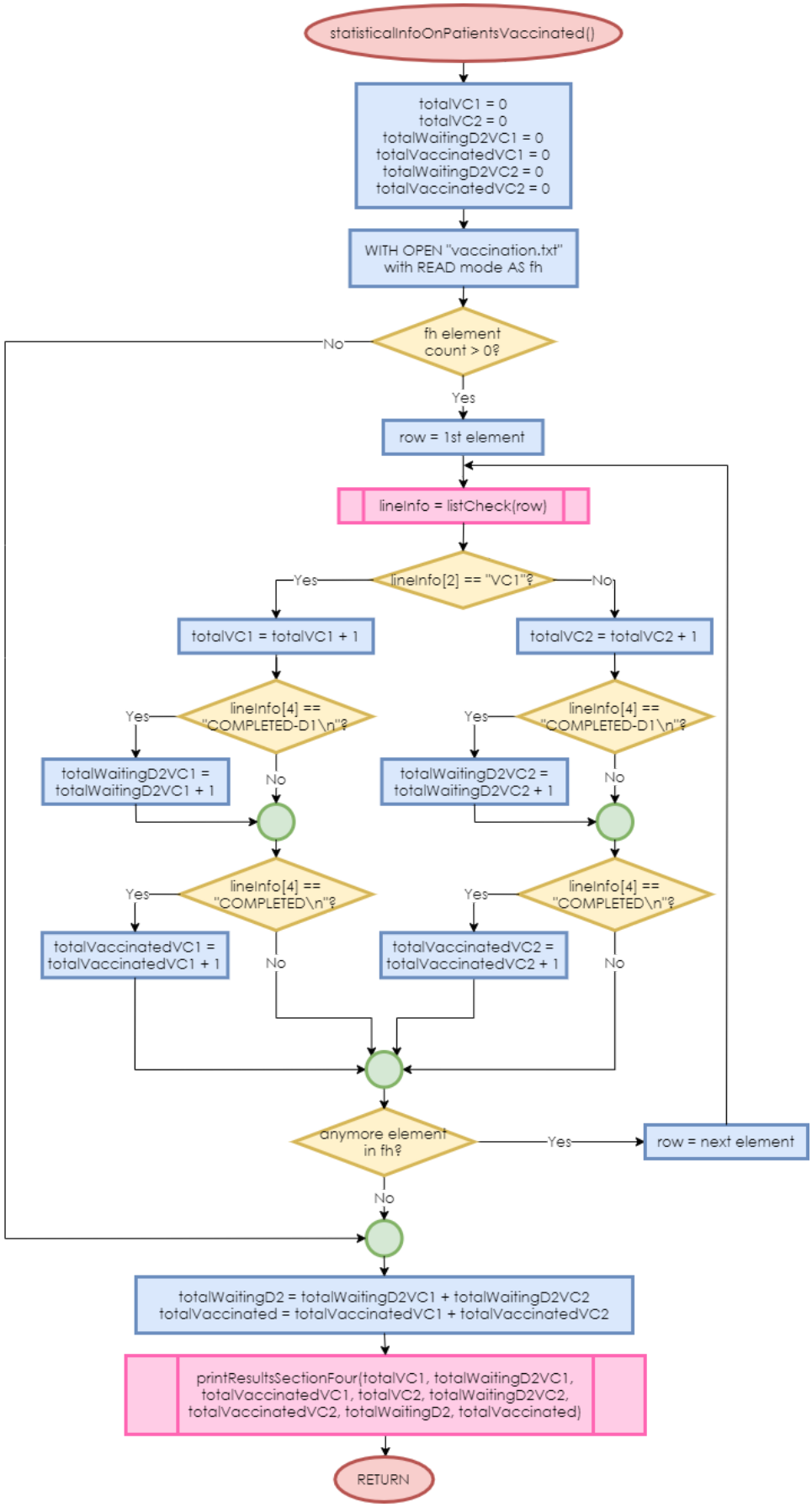
patientRecordAndVaccineStatus()



printResultsSectionFour(totalVC1, totalWaitingD2VC1, totalVaccinatedVC1, totalVC2, totalWaitingD2VC2, totalVaccinatedVC2, totalWaitingD2, totalVaccinated)

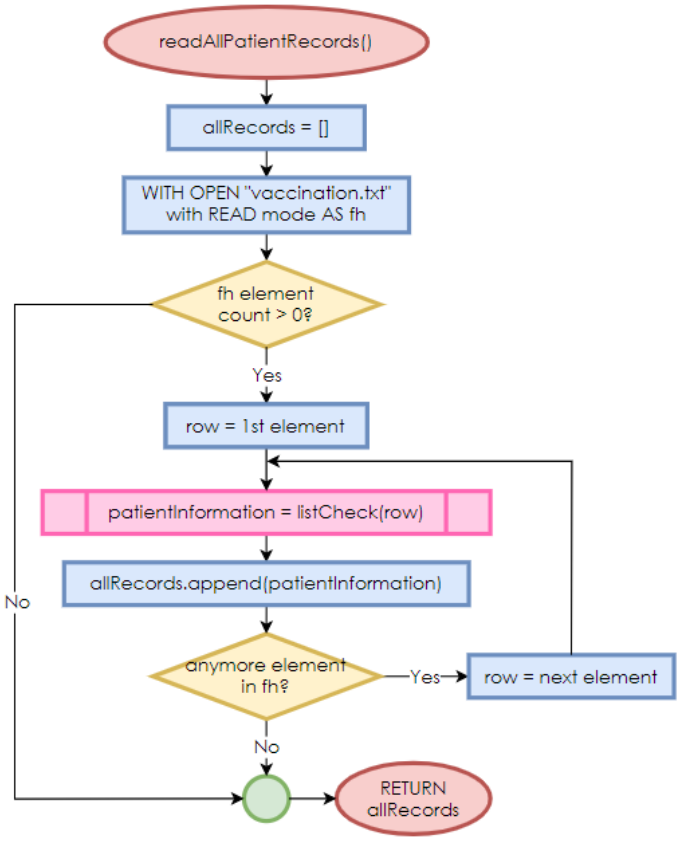


statisticalInfoOnPatientsVaccinated()

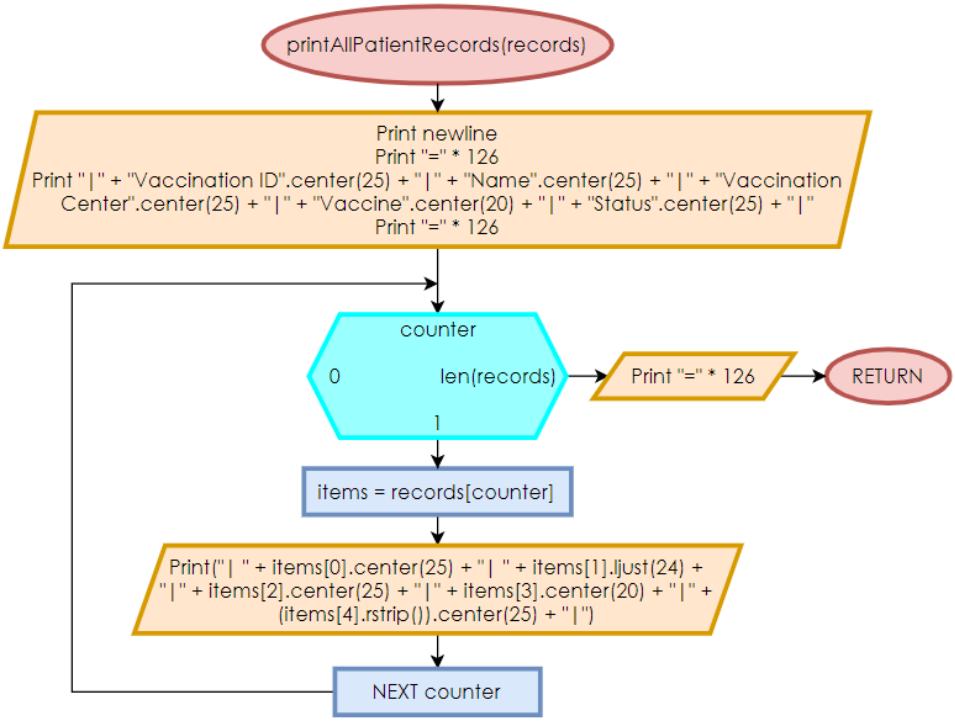


SECTION 5 FUNCTIONS

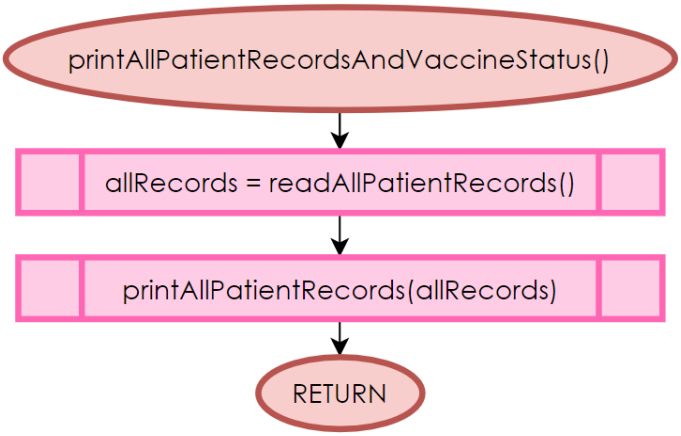
readAllPatientRecords()



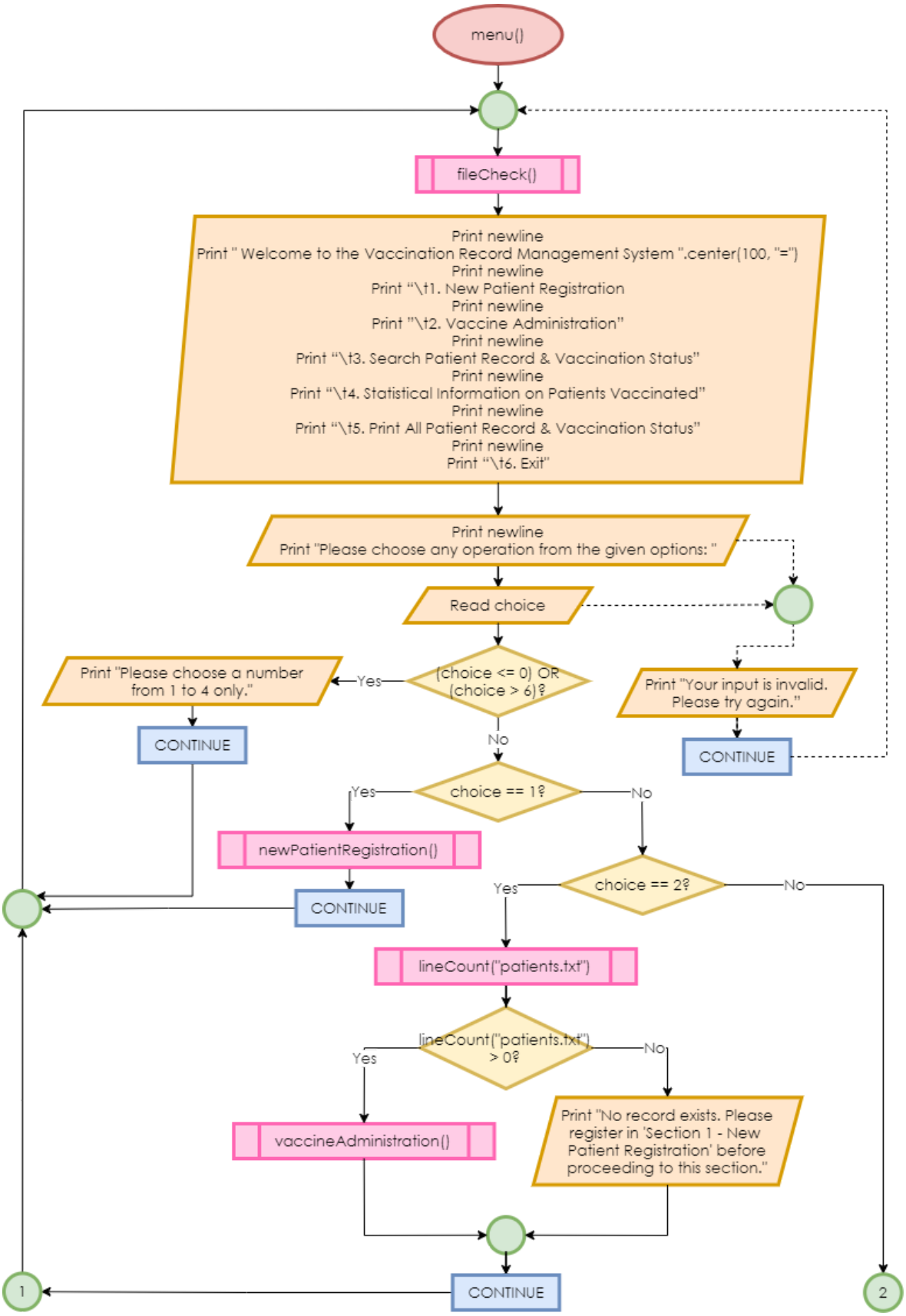
printAllPatientRecords(records)

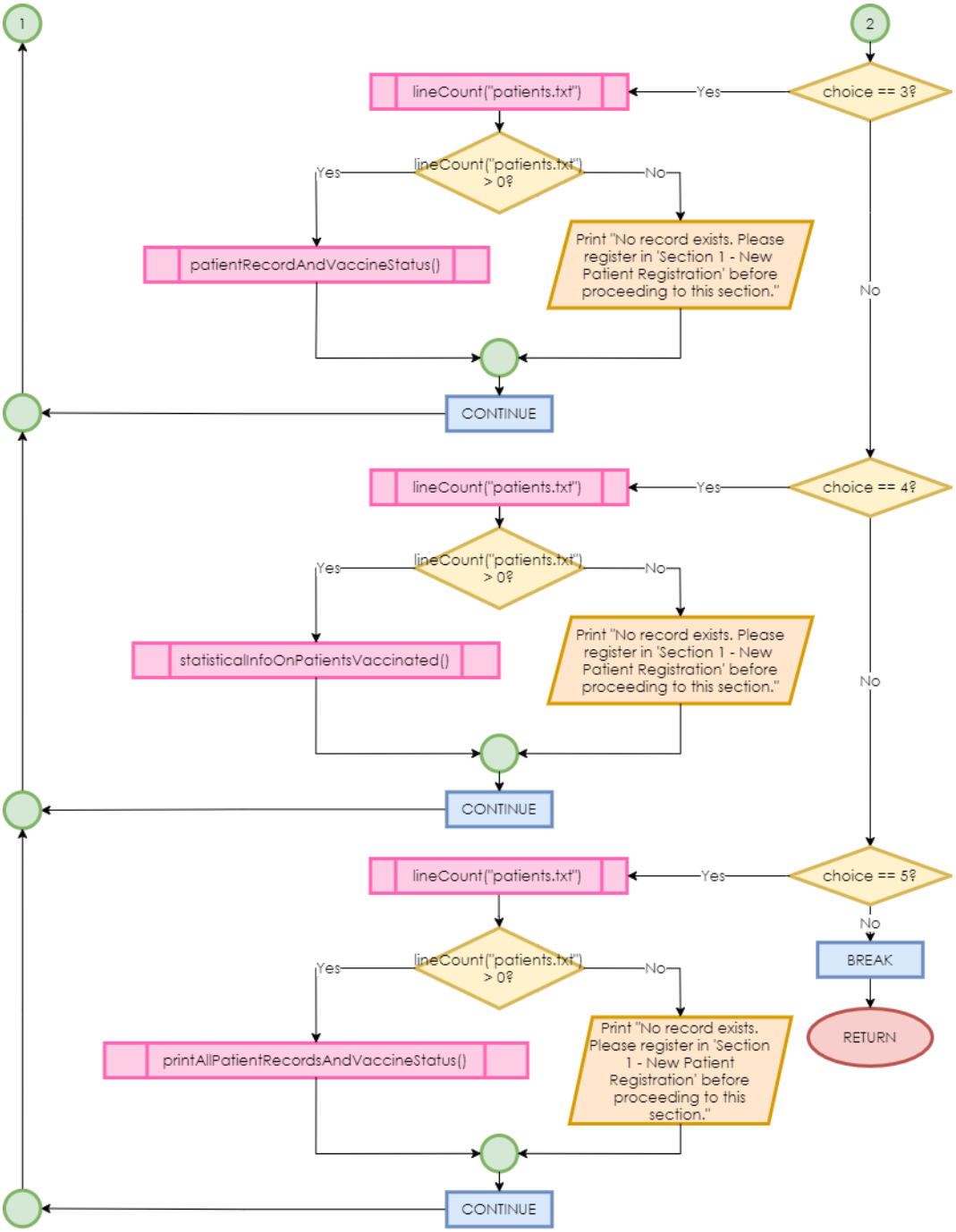


printAllPatientRecordsAndVaccineStatus()

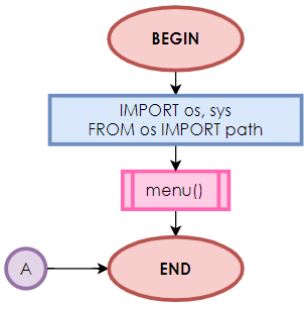


menu()


menu()
fileCheck()
Print newline
Print " Welcome to the Vaccination Record Management System ".center(100, "=")
Print newline
Print "\t1. New Patient Registration
Print newline
Print "\t2. Vaccine Administration"
Print newline
Print "\t3. Search Patient Record & Vaccination Status"
Print newline
Print "\t4. Statistical Information on Patients Vaccinated"
Print newline
Print "\t5. Print All Patient Record & Vaccination Status"
Print newline
Print "\t6. Exit"
Print newline
Print "Please choose any operation from the given options: "
Read choice
(choice <= 0) OR (choice > 6)?
Yes
Print "Please choose a number from 1 to 4 only."
CONTINUE
Print "Your input is invalid. Please try again."
CONTINUE
No
choice == 1?
Yes
No
newPatientRegistration()
CONTINUE
choice == 2?
Yes
No
lineCount("patients.txt")
lineCount("patients.txt") > 0?
Yes
No
vaccineAdministration()
Print "No record exists. Please register in 'Section 1 - New Patient Registration' before proceeding to this section."
CONTINUE
1
2


1
2
choice == 3?
Yes
lineCount("patients.txt")
lineCount("patients.txt") > 0?
Yes
No
patientRecordAndVaccineStatus()
Print "No record exists. Please register in 'Section 1 - New Patient Registration' before proceeding to this section."
No
CONTINUE
choice == 4?
Yes
lineCount("patients.txt")
lineCount("patients.txt") > 0?
Yes
No
statisticalInfoOnPatientsVaccinated()
Print "No record exists. Please register in 'Section 1 - New Patient Registration' before proceeding to this section."
No
CONTINUE
choice == 5?
Yes
lineCount("patients.txt")
lineCount("patients.txt") > 0?
Yes
No
printAllPatientRecordsAndVaccineStatus()
Print "No record exists. Please register in 'Section 1 - New Patient Registration' before proceeding to this section."
No
BREAK
RETURN
CONTINUE

MAIN CODE


BEGIN
IMPORT os, sys
FROM os IMPORT path
menu()
A
END

PROGRAM SOURCE CODE AND SAMPLE INPUT/OUTPUT

Note:

- "CF: " means a function is called for the purpose mentioned.
- "END:" indicates the input that ends the program.

==========

GENERAL FUNCTIONS

Functions that are used in other sections.

==========

fileCheck()

```
6   # Generate the necessary files
7   def fileCheck():
8       # Return True if "patients.txt" is an existing regular file
9       if os.path.isfile("patients.txt") == False:
10          fh = open("patients.txt", "w")
11          fh.close()
12
13      # Return True if "vaccination.txt" is an existing regular file
14      if os.path.isfile("vaccination.txt") == False:
15          fh = open("vaccination.txt", "w")
16          fh.close()
```

To generate "patients.txt" and "vaccination.txt" if they don't exist when the code runs.

---

listCheck(string)

```
18  # To convert string in file to list
19  def listCheck(string):
20      separate = list(string.split("|"))
21      return separate
```

To convert a string that is separated by "|" into a list.

---

characterCheck(string)

```
23  # To convert string in list to character
24  def characterCheck(string):
25      separate = list(string.split("-"))
26      return separate
```

Same with characterCheck(string) but separated by "-".

lineCount(fileName)

```
28  # To count how many lines are in the file
29  def lineCount(fileName):
30      fh = open(fileName, "r")
31      lineCount = 0
32      for line in fh:
33          if line != "\n":
34              lineCount += 1
35      fh.close()
36      return lineCount
```

To count the amount of lines excluding empty lines in the file.

---

integerValidation(lineShown)

```
38  # Integer validations
39  def integerValidation(lineShown):
40      while True:
41          try:
42              item = int(input(lineShown))
43          except:
44              print("Your input is invalid. Please try again by entering integers.\n")
45              continue
46          if item == -1:
47              sys.exit()
48          else:
49              return item
```

- To validate input as an integer.
- Receive a string as an argument to be shown as a query.
- Lets the user reenter his input if anything other than integers is inputted.
- END: -1

floatValidation(lineShown)

```
51  # Float validation
52  def floatValidation(lineShown):
53      while True:
54          try:
55              item = float(input(lineShown))
56              decimal = round(item, 2)
57          except:
58              print("Your input is invalid. Please try again by entering integers or floats.\n")
59              continue
60          if item == -1:
61              sys.exit()
62          else:
63              return decimal
```

Same with integerValidation(lineShown) but validates input as float.

---

vacIDValid(fileUsed)

```
65  # Checks if the user inputted vaccination ID exists
66  def vacIDValid(fileUsed):
67      vacID = str(input("Please insert your vaccination ID (Enter 'X' to quit): ")).upper()
68      if vacID == "X":
69          sys.exit()
70      else:
71          with open(fileUsed,"r") as fh:
72              for row in fh:
73                  # Split the string into a list
74                  vacIDLine = listCheck(str(row))
75                  if vacIDLine[0] == vacID:
76                      break
77      return vacIDLine, vacID
```

- To check if the user inputted vaccination ID exists in the file.
- Compare the vaccination ID with the user input by calling a function to split the string in the file into a list and compare the first element of the list.
- END: X

==========

SECTION 1 FUNCTIONS

Functions that are only used in Section 1 - New Patient Registration.

==========

vaccinationCenterSelection()

```
81  # Selecting a vaccination center
82  def vaccinationCenterSelection():
83      while True:
84          print("Which vaccination centre are you going to? [ VC1 / VC2 ]")
85          vc = str(input("Enter 'X' to quit: "))
86          # Validating input to be only "VC1" or "VC2"
87          if vc.upper() == "X":
88              sys.exit()
89          elif (vc.upper() != "VC1") and (vc.upper() != "VC2") and (vc.upper() != "X"):
90              print("Please choose either VC1 or VC2 only.\n")
91              continue
92          elif vc.upper() == "VC1":
93              vc = "VC1"
94          else:
95              vc = "VC2"
96          return vc
```

```
Which vaccination centre are you going to? [ VC1 / VC2 ]
Enter 'X' to quit:
```

Output

Lets the user select a vaccination center.

```
Which vaccination centre are you going to? [ VC1 / VC2 ]
Enter 'X' to quit: Test
Please choose either VC1 or VC2 only.

Which vaccination centre are you going to? [ VC1 / VC2 ]
Enter 'X' to quit:
```

Output - Input validation

Lets the user reenter his input if he inputs anything other than “VC1” or “VC2”.

END: X

nameCheck(fileUsed)

```python
# To check if the name exists in the file chosen
def nameCheck(fileUsed):
    while True:
        valid = True
        name = str(input("Please insert your name (Enter 'X' to quit): ")).upper()
        if name == "X":
            sys.exit()
        elif len(name) > 23:
            print("Please input a name within 23 characters.\n")
        else:
            with open(fileUsed,"r") as fh:
                # Reads all lines as string elements in a list to be compared later
                info = fh.readlines()
                if lineCount(fileUsed) > 0:
                    for row in info:
                        nameLine = listCheck(str(row))
                        # Check if the second section of the line in the "patients.txt" file matches the user input
                        if nameLine[1].upper() == name.upper():
                            print("A record with this name already exists.\n")
                            valid = False
                    if valid == True:
                        return name
                else:
                    return name
```

```
Please insert your name (Enter 'X' to quit):
```

Output

Lets the user enter his name and checks if the name exists in the file.

```
Please insert your name (Enter 'X' to quit): Test Test Test Test Test
Please input a name within 23 characters.

Please insert your name (Enter 'X' to quit): Hu Tao
A record with this name already exists.

Please insert your name (Enter 'X' to quit):
```

Output - Input validation

- Lets the user reenter his input if the name:
    - Exceeds 23 characters
    - Exists in the file

- Compare the name with the user input by calling a function to split a row in the file into a list and compare the second element of the list.
- END: X

vaccineSelection(listUsed)

```python
# Letting the user pick a vaccine from a list
def vaccineSelection(listUsed):
    print("You are elligble for these vaccines:", listUsed)
    while True:
        vaccine = str(input("Please choose only one vaccine from the list above (Enter 'X' to exit): ")).upper()
        if vaccine == "X":
            sys.exit()
        for vac in listUsed:
            # Matches input with the items in the list
            if vac.upper() == vaccine.upper():
                print("You have chosen " + vaccine + ".")
                return vaccine
            else:
                continue
```

```
You are elligble for these vaccines: ['AF', 'DM', 'BV', 'EC', 'CZ']
Please choose only one vaccine from the list above (Enter 'X' to exit): DM
You have chosen DM.
```

Sample Output

Lets the user pick a vaccine from a list.

```
You are elligble for these vaccines: ['AF', 'DM', 'BV', 'EC', 'CZ']
Please choose only one vaccine from the list above (Enter 'X' to exit): Test
Please choose only one vaccine from the list above (Enter 'X' to exit):
```

Output - Input validation

- Compare the vaccines in the list with the user input.
- Lets the user reenter his input if it does not match with the vaccines in the list.

- END: X

uniqueIDGenerator(vc)

```
138  # Create a unique ID: Check the current IDs in the file and add 1 to the last existant ID
139  def uniqueIDGenerator(vc):
140      vacID = 0
141      with open("patients.txt","r") as fh:
142          if lineCount("patients.txt") > 0:
143              lines = fh.readlines()
144              lastLine = lines[-1:]
145              # Split the string into a list
146              vacIDLine = listCheck(str(lastLine))
147              # Take the first section (vacID)
148              vacIDSection = characterCheck(str(vacIDLine[0]))
149              # Increase second section  of vacID by 1 to generate new sequenced vacID
150              increment = int(vacIDSection[1]) + 1
151              vacID = vc + "-" + str(increment)
152          else:
153              vacID = vc + "-1"
154
155      return vacID
```

Generates a sequential ID for the user by checking the latest existing ID and add 1 to the ID.

VC2-18
VC1-19
Sample Vaccination ID

ID Syntax: Selected VC-Number

- If there are no lines in “patients.txt” file:
    - “(Number)” section of ID will be 1.
- If there are line(s) in “patients.txt” file:
    - The last line of “patients.txt” will be selected.
    - The first element of the line will be split into “(Selected VC)-” section and “(Number)” section.
    - The “(Number)” section will increase by 1.

extraInformation()

```
157  # Extra information
158  def extraInformation():
159      # Height
160      height = floatValidation("Please input your height in cm (Enter '-1' to quit): ")
161      h = "Height:" + str(height) + "cm"
162
163      # Weight
164      weight = floatValidation("Please input your weight in kg (Enter '-1' to quit): ")
165      w = "Weight:" + str(weight) + "kg"
166
167      return h, w
```

Lets the user input his height and weight.

```
Please input your height in cm (Enter '-1' to quit): Test
Your input is invalid. Please try again by entering integers or floats.

Please input your height in cm (Enter '-1' to quit): 157.3
Please input your weight in kg (Enter '-1' to quit): Test
Your input is invalid. Please try again by entering integers or floats.

Please input your weight in kg (Enter '-1' to quit): 45.7
```

Output - Input validation

CF: Validate input as float.

END: -1

---

printResultsSectionOne(name, vacID)

```
169  # Print Section 1 results
170  def printResultsSectionOne(name, vacID):
171      print("\n", " Registration Details ".center(100, "="))
172      print("\n", "You have successfully registered.".center(100, " "))
173      print("\n\tUser:", name)
174      print("\tYour vaccination ID is " + vacID + ".")
175      print("\nDo remember to head to 'Section 2 - Vaccine Administration' after taking your first vaccination dose!")
```

```
==================================== Registration Details ====================================

                                You have successfully registered.

        User: CHANG SHIAU HUEI
        Your vaccination ID is VC1-19.

Do remember to head to 'Section 2 - Vaccine Administration' after taking your first vaccination dose!
```

Sample Output

- Print results for Section 1.

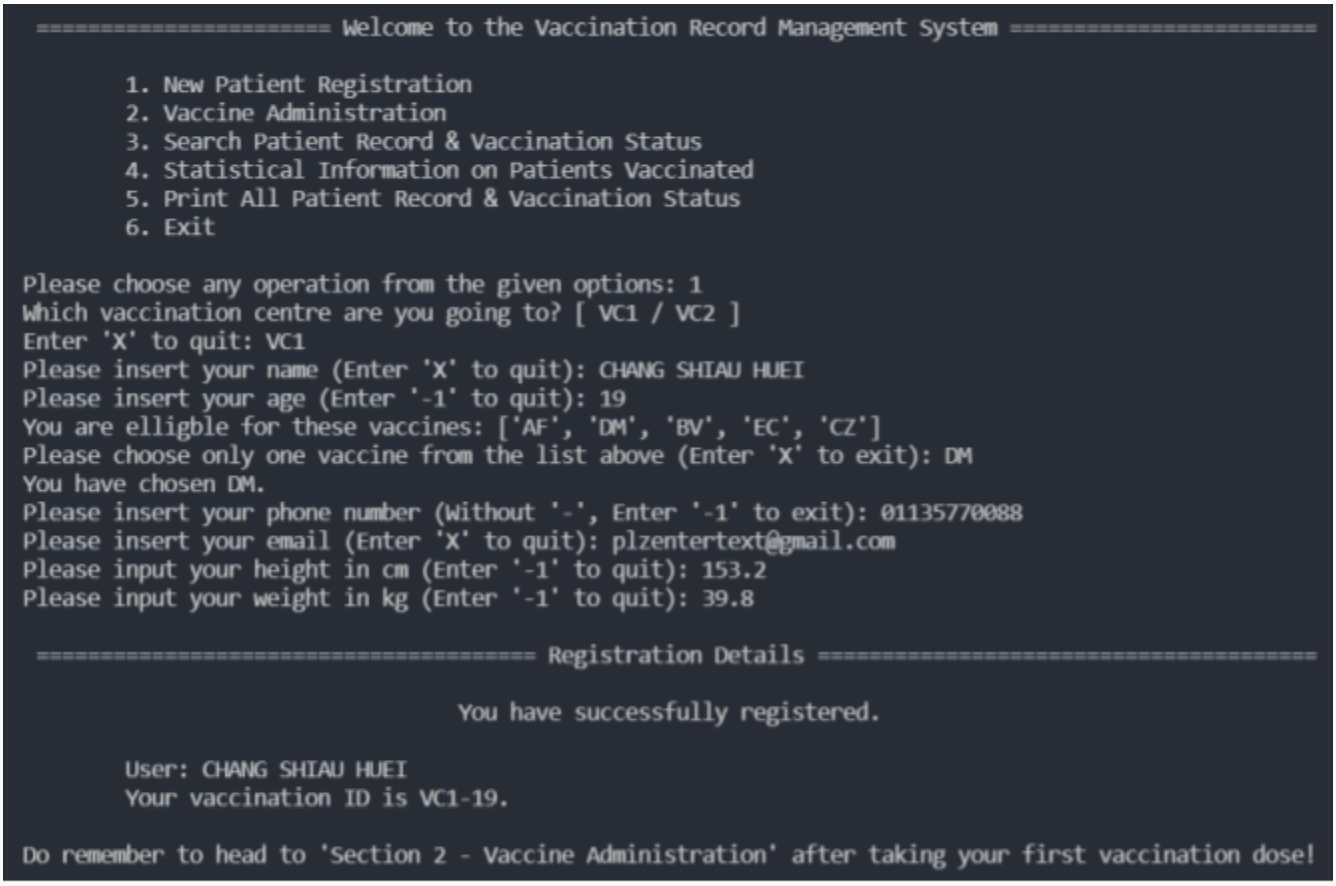
- Shows the user his vaccination ID and related information.

---

newPatientRegistration()

```python
# Section 1 - New Patient Registration
def newPatientRegistration():
    while True:
        # Selecting a vaccination center
        vc = vaccinationCenterSelection()

        # To check if the name exists in the "patients.txt" file
        name = (nameCheck("patients.txt")).upper()

        # Age validation
        age = integerValidation("Please insert your age (Enter '-1' to quit): ")

        # Check which vaccine is elligible according to age and put them in a list called elligibleVac
        # Letting the user pick a vaccine from a list
        elligibleVac = []
        # Inserting elligible vaccines into a list (elligibleVac)
        if age >= 12:
            elligibleVac.append("AF")
            elligibleVac.append("DM")
            if age >= 18:
                elligibleVac.append("BV")
                elligibleVac.append("EC")
            if age <= 45:
                elligibleVac.append("CZ")
        else:
            print("You are not elligible for any vaccine.")
            break

        # Letting the user pick a vaccine from a list called elligibleVac
        vaccine = vaccineSelection(elligibleVac)

        # Validation for phoneNum
        phoneNum = integerValidation("Please insert your phone number (Without '-', Enter '-1' to exit): ")
        phoneNum = "0" + str(phoneNum)

        # Inserting email
        email = str(input("Please insert your email (Enter 'X' to quit): "))
        if email.upper() == "X":
            sys.exit()

        # To create a unique vaccination ID
        vacID = uniqueIDGenerator(vc)

        # Extra information
        height, weight = extraInformation()

        # Print results
        printResultsSectionOne(name, vacID)

        # To put the information in "patients.txt"
        with open ("patients.txt","a") as fh:
            fh.write(f"{vacID}|{name}|{vc}|{age}|{vaccine}|{phoneNum}|{email}|{height}|{weight}\n")
        with open ("vaccination.txt","a") as fh:
            fh.write(f"{vacID}|{name}|{vc}|{vaccine}|NEW\n")
        break
```

The main function to register a new patient.



Sample Output - Process

1. CF: Gets the vaccination center chosen, name and age.

DETAILS OF VACCINES ADMINISTERED IN VC1 AND VC2	
Vaccine Code	**Age Group**
AF	12 and above
BV	18 and above
CZ	12 - 45
DM	12 and above
EC	18 and above

2. Check which vaccine is eligible according to age and put them in a list.
3. CF: Lets the user pick a vaccine from the list.
4. Quits to menu if the user is not eligible for any vaccine.

```
Please insert your phone number (Without '-', Enter '-1' to exit): Test
Your input is invalid. Please try again by entering integers.

Please insert your phone number (Without '-', Enter '-1' to exit): 011-35770282
Your input is invalid. Please try again by entering integers.

Please insert your phone number (Without '-', Enter '-1' to exit): 01135770282
```

Sample Output - Input validation

```
Please insert your email (Enter 'X' to quit):
```

Output

5. User inserts phone number and email.
6. CF: Generate a vaccination ID.
7. CF: Gets height and weight information.
8. CF: Prints results for Section 1.

```
patients.txt
18  VC2-18|VENTI|VC2|65|DM|0118743078|anemoarchon@gmail.com|Height:177.8cm|Weight:54.7kg
19  VC1-19|CHANG SHIAU HUEI|VC1|19|DM|01135770088|plzentertext@gmail.com|Height:153.2cm|Weight:39.8kg
```

Sample Output

9. Write the information inputted by the user into “patients.txt” file.

```
vaccination.txt
18  VC2-18|VENTI|VC2|DM|COMPLETED
19  VC1-19|CHANG SHIAU HUEI|VC1|DM|NEW
```

Sample Output

10. Set the status of the registered user to “NEW” and write the relevant information into “vaccination.txt” file.

==========

SECTION 2 FUNCTIONS

Functions that are only used in Section 2 - Vaccine Administration.

==========

vaccinationStatusAndIntervalBetweenDoses(vaccine)

```
# Updating vaccination status and displaying the interval between doses
def vaccinationStatusAndIntervalBetweenDoses(vaccine):
    # For EC Vaccine
    while True:
        valid = True
        if vaccine == "EC":
            print("Have you received your first vaccination dose? [ Y / N ]")
            choice = str(input("Enter 'X' to quit: ")).upper()
            if choice == "X":
                sys.exit()
            elif choice == "Y":
                status = "COMPLETED"
                print("You have finished your vaccination.")
            elif choice == "N":
                status = "NEW"
                print("Come back when you have received your first vaccination dose.")
            else:
                print("Invalid input, please try again.\n")
                valid = False

        # For vaccines other than EC
        else:
            print("Have you received your first vaccination dose? [ Y / N ]")
            choice = str(input("Enter 'X' to quit: ")).upper()
            if choice == "X":
                sys.exit()
            elif choice == "Y":
                print("Have you received your second vaccination dose? [ Y / N ]")
                choice2 = str(input("Enter 'X' to quit: ")).upper()
                if choice2 == "X":
                    sys.exit()
                elif choice2 == "N":
                    if vaccine == "AF":
                        intervalBetweenDoses = 2
                    elif (vaccine == "BV") or (vaccine == "CZ"):
                        intervalBetweenDoses = 3
                    else:
                        intervalBetweenDoses = 4
                    status = "COMPLETED-D1"
                    # Informing for the next vaccination dose session
                    print("Please come after", str(intervalBetweenDoses), "weeks for your second vaccination dose.")
                elif choice2 == "Y":
                    status = "COMPLETED"
                    print("You have finished your vaccination.")
                else:
                    # Goes back to asking if the user received the first vaccination dose or not
                    print("Invalid input, please try again.\n")
                    valid = False
            elif choice == "N":
                status = "NEW"
                print("Come back when you have received your first vaccination dose.")
            else:
                print("Invalid input, please try again.\n")
                valid = False

        if valid == True:
            return status
```

DETAILS OF VACCINES ADMINISTERED IN VC1 AND VC2				
Vaccine Code	Interval Between Doses	Status		
		Before Dose 1	After Dose 1	After Dose 2
AF	2 weeks (or 14 days)	NEW	COMPLETED-D1	COMPLETED
BV	3 weeks (or 21 days)			
CZ	3 weeks (or 21 days)			
DM	4 weeks (or 28 days)			
EC			COMPLETED	

- Decides the user's vaccination status according to the type of vaccine chosen and inputs to the queries.
- Decides and informs the interval between doses according to the vaccine chosen.

```
Have you received your first vaccination dose? [ Y / N ]
Enter 'X' to quit: Test
Invalid input, please try again.

Have you received your first vaccination dose? [ Y / N ]
Enter 'X' to quit: Y
Have you received your second vaccination dose? [ Y / N ]
Enter 'X' to quit: Test
Invalid input, please try again.

Have you received your first vaccination dose? [ Y / N ]
Enter 'X' to quit: Y
Have you received your second vaccination dose? [ Y / N ]
Enter 'X' to quit: N
Please come after 4 weeks for your second vaccination dose.
```

Sample Output - Input validation

- Lets the user reenter his input if the user input is invalid. Valid inputs are "Y", "N" or "X".
- END: X

deleteDuplicate(vacID)

```
293   # Delete duplicate items in the file
294   def deleteDuplicate(vacID):
295       with open("vaccination.txt","r") as fh:
296           lines = fh.readlines()
297           # Delete matching content
298           with open("vaccination.txt", 'w') as fh:
299               for line in lines:
300                   # find() returns -1 if no match is found
301                   if line.find(vacID) != -1:
302                       pass
303                   else:
304                       fh.write(line)
```

- Rewrite records with the same vaccination ID in the "vaccination.txt" file if another record with the associated vaccination ID exists.
- If a duplicate vaccination ID is not found, line.find(vacID) returns -1 and nothing will be changed in “vaccination.txt” file.

---

printResultsSectionTwo(name, vacID, vc, vaccine, status)

```
306   # Print Section 2 results
307   def printResultsSectionTwo(name, vacID, vc, vaccine, status):
308       print("\n", " Vaccine Administration ".center(100, "="))
309       print("\n\t1. Name\t\t\t\t:\t\t", name)
310       print("\t2. Vaccination ID\t\t:\t\t", vacID)
311       print("\t3. Vaccination Center\t\t:\t\t", vc)
312       print("\t4. Vaccine\t\t\t:\t\t", vaccine)
313       print("\t5. Vaccination Status\t\t:\t\t", status)
```

```
========================================= Vaccine Administration =========================================

        1. Name                         :               CHANG SHIAU HUEI
        2. Vaccination ID               :               VC1-19
        3. Vaccination Center           :               VC1
        4. Vaccine                      :               DM
        5. Vaccination Status           :               COMPLETED-D1
```

Sample Output

- Print results for Section 2.
- Shows the user his name, vaccination ID, chosen vaccination center, chosen vaccine and vaccination status.

---

vaccineAdministration()

```python
# Section 2 - Vaccine Administration
def vaccineAdministration():
    # Comparing the user inputted vaccination ID to the ones in the "patients.txt" file to check if it exists
    vacIDLine, vacID = vacIDValid("patients.txt")

    # Check if the first section of the line in the "patients.txt" file matches the user input
    if vacIDLine[0] == vacID:

        # Getting information
        name = (vacIDLine[1]).upper()
        vc = vacIDLine[2]
        vaccine = vacIDLine[4]
        print("The selected vaccine for user " + name + " is " + vaccine + ".")

        # Updating vaccination status and displaying the interval between doses
        status = vaccinationStatusAndIntervalBetweenDoses(vaccine)

        # Print results
        printResultsSectionTwo(name, vacID, vc, vaccine, status)

        # To rewrite records with the same "vacID" in "vaccination.txt" file and delete the line
        deleteDuplicate(vacID)

        # Update the information in "vaccination.txt"
        with open ("vaccination.txt","a") as fh:
            fh.write(f"{vacID}|{name}|{vc}|{vaccine}|{status}\n")

    else:
        print("This vaccination ID doesn't exist. Please register in 'Section 1 - New Patient Registration' before proceeding to this section.")
```

The main function to administer the vaccine.

```
========== Welcome to the Vaccination Record Management System ==========

        1. New Patient Registration
        2. Vaccine Administration
        3. Search Patient Record & Vaccination Status
        4. Statistical Information on Patients Vaccinated
        5. Print All Patient Record & Vaccination Status
        6. Exit

Please choose any operation from the given options: 2
Please insert your vaccination ID (Enter 'X' to quit): VC1-19
The selected vaccine for user CHANG SHIAU HUEI is DM.
Have you received your first vaccination dose? [ Y / N ]
Enter 'X' to quit: Y
Have you received your second vaccination dose? [ Y / N ]
Enter 'X' to quit: N
Please come after 4 weeks for your second vaccination dose.

========== Vaccine Administration ==========

        1. Name                 :       CHANG SHIAU HUEI
        2. Vaccination ID       :       VC1-19
        3. Vaccination Center   :       VC1
        4. Vaccine              :       DM
        5. Vaccination Status   :       COMPLETED-D1
```

Sample Output - Process

```
 ==================== Welcome to the Vaccination Record Management System ====================

       1. New Patient Registration
       2. Vaccine Administration
       3. Search Patient Record & Vaccination Status
       4. Statistical Information on Patients Vaccinated
       5. Print All Patient Record & Vaccination Status
       6. Exit

Please choose any operation from the given options: 2
Please insert your vaccination ID (Enter 'X' to quit): VC2-31
This vaccination ID doesn't exist. Please register in 'Section 1 - New Patient Registration' before proceeding to this section.

 ==================== Welcome to the Vaccination Record Management System ====================

       1. New Patient Registration
       2. Vaccine Administration
       3. Search Patient Record & Vaccination Status
       4. Statistical Information on Patients Vaccinated
       5. Print All Patient Record & Vaccination Status
       6. Exit

Please choose any operation from the given options:
```

Sample Output - Input validation

1. CF: Gets vaccination ID and compares it in "patients.txt" file to check if it exists.
2. If the vaccination ID does not exist, inform the user that he has to register in Section 1 beforehand, and exits to menu.
3. Gets the corresponding name, chosen vaccination center and chosen vaccine associated with the vaccination ID.
4. CF: Gets the vaccination status and interval between doses.
5. CF: Prints results for Section 2.
6. Rewrite records with the same vaccination ID in the "vaccination.txt" file if another record with the associated vaccination ID exists.

vaccination.txt

```
18    VC2-18|VENTI|VC2|DM|COMPLETED
19    VC1-19|CHANG SHIAU HUEI|VC1|DM|COMPLETED-D1
```

Sample Output

7. Write relevant information and vaccination status into “vaccination.txt” file.

==========

SECTION 3 FUNCTIONS

Functions that are only used in Section 3 - Search Patient Record & Vaccination Status.

==========

totalDosageRequired(vaccine)

```
347   # Total dosage required
348   def totalDosageRequired(vaccine):
349       if vaccine == "EC":
350           totalDosage = 1
351       else:
352           totalDosage = 2
353       return totalDosage
```

DETAILS OF VACCINES ADMINISTERED IN VC1 AND VC2	
Vaccine Code	**Dosage Required**
AF	2
BV	2
CZ	2
DM	2
EC	1

Determines the total dosage required for the user according to the type of vaccine chosen.

---

dosageRequired(status, vaccine)

```
355   # Dosage required
356   def dosageRequired(status, vaccine):
357       if status == "COMPLETED\n":
358           dosage = 0
359       else:
360           if vaccine == "EC":
361               dosage = 1
362           else:
363               if status == "NEW\n":
364                   dosage = 2
365               else:
366                   dosage = 1
367       return dosage
```

Determines the current required dosage for the user according to the type of vaccine chosen and vaccination status.

printResultsSectionThree(name, vacID, vc, vaccine, totalDosage, dosage, status)

```
369   # Print Section 3 results
370   def printResultsSectionThree(name, vacID, vc, vaccine, totalDosage, dosage, status):
371       print("\n", " Search Patient Record & Vaccination Status ".center(100, "="))
372       print("\n\t1. Name\t\t\t\t:\t\t", name)
373       print("\t2. Vaccination ID\t\t:\t\t", vacID)
374       print("\t3. Vaccination Center\t\t:\t\t", vc)
375       print("\t4. Vaccine\t\t\t:\t\t", vaccine)
376       print("\t5. Total dosage Required\t:\t\t", str(totalDosage))
377       print("\t6. Dosage Required\t\t:\t\t", str(dosage))
378       print("\t7. Vaccination Status\t\t:\t\t", status)
```

```
============================ Search Patient Record & Vaccination Status ============================

        1. Name                         :               CHANG SHIAU HUEI
        2. Vaccination ID               :               VC1-19
        3. Vaccination Center           :               VC1
        4. Vaccine                      :               DM
        5. Total dosage Required        :               2
        6. Dosage Required              :               1
        7. Vaccination Status           :               COMPLETED-D1
```

Sample Output

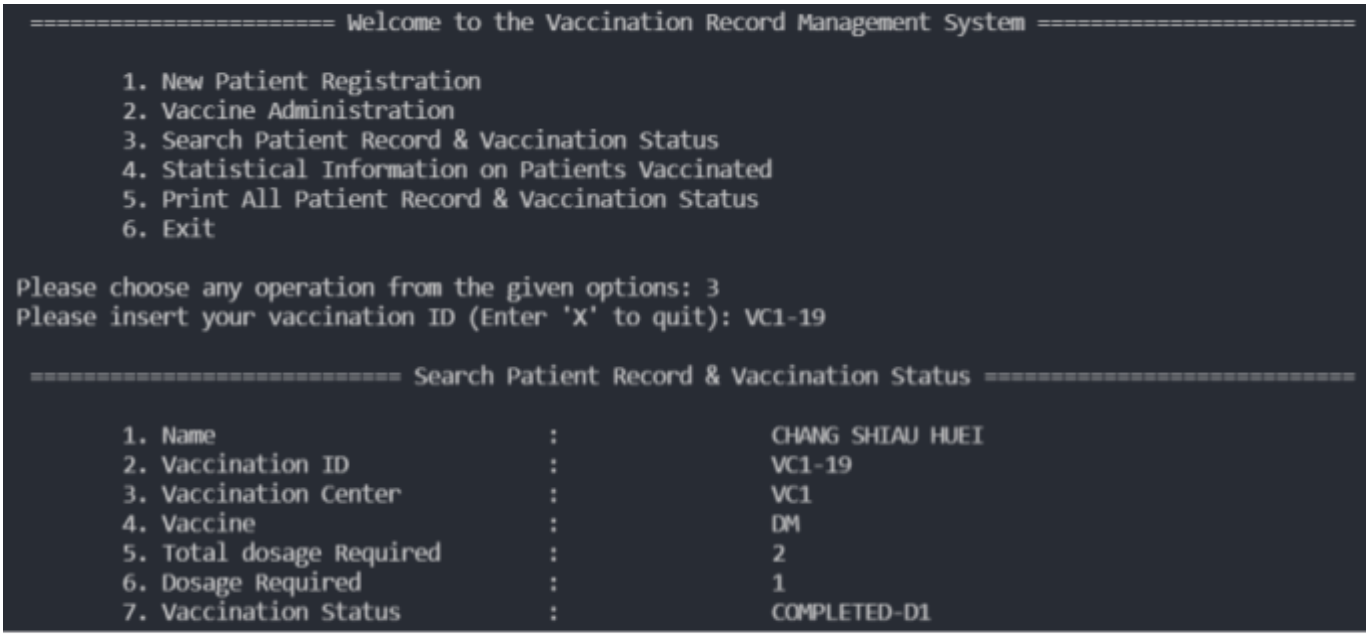
- Print results for Section 3.
- Shows the user his name, vaccination ID, chosen vaccination center, chosen vaccine total dosage required, current required dosage and vaccination status.

---

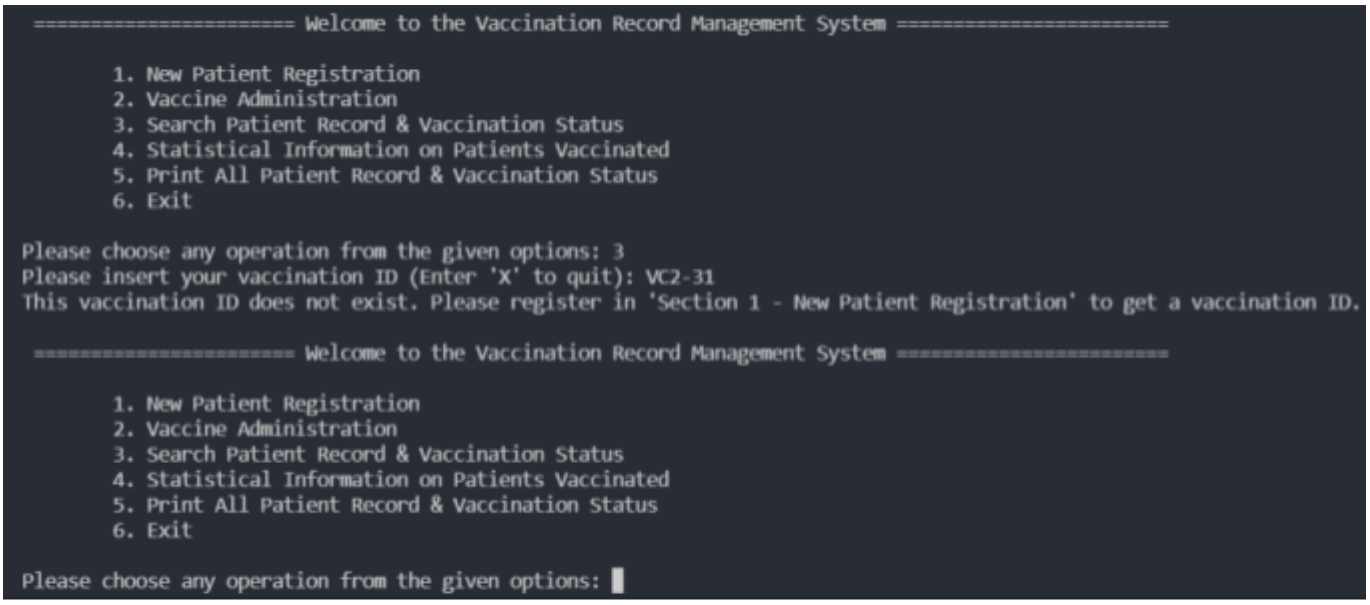
patientRecordAndVaccineStatus()

```
380   # Section 3 - Search Patient Record & Vaccination Status
381   def patientRecordAndVaccineStatus():
382       # Comparing the user inputted vaccination ID to the ones in the "vaccination.txt" file to check if it exists
383       vacIDLine, vacID = vacIDValid("vaccination.txt")
384
385       # Check if the first section of the line in the "vaccination.txt" file matches the user input
386       if vacIDLine[0] == vacID.upper():
387           # Getting VC information
388           name = vacIDLine[1]
389           vc = vacIDLine[2]
390           vaccine = vacIDLine[3]
391           status = vacIDLine[4]
392
393           # Total dosage required information
394           totalDosage = totalDosageRequired(vaccine)
395
396           # Dosage required information
397           dosage = dosageRequired(status, vaccine)
398
399           # Print results
400           printResultsSectionThree(name, vacID, vc, vaccine, totalDosage, dosage, status)
401
402       else:
403           print("This vaccination ID does not exist. Please register in 'Section 1 - New Patient Registration' to get a vaccination ID.")
```

The main function to display the patient record and vaccination status.



Sample Output - Process



Sample Output - Input validation

1. CF: Gets vaccination ID and compares it in "vaccination.txt" file to check if it exists.
2. If the vaccination ID does not exist, inform the user that he has to register in Section 1 beforehand, and exits to menu.
3. Gets the corresponding name, chosen vaccination center, chosen vaccine and vaccination status associated with the vaccination ID.
4. CF: Determine the total and current dosage required.
5. CF: Print results for Section 3.

==========

SECTION 4 FUNCTIONS

Functions that are only used in Section 4 - Statistical Information on Patients Vaccinated.

==========

printResultsSectionFour(totalVC1, totalWaitingD2VC1, totalVaccinatedVC1, totalVC2, totalWaitingD2VC2, totalVaccinatedVC2, totalWaitingD2, totalVaccinated)

```
407   # Print Section 4 results
408   def printResultsSectionFour(totalVC1, totalWaitingD2VC1, totalVaccinatedVC1, totalVC2, totalWaitingD2VC2, totalVaccinatedVC2, totalWaitingD2, totalVaccinated):
```

```
409       print("\n", " Statistical Information on Patients Vaccinated ".center(100, "="))
410       print("\n\tFor VC1:\n\t\tNumber of people receiving vaccine in VC1\t:\t", str(totalVC1))
411       print("\t\tPeople who are waiting for dose 2\t\t:\t", str(totalWaitingD2VC1))
412       print("\t\tPeople who have completed vaccination\t\t:\t", str(totalVaccinatedVC1))
413       print("\n\tFor VC2:\n\t\tNumber of people receiving vaccine in VC2\t:\t", str(totalVC2))
414       print("\t\tPeople who are waiting for dose 2\t\t:\t", str(totalWaitingD2VC2))
415       print("\t\tPeople who have completed vaccination\t\t:\t", str(totalVaccinatedVC2), "\n", "-" * 100)
416       print("\n\t\tTotal people who are waiting for dose 2\t\t:\t", str(totalWaitingD2))
417       print("\t\tTotal people that have completed vaccination\t:\t", str(totalVaccinated))
```

```
========================= Statistical Information on Patients Vaccinated =========================

        For VC1:
                Number of people receiving vaccine in VC1       :       10
                People who are waiting for dose 2               :       5
                People who have completed vaccination           :       4

        For VC2:
                Number of people receiving vaccine in VC2       :       9
                People who are waiting for dose 2               :       3
                People who have completed vaccination           :       2
----------------------------------------------------------------------------------------------------

                Total people who are waiting for dose 2         :       8
                Total people that have completed vaccination    :       6
```

Sample Output

- Print results for Section 4.
- Shows the user:
    - Number of people receiving vaccine in VC1 and VC2 separately
    - People who are waiting for dose 2 in VC1 and VC2 separately and total
    - People who have completed vaccination in VC1 and VC2 separately and total

statisticalInfoOnPatientsVaccinated()

```python
# Section 4 - Statistical Information on Patients Vaccinated
def statisticalInfoOnPatientsVaccinated():
    # Reset values
    totalVC1 = 0
    totalVC2 = 0
    totalWaitingD2VC1 = 0
    totalVaccinatedVC1 = 0
    totalWaitingD2VC2 = 0
    totalVaccinatedVC2 = 0

    # Opening the file for calculation
    with open("vaccination.txt","r") as fh:
        for row in fh:
            # Split the string into a list
            lineInfo = listCheck(str(row))
            # To count the people waiting for D2 and people who already vaccinated in VC1 and VC2 separately
            if lineInfo[2] == "VC1":
                totalVC1 += 1
                if lineInfo[4] == "COMPLETED-D1\n":
                    totalWaitingD2VC1 += 1
                if lineInfo[4] == "COMPLETED\n":
                    totalVaccinatedVC1 += 1
            else:
                totalVC2 += 1
                if lineInfo[4] == "COMPLETED-D1\n":
                    totalWaitingD2VC2 += 1
                if lineInfo[4] == "COMPLETED\n":
                    totalVaccinatedVC2 += 1

    # Get the total from both VCs
    totalWaitingD2 = totalWaitingD2VC1 + totalWaitingD2VC2
    totalVaccinated = totalVaccinatedVC1 + totalVaccinatedVC2

    # Print results
    printResultsSectionFour(totalVC1, totalWaitingD2VC1, totalVaccinatedVC1, totalVC2, totalWaitingD2VC2, totalVaccinatedVC2, totalWaitingD2, totalVaccinated)
```

The main function to display the statistical information on patients vaccinated.

```
========== Welcome to the Vaccination Record Management System ==========

        1. New Patient Registration
        2. Vaccine Administration
        3. Search Patient Record & Vaccination Status
        4. Statistical Information on Patients Vaccinated
        5. Print All Patient Record & Vaccination Status
        6. Exit

Please choose any operation from the given options: 4

============================== Statistical Information on Patients Vaccinated ==============================

        For VC1:
                Number of people receiving vaccine in VC1       :       10
                People who are waiting for dose 2               :       5
                People who have completed vaccination           :       4

        For VC2:
                Number of people receiving vaccine in VC2       :       9
                People who are waiting for dose 2               :       3
                People who have completed vaccination           :       2
 ------------------------------------------------------------------------------------------------------

                Total people who are waiting for dose 2         :       8
                Total people that have completed vaccination    :       6
```

Sample Output - Process

1. Calculates:
    a. Number of people receiving vaccine in VC1 and VC2 separately
    b. People who are waiting for dose 2 in VC1 and VC2 separately and total
    c. People who have completed vaccination in VC1 and VC2 separately and total
2. CF: Print results for Section 4.

==========

SECTION 5 FUNCTIONS

Functions that are only used in Section 5 - Print All Patient Records & Vaccination Status.

==========

readAllPatientRecords()

```
457  # Reading the patients' record
458  def readAllPatientRecords():
459      allRecords = []
460      with open("vaccination.txt","r") as fh:
461          for row in fh:
462              # Split the string into a list
463              patientInformation = listCheck(str(row))
464              # Putting the list into the master list
465              allRecords.append(patientInformation)
466      return allRecords
```

- To read all the patient's records in “vaccination.txt” file.
- CF: Split the strings in “vaccination.txt” file into a list.
- Appends the information into a master list.

---

printAllPatientRecords(records)

```
468  # Printing the patients' record
469  def printAllPatientRecords(records):
470      # Print header
471      print("\n" + "=" * 126)
472      print("|" + "Vaccination ID".center(25) + "|" + "Name".center(25) + "|" + "Vaccination Center".center(25) + "|" + "Vaccine".center(20) + "|" + "Status".center(25) + "|")
473      print("=" * 126)
474      # Printing the list
475      for counter in range(len(records)):
476          items = records[counter]
477          print("|" + items[0].center(25) + "| " + items[1].ljust(24) + "|" + items[2].center(25) + "|" + items[3].center(20) + "|" + (items[4].rstrip()).center(25) + "|")
478      print("=" * 126)
```

Vaccination ID	Name	Vaccination Center	Vaccine	Status
VC1-1	IAN WONG	VC1	AF	NEW
VC2-2	ZHONG LI	VC2	BV	COMPLETED
VC1-3	CAITLYN	VC1	EC	COMPLETED
VC1-4	YUEN LOONG	VC1	AF	COMPLETED
VC2-5	WILLIAM	VC2	CZ	COMPLETED-D1
VC2-6	WING SING	VC2	DM	COMPLETED-D1
VC2-7	TOMMY TEE	VC2	DM	COMPLETED-D1
VC2-8	AARON YAO	VC2	CZ	NEW
VC1-9	PEGGY CHANG	VC1	AF	COMPLETED-D1
VC2-10	DARREN CHAN	VC2	DM	NEW
VC1-11	CHAI WAN TOO	VC1	EC	COMPLETED
VC1-12	MORELLIKUN	VC1	BV	COMPLETED-D1
VC2-13	MEGAN CHOO	VC2	AF	NEW
VC1-14	BENETT YANG	VC1	CZ	COMPLETED-D1
VC2-15	JOHN PHILIPPE	VC2	EC	NEW
VC1-16	HU TAO	VC1	BV	COMPLETED
VC1-17	NING GUANG	VC1	DM	COMPLETED-D1
VC2-18	VENTI	VC2	DM	COMPLETED
VC1-19	CHANG SHIAU HUEI	VC1	DM	COMPLETED-D1

Sample Output

- Print results for Section 5 in a table.
- Displays content in “vaccination.txt” file.

---

printAllPatientRecordsAndVaccineStatus()

```
480   # Section 5 - Print All Patient Records & Vaccination Status
481   def printAllPatientRecordsAndVaccineStatus():
482       allRecords = readAllPatientRecords()
483       printAllPatientRecords(allRecords)
```

The main function to display all patient’s records and vaccination status.

```
---------- Welcome to the Vaccination Record Management System ----------

        1. New Patient Registration
        2. Vaccine Administration
        3. Search Patient Record & Vaccination Status
        4. Statistical Information on Patients Vaccinated
        5. Print All Patient Record & Vaccination Status
        6. Exit

Please choose any operation from the given options: 5
```

Vaccination ID	Name	Vaccination Center	Vaccine	Status
VC1-1	IAN WONG	VC1	AF	NEW
VC2-2	ZHONG LI	VC2	BV	COMPLETED
VC1-3	CAITLYN	VC1	EC	COMPLETED
VC1-4	YUEN LOONG	VC1	AF	COMPLETED
VC2-5	WILLIAM	VC2	CZ	COMPLETED-D1
VC2-6	WING SING	VC2	DM	COMPLETED-D1
VC2-7	TOMMY TEE	VC2	DM	COMPLETED-D1
VC2-8	AARON YAO	VC2	CZ	NEW
VC1-9	PEGGY CHANG	VC1	AF	COMPLETED-D1
VC2-10	DARREN CHAN	VC2	DM	NEW
VC1-11	CHAI WAN TOO	VC1	EC	COMPLETED
VC1-12	MORELLIKUN	VC1	BV	COMPLETED-D1
VC2-13	MEGAN CHOO	VC2	AF	NEW
VC1-14	BENETT YANG	VC1	CZ	COMPLETED-D1
VC2-15	JOHN PHILIPPE	VC2	EC	NEW
VC1-16	HU TAO	VC1	BV	COMPLETED
VC1-17	NING GUANG	VC1	DM	COMPLETED-D1
VC2-18	VENTI	VC2	DM	COMPLETED
VC1-19	CHANG SHIAU HUEI	VC1	DM	COMPLETED-D1

Sample Output - Process

1. CF: Read all the patient's records in “vaccination.txt” file.
2. CF- Print results for Section 5.

==========

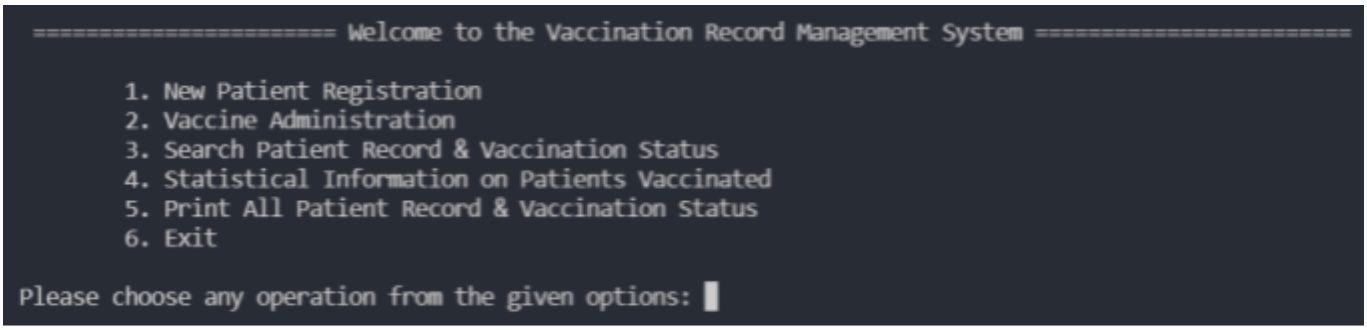
MAIN FUNCTION

==========

```
485  def menu():
486      while True:
487          # Check if the file needed exists, if not then create new ones
488          fileCheck()
489
490          # Printing menu
491          print("\n", " Welcome to the Vaccination Record Management System ".center(100, "="))
492          print("\n\t1. New Patient Registration\n\t2. Vaccine Administration\n\t3. Search Patient Record & Vaccination Status")
493          print("\t4. Statistical Information on Patients Vaccinated\n\t5. Print All Patient Record & Vaccination Status\n\t6. Exit")
494
495          # Input validation
496          try:
497              choice = int(input("\nPlease choose any operation from the given options: "))
498          except:
499              print("Your input is invalid. Please try again.")
500              continue
501          if choice <= 0 or choice > 6:
502              print("Please choose a number from 1 to 4 only.")
503              continue
504
505          # New Patient Registration
506          elif choice == 1:
507              newPatientRegistration()
508              continue
509
510          # Vaccine Administration
511          elif choice == 2:
512              if lineCount("patients.txt") > 0:
513                  vaccineAdministration()
514              else:
515                  print("No record exists. Please register in 'Section 1 - New Patient Registration' before proceeding to this section.")
516              continue
517
518          # Search Patient Record & Vaccination Status
519          elif choice == 3:
520              if lineCount("patients.txt") > 0:
521                  patientRecordAndVaccineStatus()
522              else:
523                  print("No record exists. Please register in 'Section 1 - New Patient Registration' before proceeding to this section.")
524              continue
525
526          # Statistical Information on Patients Vaccinated
527          elif choice == 4:
528              if lineCount("patients.txt") > 0:
529                  statisticalInfoOnPatientsVaccinated()
530              else:
531                  print("No record exists. Please register in 'Section 1 - New Patient Registration' before proceeding to this section.")
532              continue
533
534          # Print Patient Record & Vaccination Status
535          elif choice == 5:
536              if lineCount("patients.txt") > 0:
537                  printAllPatientRecordsAndVaccineStatus()
538              else:
539                  print("No record exists. Please register in 'Section 1 - New Patient Registration' before proceeding to this section.")
540              continue
541
542          # End the program
543          else:
544              break
```

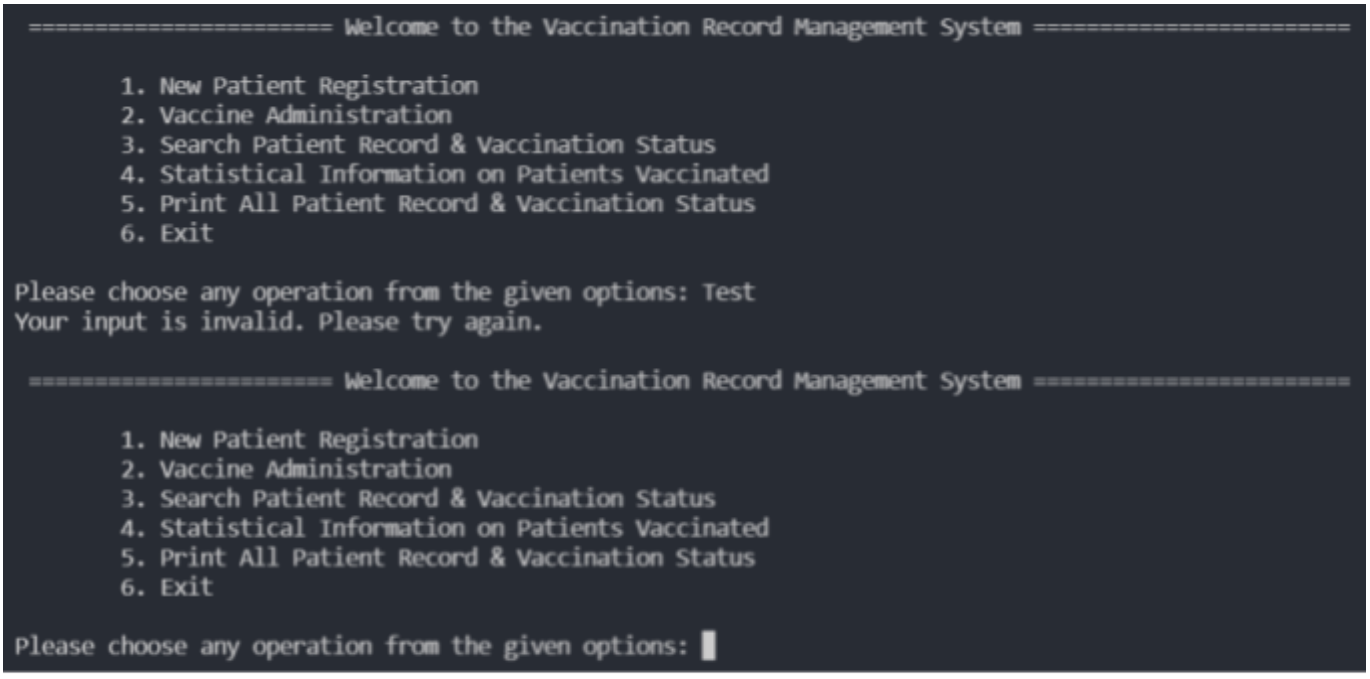
The main menu for the entire program.

1. CF: Generate “patients.txt” and “vaccination.txt” if they don’t exist when the code runs.



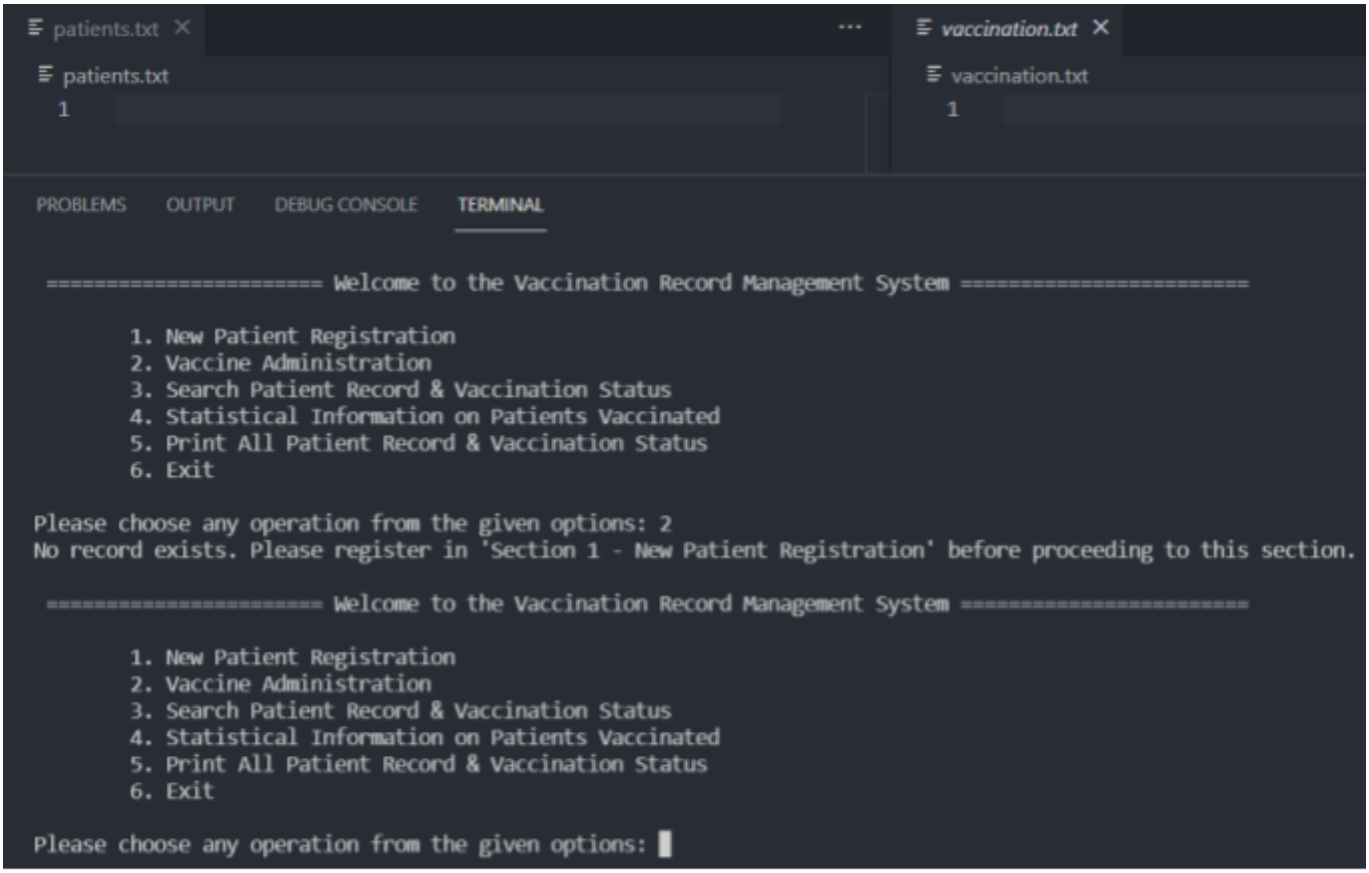
Output

2. Displays the main menu.



Output - Input validation

3. Lets the user reenter his input if he inputs anything other than 0 to 6.



Sample Output - Input validation

4. If the user tries to access Section 2 to 5 when there are no records in the "patients.txt" file, the program will inform them to register in Section 1 beforehand and exit back to the main menu.
5. END: 6

==========

MAIN CODE

==========

```
4    import os.path, sys
545  # Start the program
546  menu()
```

- Imports path from os module and sys module.
- CF: Display main menu.

CONTENT IN TEXT FILES

---

patient.txt

patients.txt

```
VC1-1|IAN WONG|VC1|19|AF|0149772234|ian.wyl@gmail.com|Height:170.0cm|Weight:55.0kg
VC2-2|ZHONG LI|VC2|80|BV|0167877666|zhongligeo@gmail.com|Height:180.0cm|Weight:67.0kg
VC1-3|CAITLYN|VC1|38|EC|0134765893|caitlyn4765@ymail.com|Height:168.0cm|Weight:43.0kg
VC1-4|YUEN LOONG|VC1|17|AF|014972234|idapowwwuwu@hotmail.com|Height:177.0cm|Weight:55.0kg
VC2-5|WILLIAM|VC2|22|CZ|016969782|goldsanitaryiwi@gmail.com|Height:166.0cm|Weight:65.0kg
VC2-6|WING SING|VC2|30|DM|0141783174|hohowingding@hotmail.com|Height:170.0cm|Weight:57.0kg
VC2-7|TOMMY TEE|VC2|13|DM|0196788546|123tommythetrain@gmail.com|Height:169.0cm|Weight:69.0kg
VC2-8|AARON YAO|VC2|17|CZ|0113564734|aaron_bw@gmail.com|Height:179.0cm|Weight:55.0kg
VC1-9|PEGGY CHANG|VC1|19|AF|01135778802|plzentertext@gmail.com|Height:149.0cm|Weight:38.0kg
VC2-10|DARREN CHAN|VC2|69|DM|013456669|darrenbanana@hotmail.com|Height:185.0cm|Weight:69.0kg
VC1-11|CHAI WAN TOO|VC1|55|EC|019789444|qingchaiwant2@gmail.com|Height:142.0cm|Weight:42.0kg
VC1-12|MORELLIKUN|VC1|32|BV|0187865324|morellimorello@gmail.com|Height:180.0cm|Weight:75.0kg
VC2-13|MEGAN CHOO|VC2|15|AF|019876333|meganchookul4u@ymail.com|Height:150.0cm|Weight:35.0kg
VC1-14|BENETT YANG|VC1|42|CZ|016442442|ytho@rocketmail.com|Height:169.0cm|Weight:40.0kg
VC2-15|JOHN PHILIPPE|VC2|40|EC|0183392573|felipeeeeeee3@hotmail.com|Height:190.0cm|Weight:57.0kg
VC1-16|HU TAO|VC1|25|BV|0197863548|whotao?@liyue.com|Height:151.0cm|Weight:40.0kg
VC1-17|NING GUANG|VC1|37|DM|017777277|morameat@jadepalace.com|Height:169.0cm|Weight:42.0kg
VC2-18|VENTI|VC2|65|DM|0118743078|anemoarchon@gmail.com|Height:177.8cm|Weight:54.7kg
VC1-19|CHANG SHIAU HUEI|VC1|19|DM|01135770088|plzentertext@gmail.com|Height:153.2cm|Weight:39.8kg
```

vaccination .txt

vaccination.txt

```
VC1-1|IAN WONG|VC1|AF|NEW
VC2-2|ZHONG LI|VC2|BV|COMPLETED
VC1-3|CAITLYN|VC1|EC|COMPLETED
VC1-4|YUEN LOONG|VC1|AF|COMPLETED
VC2-5|WILLIAM|VC2|CZ|COMPLETED-D1
VC2-6|WING SING|VC2|DM|COMPLETED-D1
VC2-7|TOMMY TEE|VC2|DM|COMPLETED-D1
VC2-8|AARON YAO|VC2|CZ|NEW
VC1-9|PEGGY CHANG|VC1|AF|COMPLETED-D1
VC2-10|DARREN CHAN|VC2|DM|NEW
VC1-11|CHAI WAN TOO|VC1|EC|COMPLETED
VC1-12|MORELLIKUN|VC1|BV|COMPLETED-D1
VC2-13|MEGAN CHOO|VC2|AF|NEW
VC1-14|BENETT YANG|VC1|CZ|COMPLETED-D1
VC2-15|JOHN PHILIPPE|VC2|EC|NEW
VC1-16|HU TAO|VC1|BV|COMPLETED
VC1-17|NING GUANG|VC1|DM|COMPLETED-D1
VC2-18|VENTI|VC2|DM|COMPLETED
VC1-19|CHANG SHIAU HUEI|VC1|DM|COMPLETED-D1
```

CONCLUSION

This program applied modular programming techniques so it consists of 5 main sections that each contains a different amount of functions, 7 general functions, 1 main function and the main code. This is because codes can be reused which increases readability. Moreover, the program is also menu-driven which provides a better user experience for the patients, especially the digital immigrants. Good programming practices such as variable naming conventions in camel casing, comments and indentation are also implemented for better manageability.

I have personally learned a lot throughout the assignment in terms of programming practices and techniques, and I am glad that I had the chance to solve problems like these as it puts my practical skills to the test.

REFERENCES

GeeksforGeeks. (2019). *Python Exit Commands: quit(), exit(), sys.exit() and os._exit()*. GeeksforGeeks. https://www.geeksforgeeks.org/python-exit-commands-quit-exit-sys-exit-and-os-_exit/

Kanyakumari, D. (2019). *Malaysian Health Sector in Desperate Need of Manpower Boost*. CNA. https://www.channelnewsasia.com/news/asia/malaysian-health-sector-doctors-overworked-12114348

Python. (n.d.). *os.path — Common Pathname Manipulations*. Python. https://docs.python.org/3/library/os.path.html