



INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT010-3-1-PYP

PYTHON PROGRAMMING

APD1F2106CE

Name: Su Xin Hong

TP No.: TP061159

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 and Assumptions	3
Design of the Program	5
Pseudocode	5
Welcome Page Function	6
Registration Functions	7
Vaccination Function	10
Search Function	13
Statistics Function	14
Run Application Function	16
Main Code	17
Flowchart	18
Welcome Page Function	19
Registration Functions	20
Vaccination Function	26
Search Function	37
Statistics Function	40
Run Application Function	49
Main Code	55
Program Source Code and Explanation	56
Welcome Page Function	56
Registration Function	57
Vaccination Function	61
Search Function	66
Statistics Function	67

Run Application Function.....	70
Main Code.....	72
Screenshots of Sample Input/Output and Explanation	73
Text Files	73
patients.txt.....	73
vaccination.txt.....	74
Welcome Page Function	75
Registration Function.....	78
Vaccination Function	81
Search Function	85
Statistics Function.....	86
Quit Program.....	88
Conclusion	89
References.....	91

Introduction and Assumptions

This documentation is about the Covid-19 vaccination record management program. According to the situations given on the question paper, Covid-19 virus is currently widely spread in the world and caused a global pandemic. Hence, the people are needed to be vaccinated so that they have a lesser chance to get infected by the virus and reduce the effect of the virus on them. In order to track and record all vaccination data a program is needed. This program is coded in Python language. There are 2 vaccination centres which are VC1 and VC2. There are also 5 types of vaccine available for the patients to choose according to their criteria and which age group that they are in. All of the requirements and the details all about the vaccines were presented as the table below which is provided in the question paper.

Vaccine Code	Dosage Required	Interval Between Doses	Age Group
AF	2	2 weeks (or 14 days)	12 and above
BV	2	3 weeks (or 21 days)	18 and above
CZ	2	3 weeks (or 21 days)	12 - 45
DM	2	4 weeks (or 28 days)	12 and above
EC	1		18 and above

The program that is created followed all the requirements and filter all of the patients and show them which are the vaccine that are suitable and let them choose by their own favor. Besides that, the patients also can choose which vaccination centres that they want to go to.

This program will be a menu driven program where the user will give input in the program to choose what they want to do in the program. There will be functions like registering, vaccination, search and statistics. The registering function is to register a new patient, the vaccination is to give vaccine to the patients and record the time and date given and update their status, search function is for searching a particular patient's information while the statistics function will be use to see an overall data of the vaccination centres.

In this program it is assumed that all of the patients will come back for their next dose after the first dose given following the interval themselves if needed. The program won't check if they follow the interval given and get back to get the other dose after the interval given for each of the vaccine.

Moreover, it is also assumed that the user of this program follows all the instruction and give the inputs following the instruction. Or else crashes might happen in the program.

Besides, it is also assumed that the user of this program won't regret what function had they choose while halfway running the function. Because there is not exit while the function is running halfway. As an example, when the user is registering a new patient and while giving input for the information, he wants to look at the statistics, it is not possible to do so in this program. He has to finish the program or kill it then go back to the menu page and choose the correct function.

Design of the Program

Pseudocode

The pseudocode is divided into several section according to the menu requirement in the question paper. They are divided into Welcome Page Function, Registration Function. Vaccination Function, Search Function and Statistics Function. Each of them contains 1 function while except Registration Function contains a combination of 2 functions to complete it. Then the flow to run each of the is put into another function called Run Application Function. The main code of this program is only running the Run Application Function. Hence, there is no global variable in this program.

Welcome Page Function

```
FUNCTION welcome_page()
    operation = ""
    PRINT "Welcome, please choose your operation."
    PRINT NEWLINE
    PRINT "Enter 'register' to register new patient."
    PRINT "Enter 'vaccine' for vaccine administration."
    PRINT "Enter 'search' to search for patient record."
    PRINT "Enter 'statistics' to see patients statistics."
    PRINT "Enter '0' to quit."
    PRINT NEWLINE
    DOWHILE TRUE
        PRINT "Please enter your operation here: "
        READ operation
        IF (operation != 'register') AND (operation != 'vaccine') AND (operation != 'search') AND (operation != 'statistics') AND (operation != '0') THEN
            PRINT 'Please input the right operation.'
            SKIP TO NEXT ITERATION
        ELSE
            EXITLOOP
        ENDIF
    ENDDO
    RETURN operation
ENDFUNCTION
```

Registration Functions

The registration function is separated into 2 functions to filter out patients with age lesser than 12.

```
FUNCTION registration_age_name_vc()
    DOWHILE TRUE
        PRINT "Please choose your desired vaccination centre ['VC1' or 'VC2']:"
        READ vaccinationCentre
        IF (vaccinationCentre != 'VC1') AND (vaccinationCentre != 'VC2') THEN
            PRINT 'Please input the right code for vaccination centre.'
            PRINT NEWLINE
        ELSE
            EXITLOOP
        ENDIF
    ENDDO

    PRINT 'Please input your name (without space, use "_" to indicate space): '
    READ name
    name = REPLACE ' ' IN name WITH '_'
    DOWHILE TRUE
        TRY
            PRINT 'Please input your age: '
            READ age
            INT age
            EXITLOOP
        EXCEPT
            PRINT 'Please input your age in numerical form.'
        ENDTRY
    ENDDO

    IF (age < 12) THEN
        PRINT 'Sorry you are too young for vaccine.'
        PRINT '=' * 50
    ENDIF

    RETURN (age, name, vaccinationCentre)
ENDFUNCTION
```



```
FUNCTION registration_operation (age, name, vaccinationCentre)
    PRINT "Please input your email: "
    READ email
    PRINT "Please input your contact number [without space]: "
    READ contactNumber

    IF (age >= 18) AND (age <= 45) THEN
        DOWHILE TRUE
            PRINT 'The vaccine suitable for you are "AF", "BV", "CZ", "DM" and "EC".'
            PRINT 'Please input the vaccine code that you want to have: '
            READ vaccine
            IF (vaccine != "AF") AND (vaccine != "BV") AND (vaccine != "CZ") AND (vaccine != "DM") AND (vaccine != "EC") THEN
                PRINT 'Please input the right code for the vaccine.'
                PRINT NEWLINE
            ELSE
                EXITLOOP
            ENDIF
        ENDDO
    ENDIF

    IF (age > 45) THEN
        DOWHILE TRUE
            PRINT 'The vaccine suitable for you are "AF", "BV", "DM" and "EC".'
            PRINT 'Please input the vaccine code that you want to have: '
            READ vaccine
            IF (vaccine != "AF") AND (vaccine != "BV") AND (vaccine != "DM") AND (vaccine != "EC") THEN
                PRINT 'Please input the right code for the vaccine.'
                PRINT NEWLINE
            ELSE
                EXITLOOP
            ENDIF
        ENDDO
    ENDIF
```

```
IF (age >= 12) AND (age < 18) THEN
    DOWHILE TRUE
        PRINT 'The vaccine suitable for you are "AF", "CZ" and "DM".'
        PRINT 'Please input the vaccine code that you want to have: '
        READ vaccine
        IF (vaccine != "AF") AND (vaccine != "CZ") AND (vaccine != "DM") THEN
            PRINT 'Please input the right code for the vaccine.'
            PRINT NEWLINE
        ELSE
            EXITLOOP
        ENDIF
    ENDDO
ENDIF

PRINT '=' * 50
IF (vaccine == 'EC') THEN
    dosageRequired = '1Dose'
ELSE
    dosageRequired = '2Dose'
ENDIF

OPENFILE "patients.txt" FOR READ
patient = []
READFILELINES "patients.txt", patient
patientAmount = SIZE(patient) - 1
CLOSEFILE "patients.txt"
patientCode = patientAmount + 1
FILL patientAmount TO 4 DIGITS
PRINT 'Your code is', patientCode, 'Please remember your code.'
```

```
OPENFILE "patients.txt" FOR APPEND
informationInput = NEWLINE, patientCode, ' ', name, ' ', age, ' ', vaccinationCentre, ' ', vaccine, ' ', dosageRequired, ' ', contactNumber, ' ', eMail, ' ', 'NEW'
APPENDFILE "patients.txt", informationInput
CLOSEFILE "patients.txt"

PRINT 'Thanks for registering', name, '.', 'Your registered vaccine is', vaccine
PRINT 'Your code is ', patientCode
PRINT 'Press Enter to continue.'
PRINT '=' * 50
ENDFUNCTION
```

Vaccination Function

```

FUNCTION vaccination_operation()
    DOWHILE TRUE
        PRINT 'Had you done your registration? ["yes" if yes, "no" if no]: '
        READ registration
        IF (registration != 'yes') AND (registration != 'no') THEN
            PRINT 'Please input the right answer.'
            SKIP TO NEXT ITERATION
        ELSE
            IF (registration == 'no') THEN
                ageNameVc = []
                ageNameVc = CALL registration_age_name_vc()
                name = ageNameVc[1]
                age = ageNameVc[0]
                vaccinationCentre = ageNameVc[2]
                IF (age < 12) THEN
                    EXITLOOP
                ENDIF
                CALL registration_operation (age, name, vaccinationCentre)
                SKIP TO NEXT ITERATION
            ELSE
                IF (registration == 'yes') THEN
                    TRY
                        DOWHILE TRUE
                            TRY
                                PRINT 'what is your ID code: '
                                READ patientCode
                                INT patientCode
                                EXITLOOP
                            EXCEPT
                                PRINT 'Please input the right ID code.'
                            ENDTRY
                        ENDDO
                    ENDTRY
                ENDIF
            ENDIF
        ENDIF
    ENDWHILE
ENDFUNCTION

```

```

OPENFILE "patients.txt" FOR READ
allPatientInformation = []
READFILELINES "patients.txt", allPatientInformation
CLOSEFILE "patients.txt"

lineInTextFile = patientCode
patientInformation = allPatientInformation[lineInTextFile]
patientNewInformation = patientInformation
patientInformation = SPLIT (patientInformation, ' ')
patientName = patientInformation [1]
PRINT 'Hello', patientName
registeredVaccine = patientInformation [4]
PRINT 'Your vaccine registered is', registeredVaccine
patientStatus = patientInformation [8]
PRINT 'Your vaccine status was', patientStatus

patientNewInformation = SPLIT (patientNewInformation, ' ')
IF (registeredVaccine == 'EC') THEN
    patientNewInformation [8] = 'COMPLETED'
ELSE
    IF (patientStatus == 'COMPLETED-D1') THEN
        patientNewInformation [8] = 'COMPLETED'
    ELSE
        IF (patientStatus == 'NEW') THEN
            patientNewInformation [8] = 'COMPLETED-D1'
        ELSE
            IF (patientStatus == 'COMPLETED') THEN
                patientNewInformation [8] = 'COMPLETED'
            ENDIF
        ENDIF
    ENDIF
ENDIF
ENDIF
ENDIF

```

```
OPENFILE "patients.txt" FOR READ
patient = []
READFILELINES "patients.txt", patient
patientAmount = SIZE(patient) - 1
CLOSEFILE "patients.txt"
PRINT NEWLINE
IF (patientAmount != patientNewInformation [0]) AND (patientNewInformation [8] == 'COMPLETED') THEN
    patientNewInformation [8] = 'COMPLETED', NEWLINE
ELSE
    IF (patientAmount != patientNewInformation [0]) AND (patientNewInformation [8] == 'COMPLETED-D1') THEN
        patientNewInformation [8] = 'COMPLETED-D1', NEWLINE
    ENDIF
ENDIF

allPatientInformation[lineInTextFile] = patientNewInformation
allPatientInformation[lineInTextFile] = allPatientInformation[lineInTextFile][0], ' ', allPatientInformation[lineInTextFile][1], ' ',
allPatientInformation[lineInTextFile][2], ' ', allPatientInformation[lineInTextFile][3], ' ',
allPatientInformation[lineInTextFile][4], ' ', allPatientInformation[lineInTextFile][5], ' ',
allPatientInformation[lineInTextFile][7], ' ', allPatientInformation[lineInTextFile][8]

OPENFILE "patients.txt" FOR WRITE
WRITEFILE "patients.txt", allPatientInformation
CLOSEFILE "patients.txt"
PRINT '=' * 50
```

```
IF (patientStatus == 'COMPLETED-D1') THEN
    PRINT "Congratulations, you had completed your vaccine, you don't have to come again.", NEWLINE
    PRINT 'Press Enter to continue.'
    PRINT '=' * 50
ELSE
    IF (patientStatus == 'NEW') AND (registeredVaccine == 'EC') THEN
        PRINT "Congratulations, you had completed your vaccine, you don't have to come again.", NEWLINE
        PRINT 'Press Enter to continue.'
        PRINT '=' * 50
    ELSE
        IF (patientStatus == 'NEW') AND ((registeredVaccine == 'BV') OR (registeredVaccine == 'CZ'))
            PRINT "You had completed your first vaccine, please come back in 3 weeks (21 days) to get your second vaccine.", NEWLINE
            PRINT 'Press Enter to continue.'
            PRINT '=' * 50
        ELSE
            IF (patientStatus == 'NEW') AND (registeredVaccine == 'AF') THEN
                PRINT "You had completed your first vaccine, please come back in 2 weeks (14 days) to get your second vaccine." NEWLINE
                PRINT 'Press Enter to continue.'
                PRINT '=' * 50
            ELSE
                IF (patientStatus == 'NEW') AND (registeredVaccine == 'DM') THEN
                    PRINT "You had completed your first vaccine, please come back in 4 weeks (28 days) to get your second vaccine."
                    ,NEWLINE
                    PRINT 'Press Enter to continue.'
                    PRINT '=' * 50
                ELSE
                    IF (patientStatus == 'COMPLETED') THEN
                        PRINT "You had already completed your vaccine, why you came?"
                        PRINT 'Press Enter to continue.'
                        PRINT '=' * 50
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF
ENDIF
PRINT '=' * 50
```

```
IF (patientStatus == 'NEW') THEN
    doseGiven = 'D1'
ELSE
    IF (patientStatus == 'COMPLETED-D1') THEN
        doseGiven = 'D2'
    ENDIF
ENDIF

IF (patientStatus == 'NEW') OR (patientStatus == 'COMPLETED-D1') THEN
    OPENFILE "vaccination.txt" FOR APPEND
    IMPORT datetime
    vaccinationTime = NOW(datetime)
    recordInput = [patientCode, patientName, registeredVaccine, vaccinationTime, doseGiven]
    recordInput = recordInput[0], recordInput[1], recordInput[2], recordInput[3], recordInput[4]
    APPENDFILE "vaccination.txt", recordInput
    CLOSEFILE "vaccination.txt"
ENDIF

EXCEPT
    PRINT 'Your input code is not registered.'
    PRINT '=' * 50
    EXITLOOP
ENDTRY

ENDIF

EXITLOOP

ENDDO

ENDFUNCTION
```

Search Function

```
FUNCTION search_operation()
    DOWHILE TRUE
        TRY
            PRINT 'What is your ID code: '
            READ patientCode
            INT patientCode
            EXITLOOP
        EXCEPT
            PRINT 'Please input the right ID code.'
        ENDTRY
    ENDDO

    TRY
        OPENFILE "patient.txt" FOR READ
        allPatientInformation = []
        READFILELINES "patients.txt", allPatientInformation
        CLOSEFILE "patients.txt"
        lineInTextFile = patientCode
        patientInformation = allPatientInformation[lineInTextFile]
        PRINT '=' * 50
        PRINT 'This is the information about the patient.'
        PRINT 'The informations are arranged as sequence below.', NEWLINE
        PRINT '|Code| |Name| |Age| |Vaccine_Centre| |Vaccine_Name| |Dosage_Required| |Contact_No| |E-mail| |Status|'
        PRINT patientInformation
        PRINT 'Press Enter to continue.'
        PRINT '=' * 50
    EXCEPT
        PRINT 'Your input code is not registered.'
        PRINT '=' * 50
    ENDTRY
ENDFUNCTION
```

Statistics Function

```
FUNCTION statistics_operation()
    OPENFILE "patients.txt" FOR READ
    allPatientInformation = []
    READFILELINES "patients.txt", allPatientInformation
    CLOSEFILE "patients.txt"

    VC1Total = 0
    FOR EACH line IN allPatientInformation
        line = SPLIT(line, ' ')
        TRY
            IF (line[3] == 'VC1') THEN
                VC1Total = VC1Total + 1
            ENDIF
        EXCEPT
            SKIP TO NEXT ITERATION
        ENDTRY
    ENDFOR

    VC1Waiting = 0
    FOR EACH line IN allPatientInformation
        line = SPLIT(line, ' ')
        TRY
            IF (line[3] == 'VC1') AND (line[8] == 'COMPLETED-D1') THEN
                VC1Waiting = VC1Waiting + 1
            ENDIF
        EXCEPT
            SKIP TO NEXT ITERATION
        ENDTRY
    ENDFOR
```

```
    VC1Completed = 0
    FOR EACH line IN allPatientInformation
        line = SPLIT(line, ' ')
        TRY
            IF (line[3] == 'VC1') AND (line[8] == 'COMPLETED') THEN
                VC1Completed = VC1Completed + 1
            ENDIF
        EXCEPT
            SKIP TO NEXT ITERATION
        ENDTRY
    ENDFOR

    VC2Total = 0
    FOR EACH line IN allPatientInformation
        line = SPLIT(line, ' ')
        TRY
            IF (line[3] == 'VC2') THEN
                VC2Total = VC2Total + 1
            ENDIF
        EXCEPT
            SKIP TO NEXT ITERATION
        ENDTRY
    ENDFOR

    VC2Waiting = 0
    FOR EACH line IN allPatientInformation
        line = SPLIT(line, ' ')
        TRY
            IF (line[3] == 'VC2') AND (line[8] == 'COMPLETED-D1') THEN
                VC2Waiting = VC2Waiting + 1
            ENDIF
        EXCEPT
            SKIP TO NEXT ITERATION
        ENDTRY
    ENDFOR
```

```
VC2Complete = 0
FOR EACH line IN allPatientInformation
    line = SPLIT(line, ' ')
    TRY
        IF (line[3] == 'VC2') AND (line[8] == 'COMPLETED') THEN
            VC2Complete = VC2Complete + 1
        ENDIF
    EXCEPT
        SKIP TO NEXT ITERATION
    ENDTRY
ENDFOR

PRINT 'These are the statistics of both vaccination centre.'
PRINT '-' * 50
PRINT 'vaccination Centre',NEWTAB,NEWTAB, 'VC1',NEWTAB, 'VC2'
PRINT '-' * 50
PRINT 'Total Patients',NEWTAB,NEWTAB,NEWTAB, VC1Total,NEWTAB, VC2Total,NEWTAB
PRINT 'Waiting 2nd Dose Patients',NEWTAB, VC1Waiting,NEWTAB, VC2Waiting,NEWTAB
PRINT 'Completed Patients',NEWTAB,NEWTAB, VC1Complete,NEWTAB, VC2Complete,NEWTAB
PRINT '-' * 50
PRINT 'Press Enter to continue.'
PRINT '=' * 50
ENDFUNCTION
```


Run Application Function

```
FUNCTION run_application()
    DOWHILE TRUE
        operation = CALL welcome_page()

        IF (operation == 'register') THEN
            PRINT NEWLINE, "You had chosed registration."
            ageNameVC = CALL registration_age_name_vc ()
            name = ageNameVc[1]
            age = ageNameVc[0]
            vaccinationCentre = ageNameVc[2]
            IF (age < 12) THEN
                SKIP TO NEXT ITERATION
            ENDIF
            CALL registration_operation(age, name, vaccinationCentre)
        ENDIF

        IF (operation == 'vaccine') THEN
            PRINT NEWLINE, "You had chosed vaccine administration."
            CALL vaccination_operation()
        ENDIF

        IF (operation == 'search') THEN
            PRINT NEWLINE, "You had chosed to search for patients records."
            CALL search_operation()
        ENDIF

        IF (operation == 'statistics') THEN
            PRINT NEWLINE, "You had chosed to see patient statistics."
            CALL statistics_operation()
        ENDIF

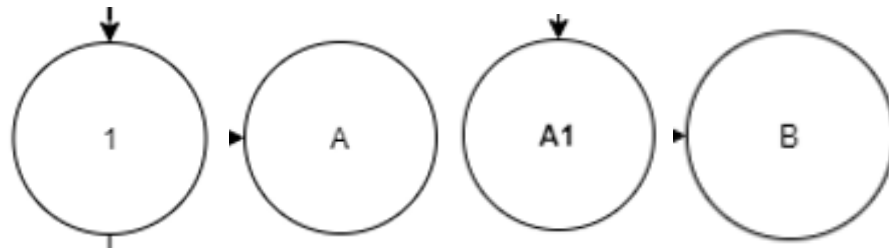
        IF (operation == 0) THEN
            PRINT NEWLINE, "Thank you !"
            EXITLOOP
        ENDIF
    ENDDO
ENDFUNCTION
```

Main Code

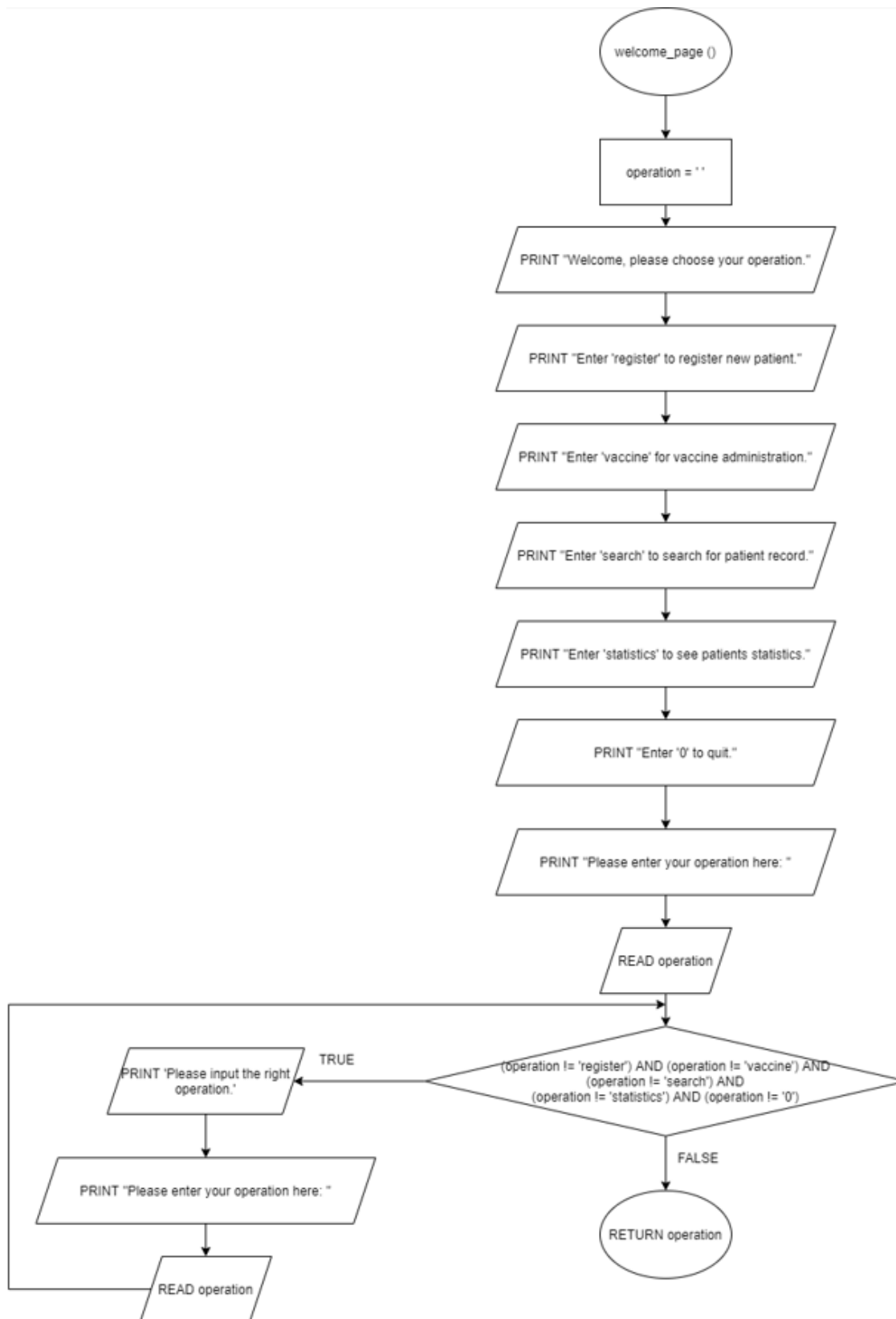
```
PROGRAM Vaccine Record Program
BEGIN
  CALL run_application()
END
```

Flowchart

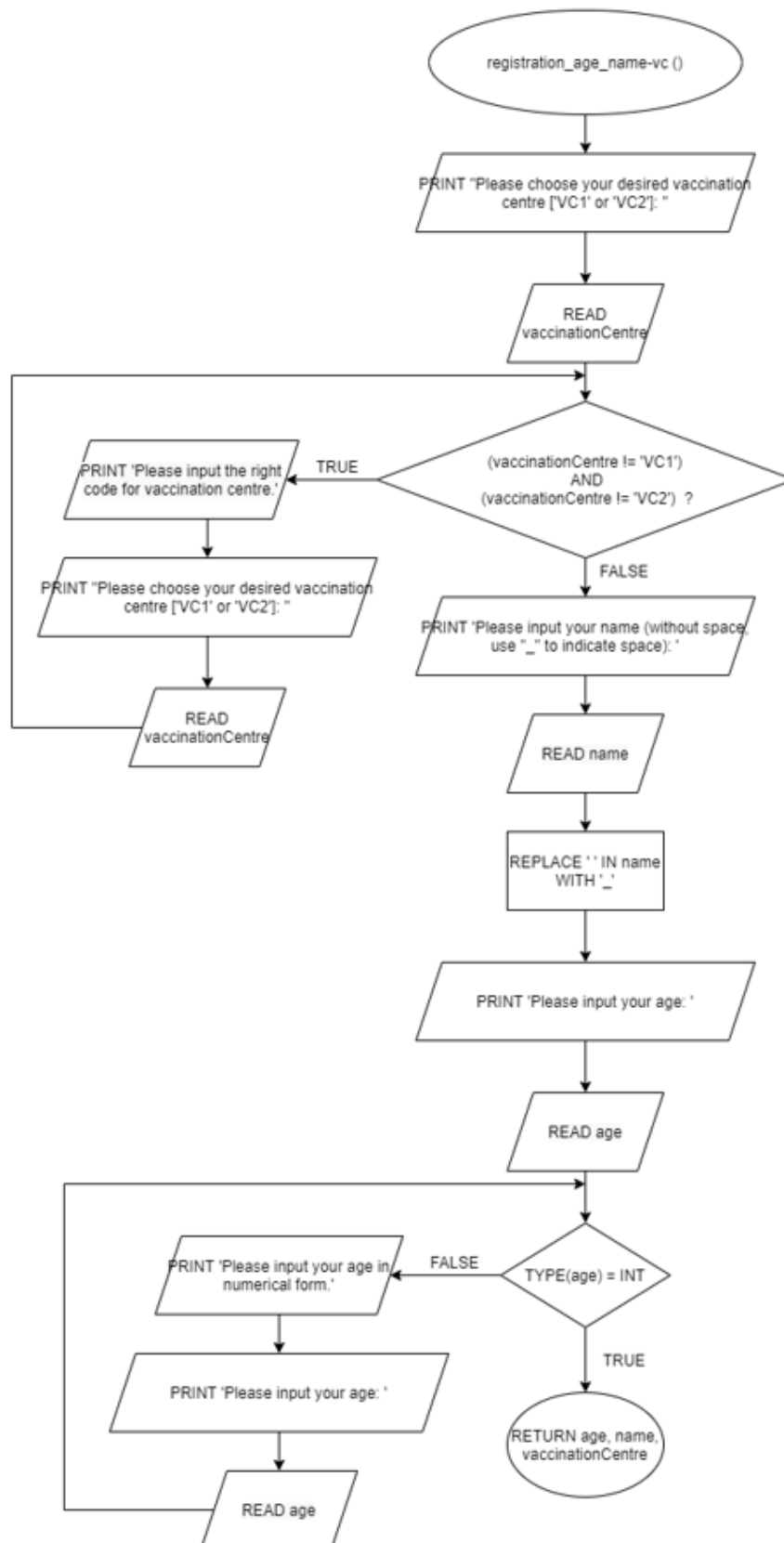
Before going into the flow chart, it is important to note that the number or alphabet bubble that cuts the flow chart is not related to other function. It resets in after every function. Below are some of the examples of the bubbles.

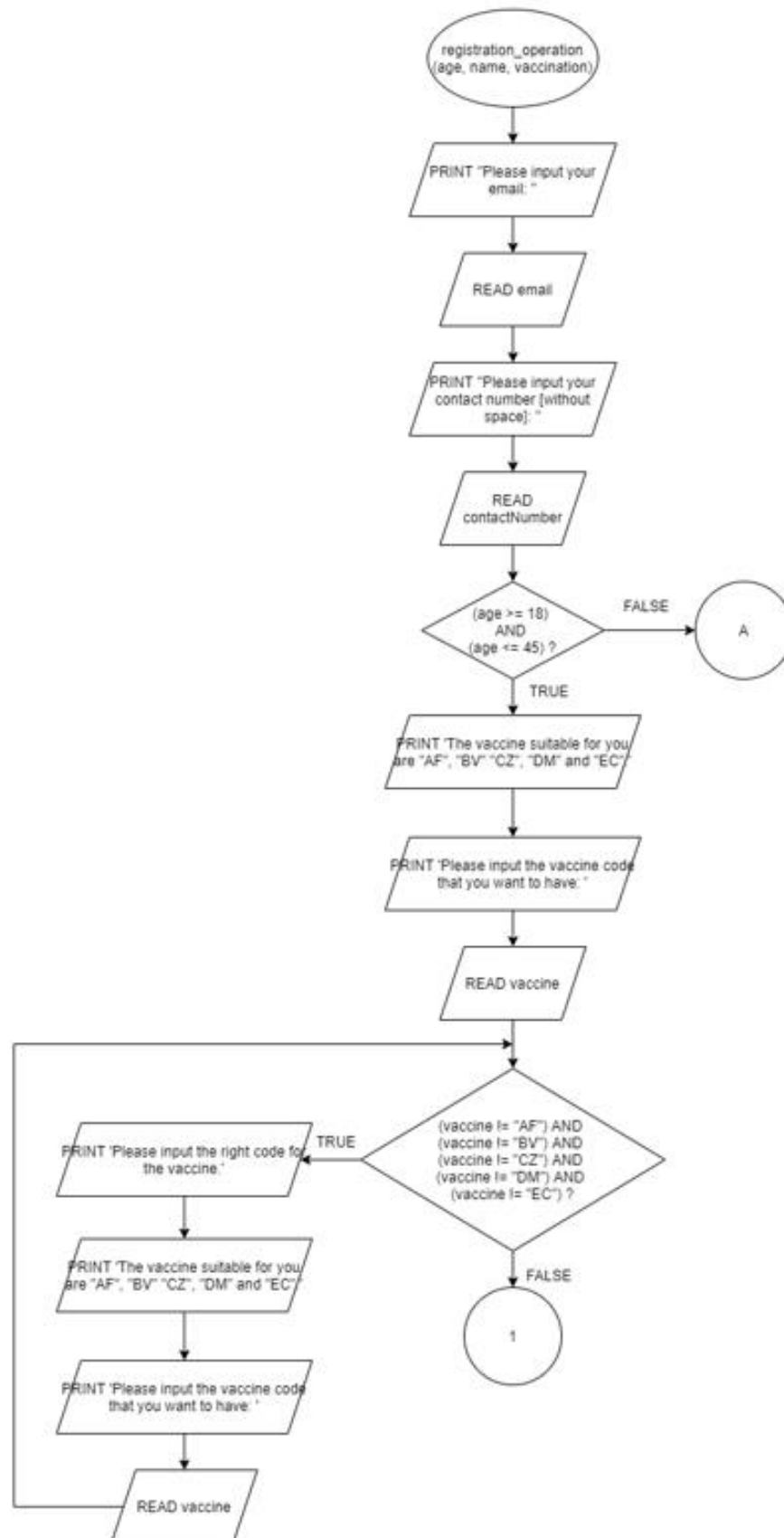


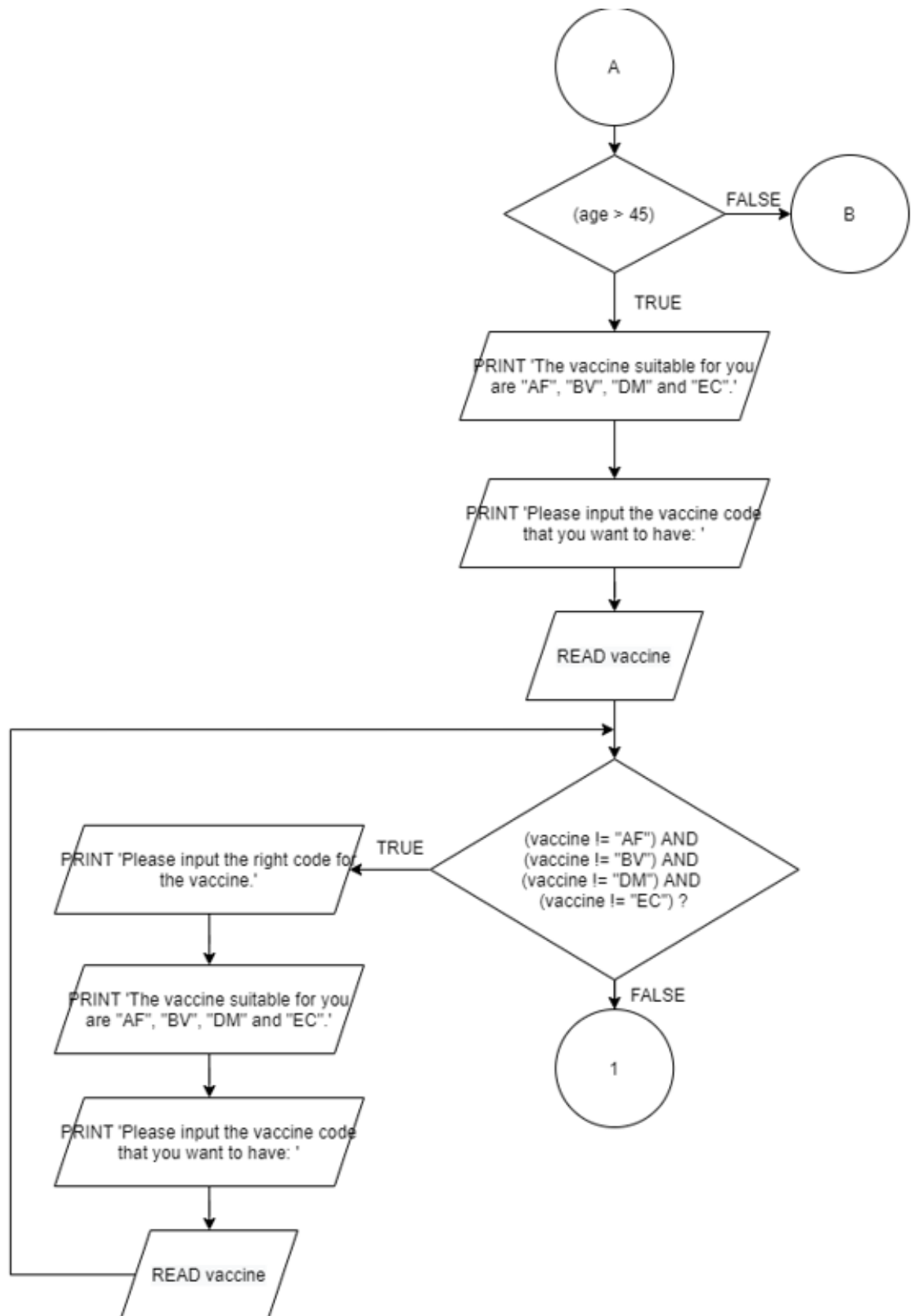
Welcome Page Function

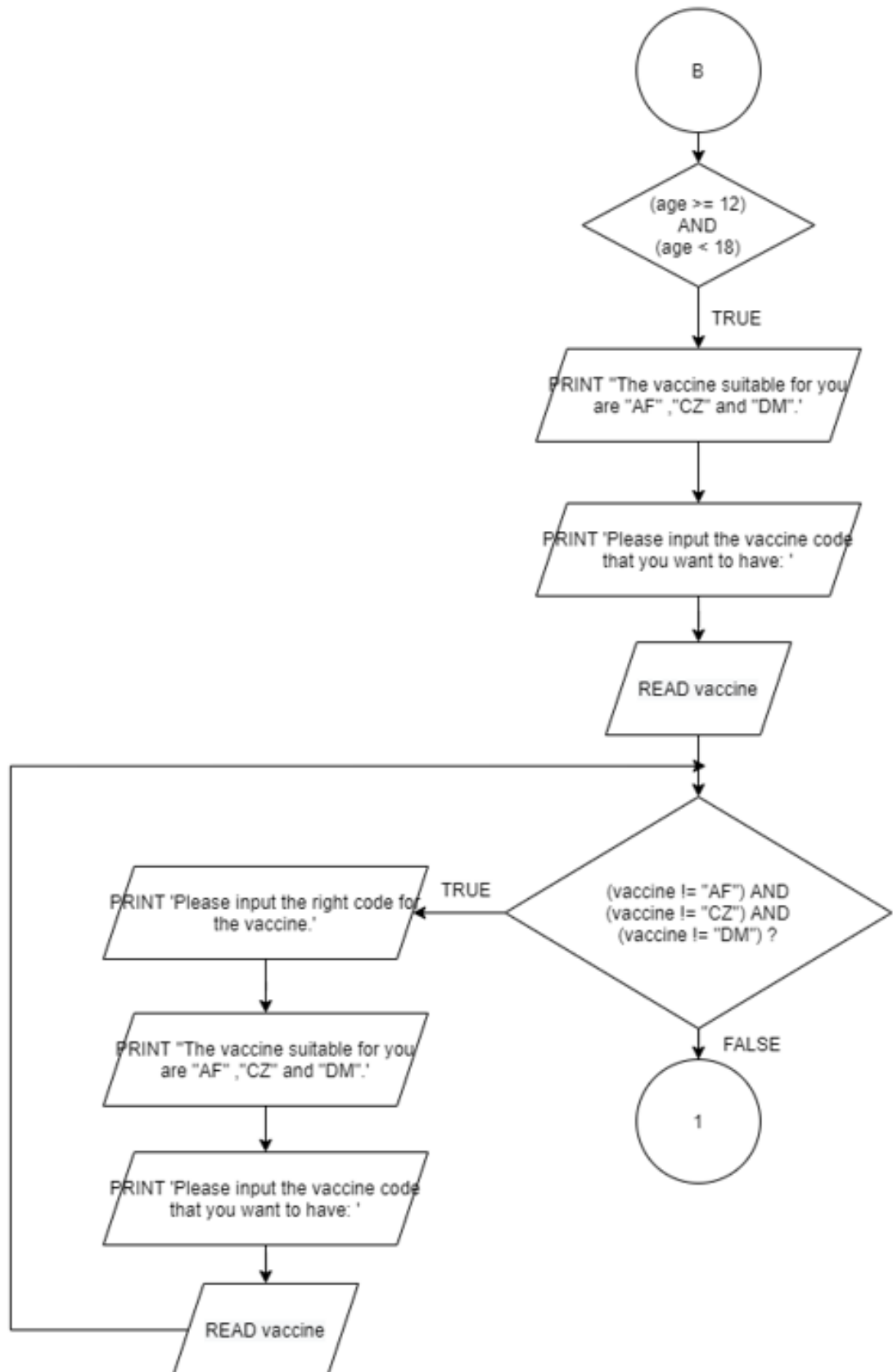


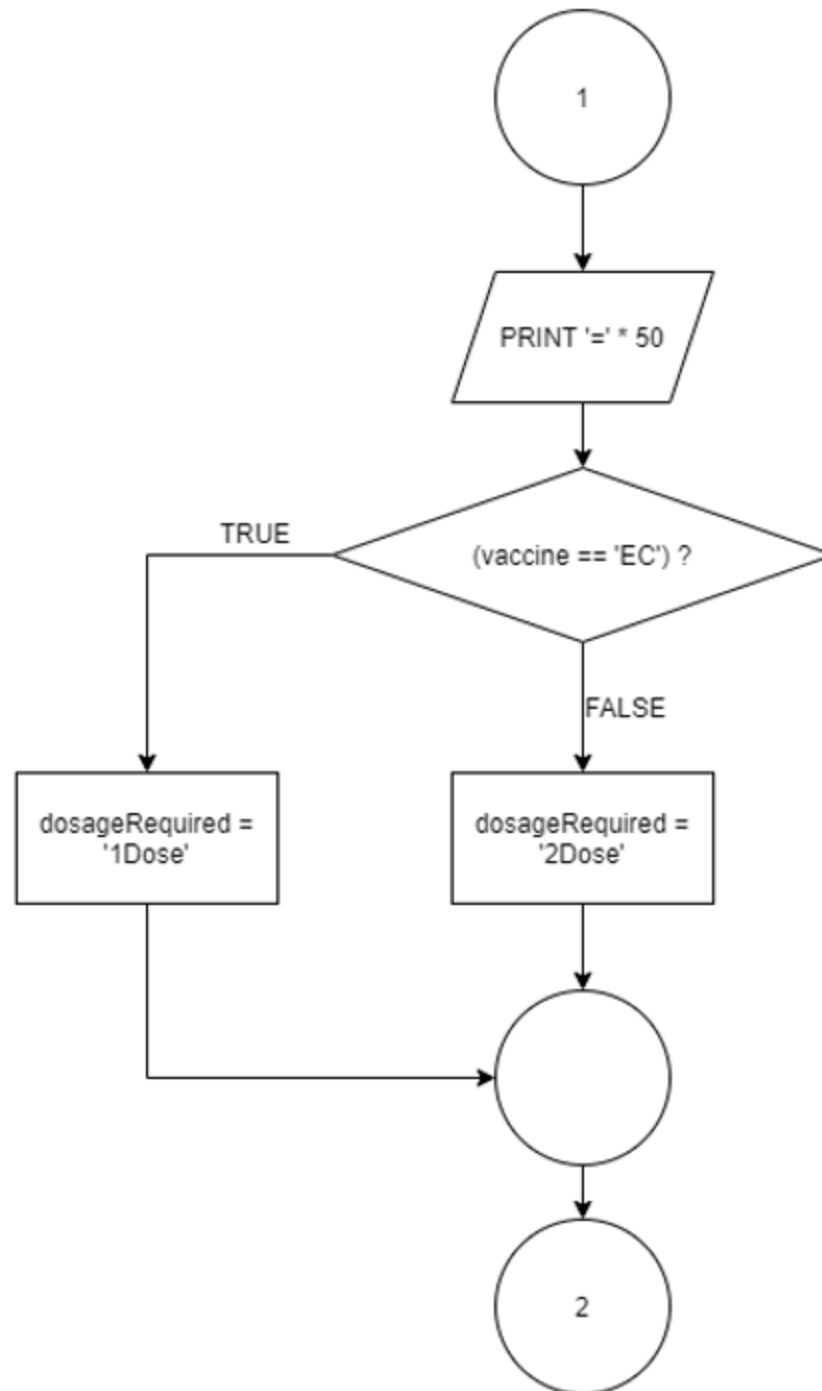
Registration Functions

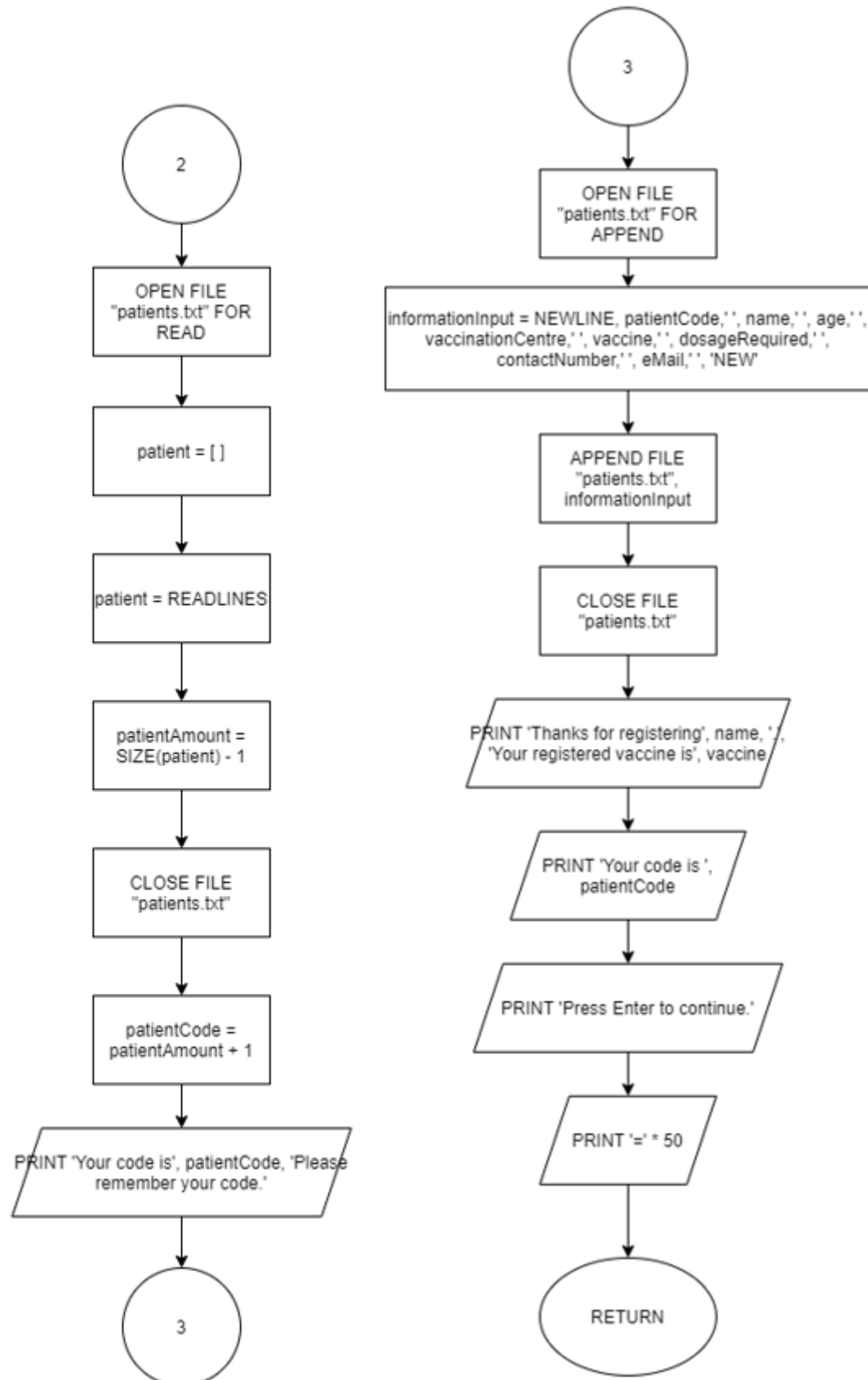




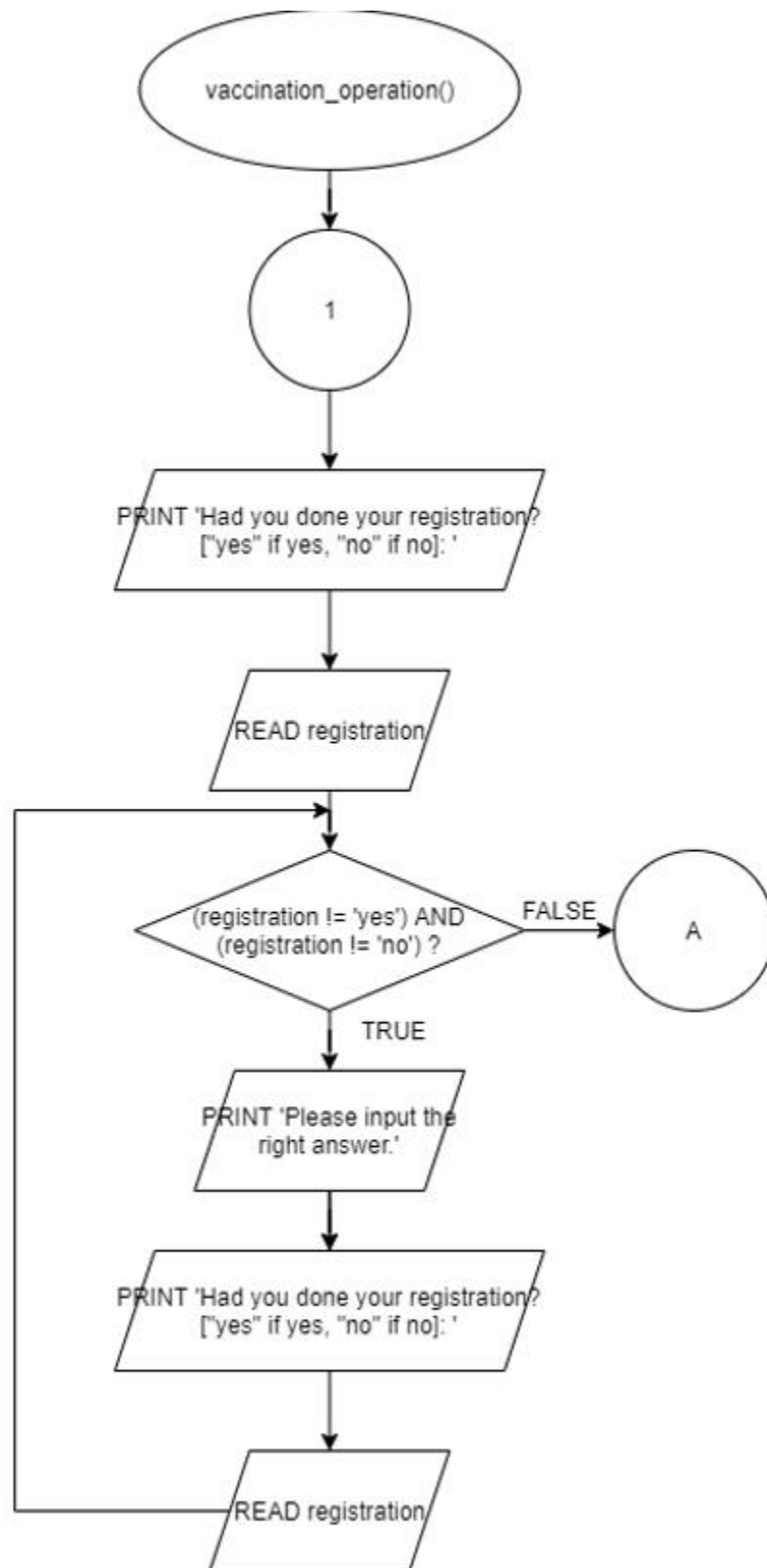


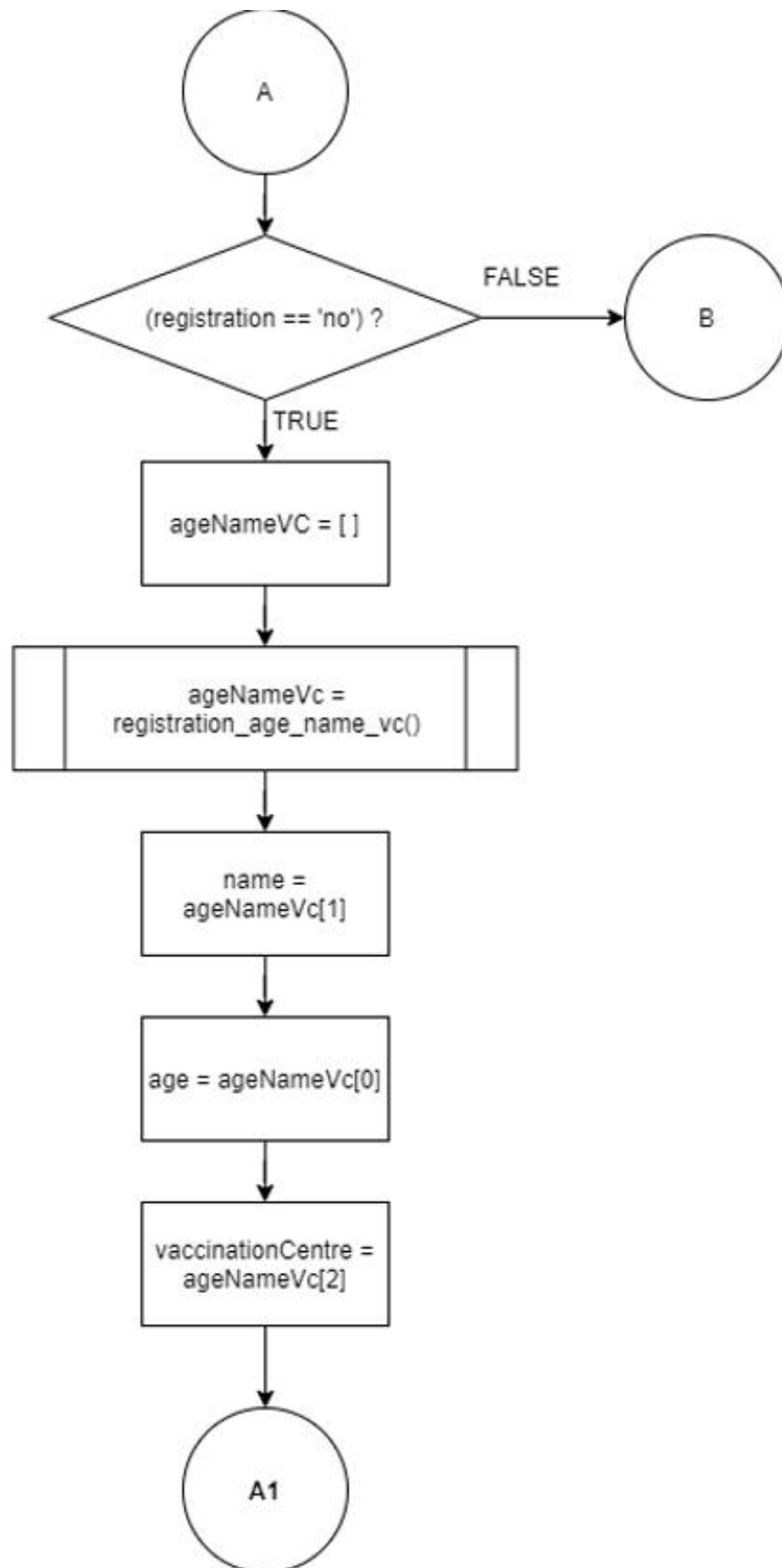


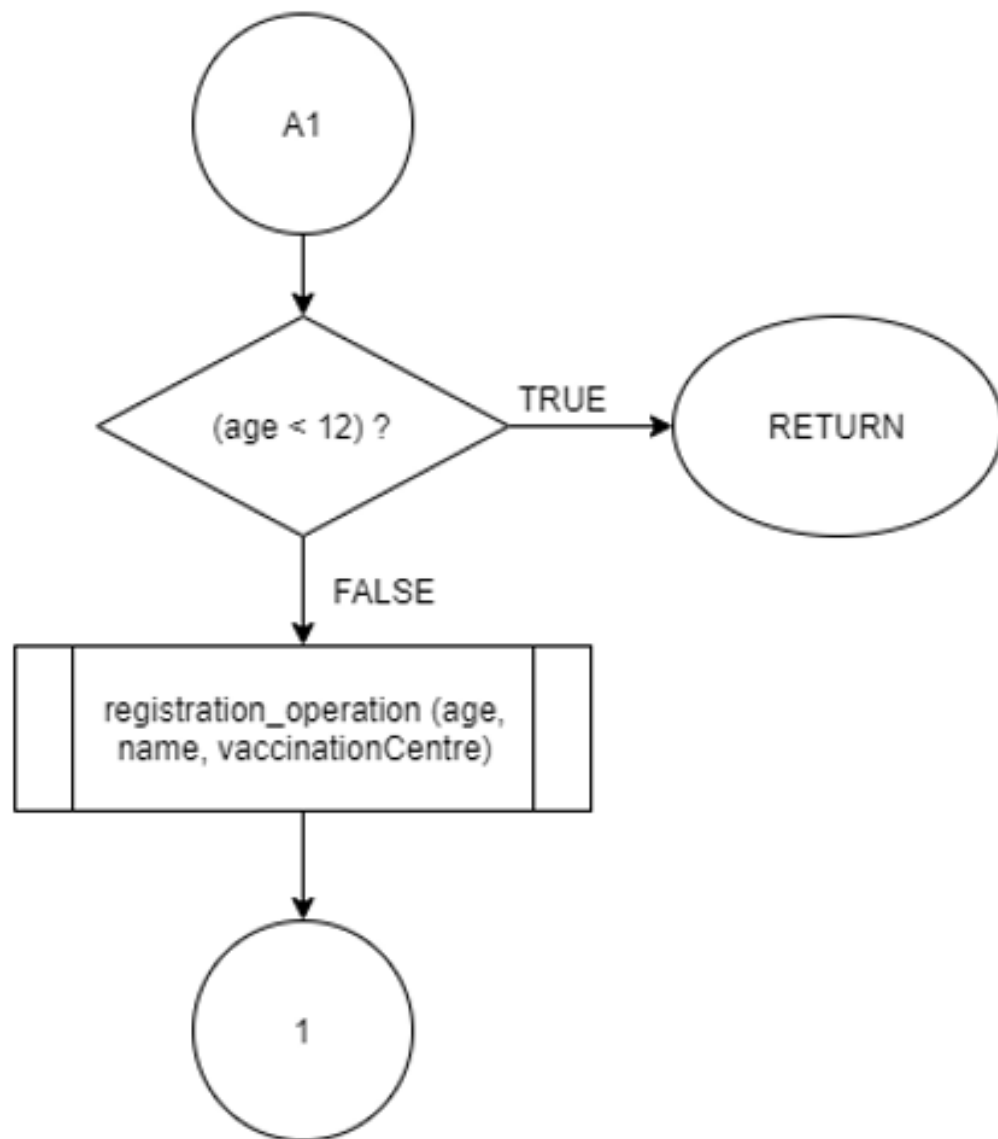


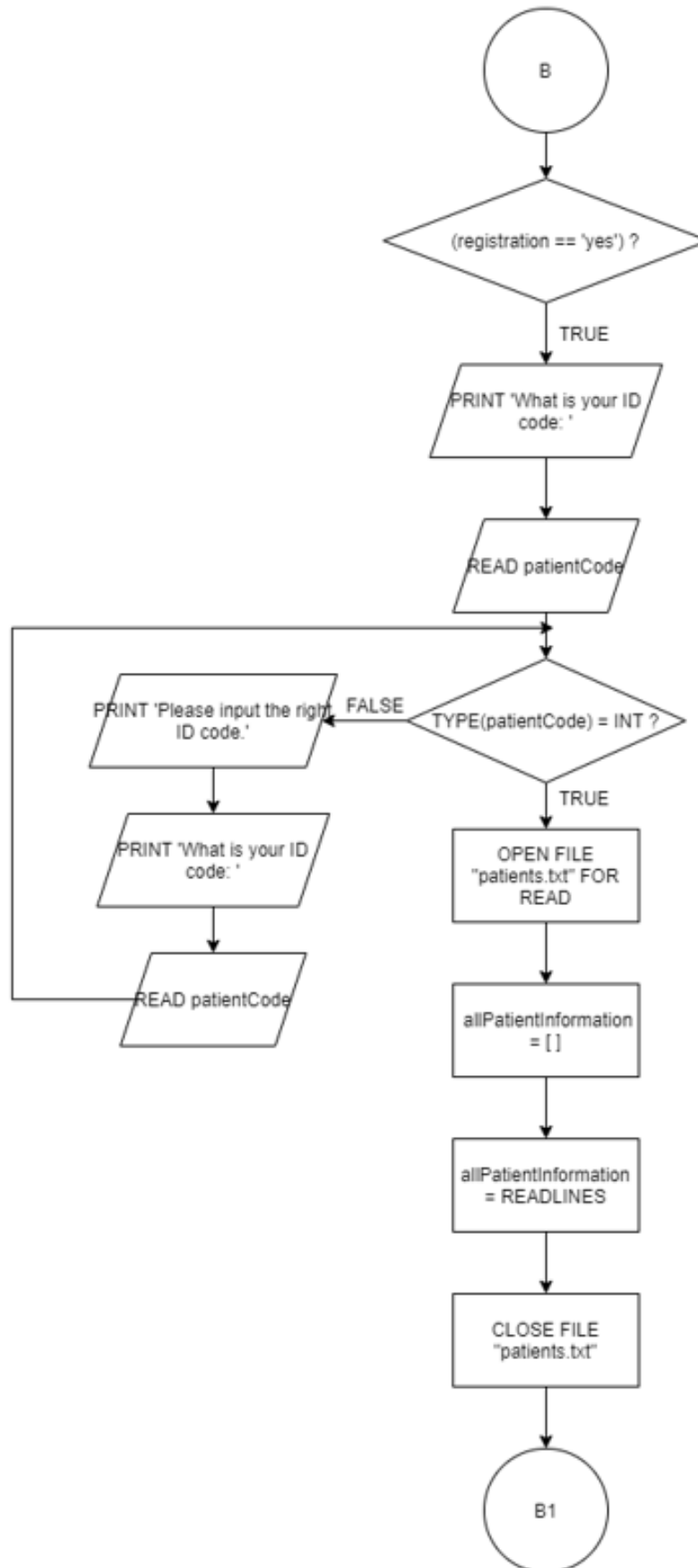


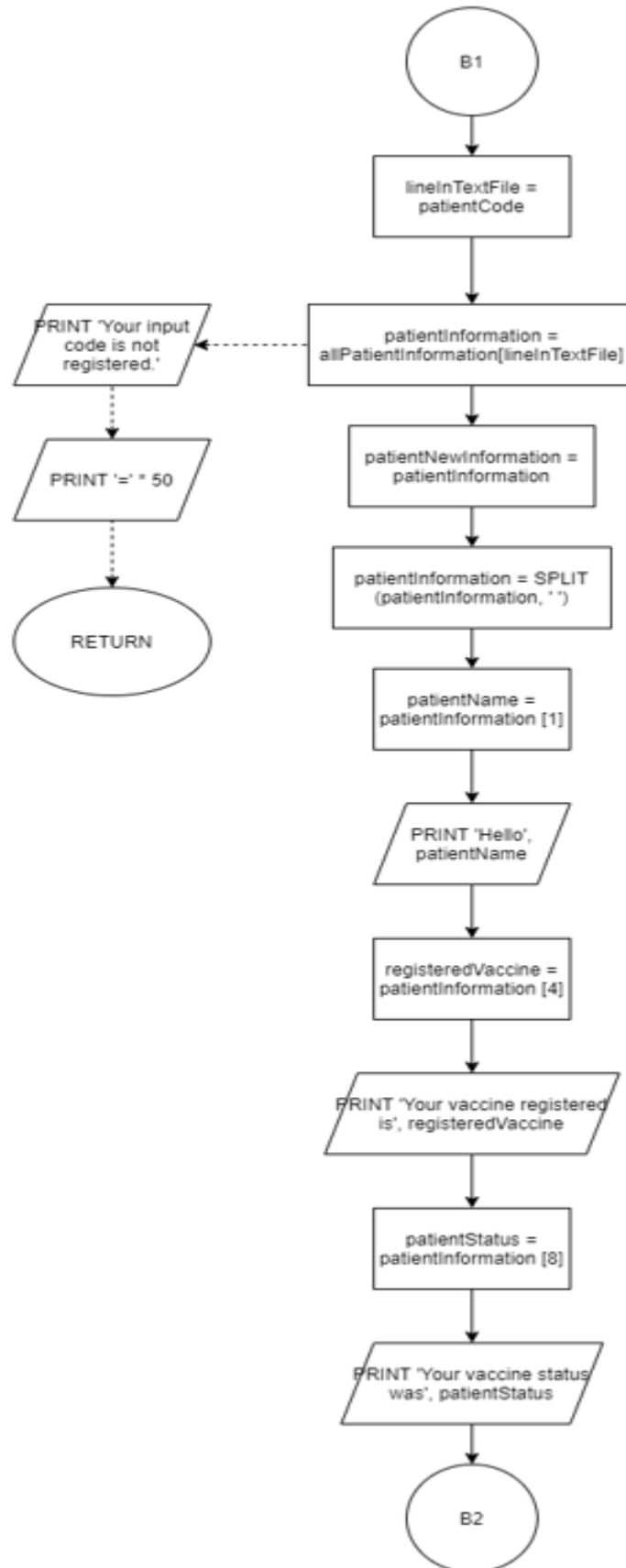
Vaccination Function

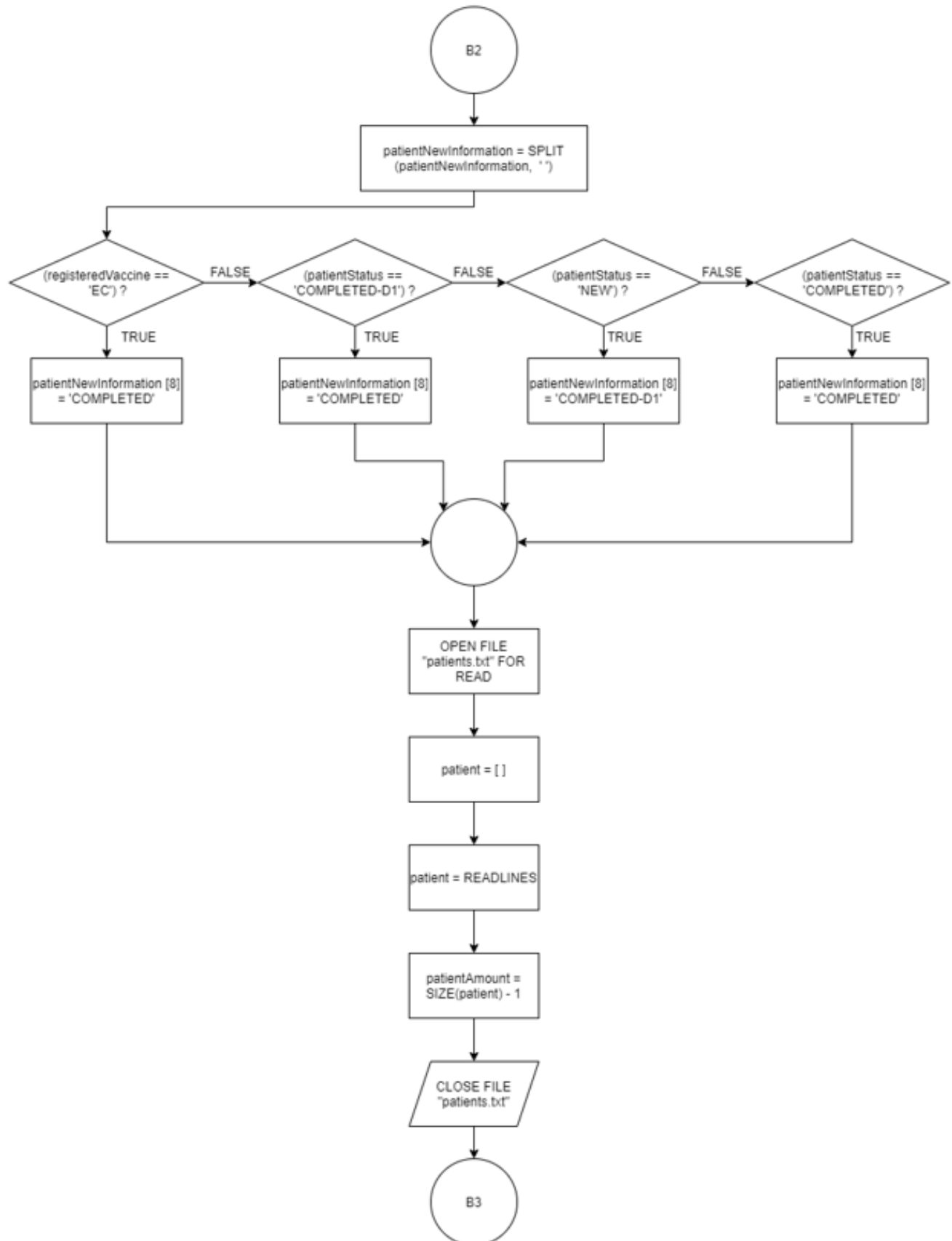


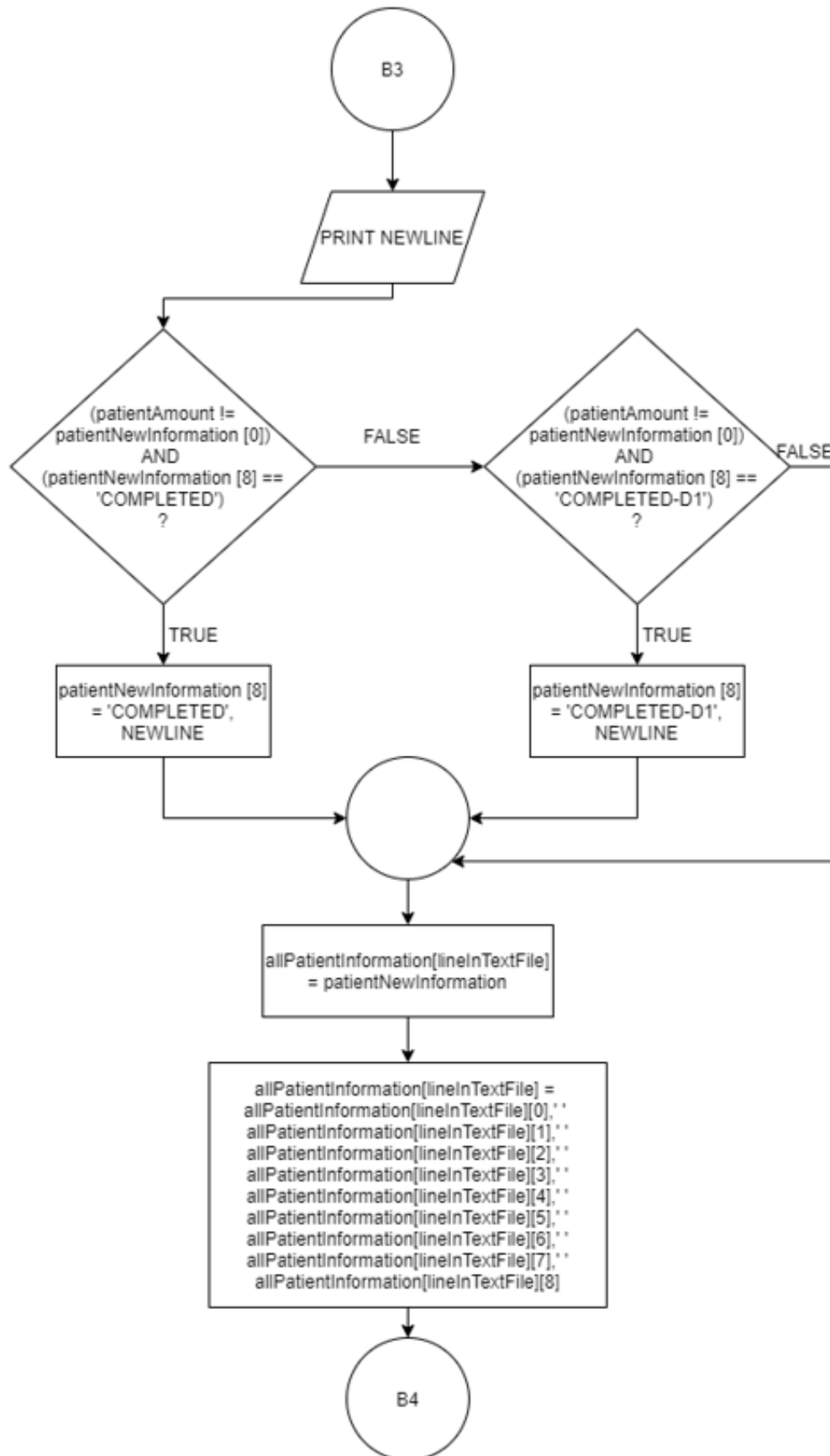


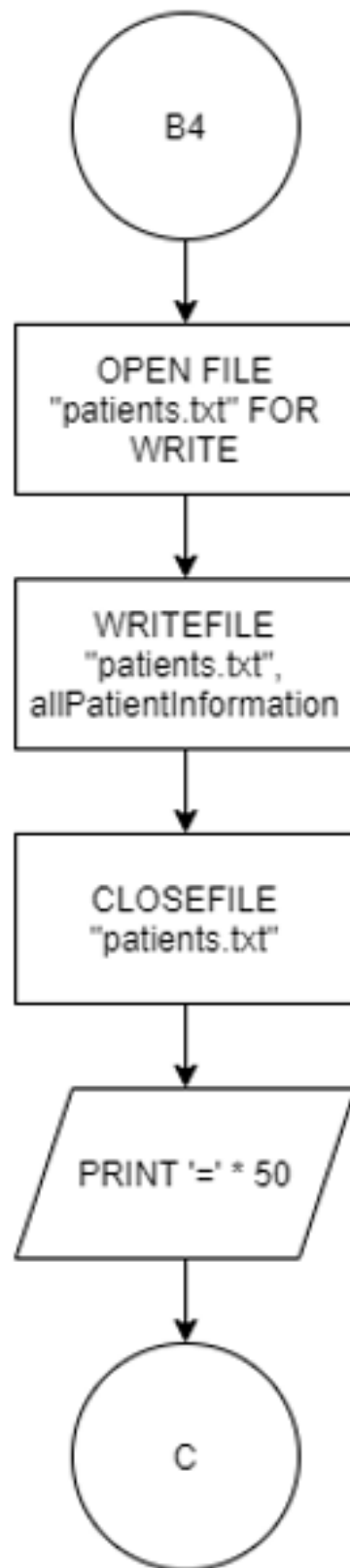


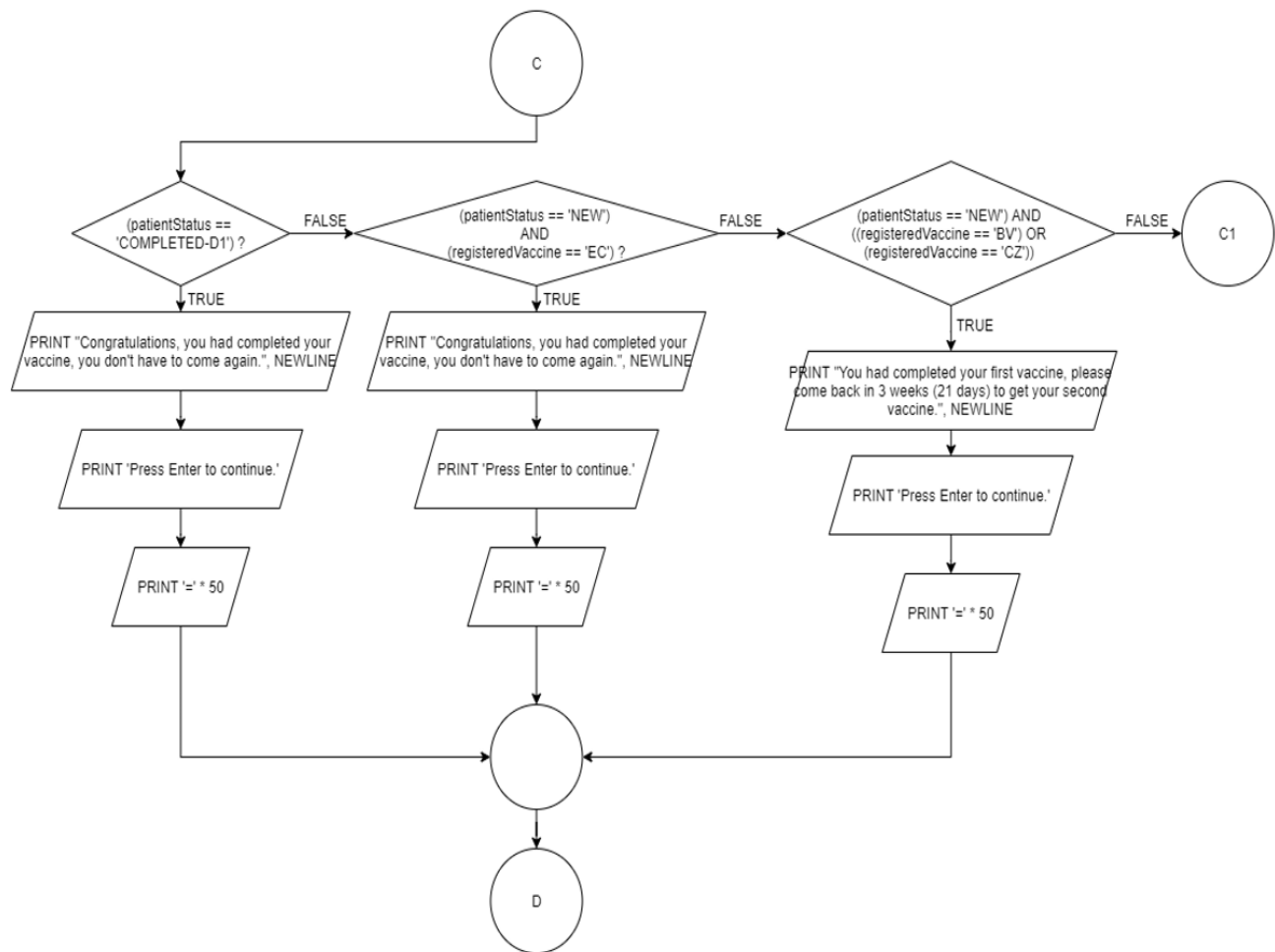


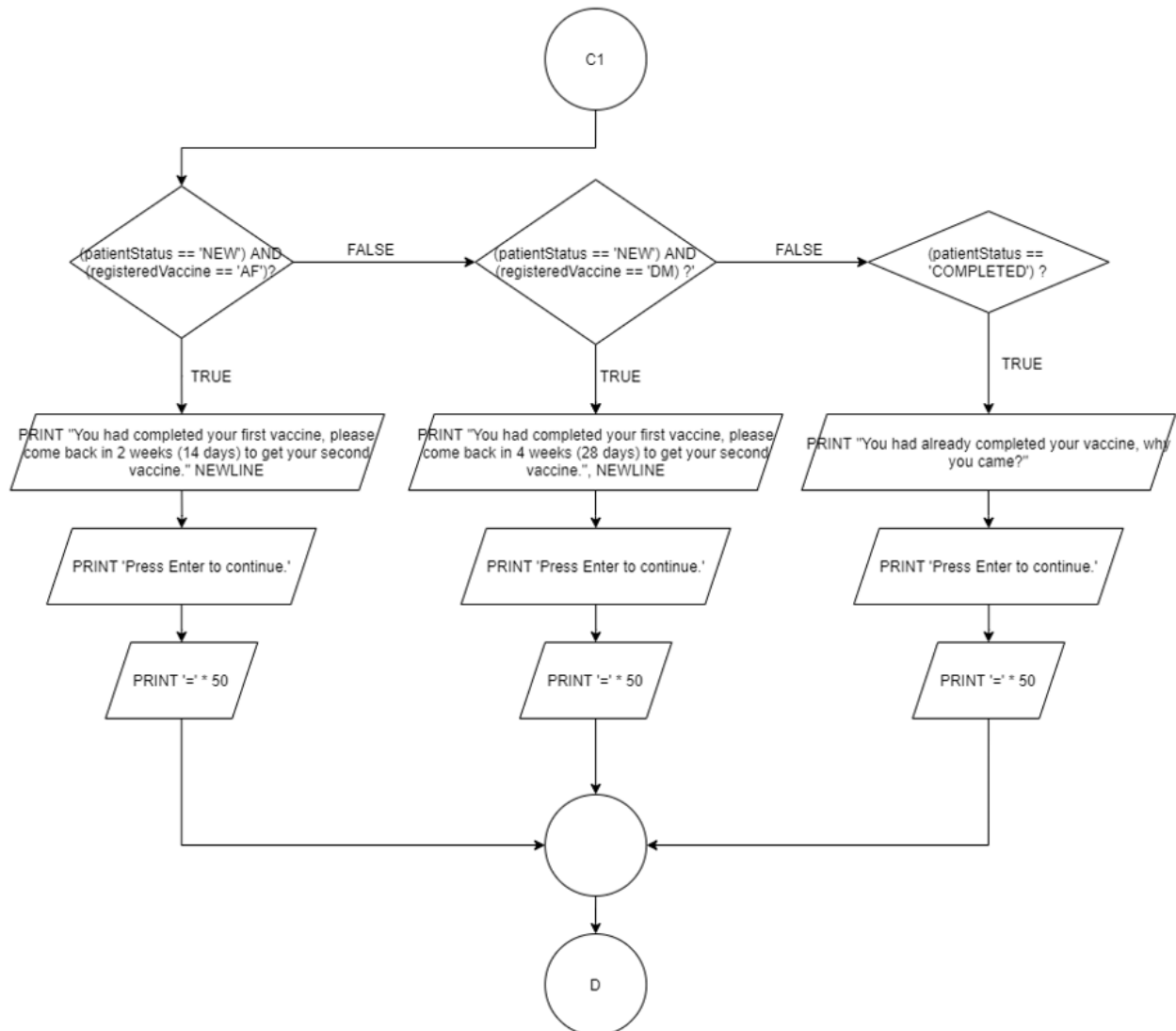


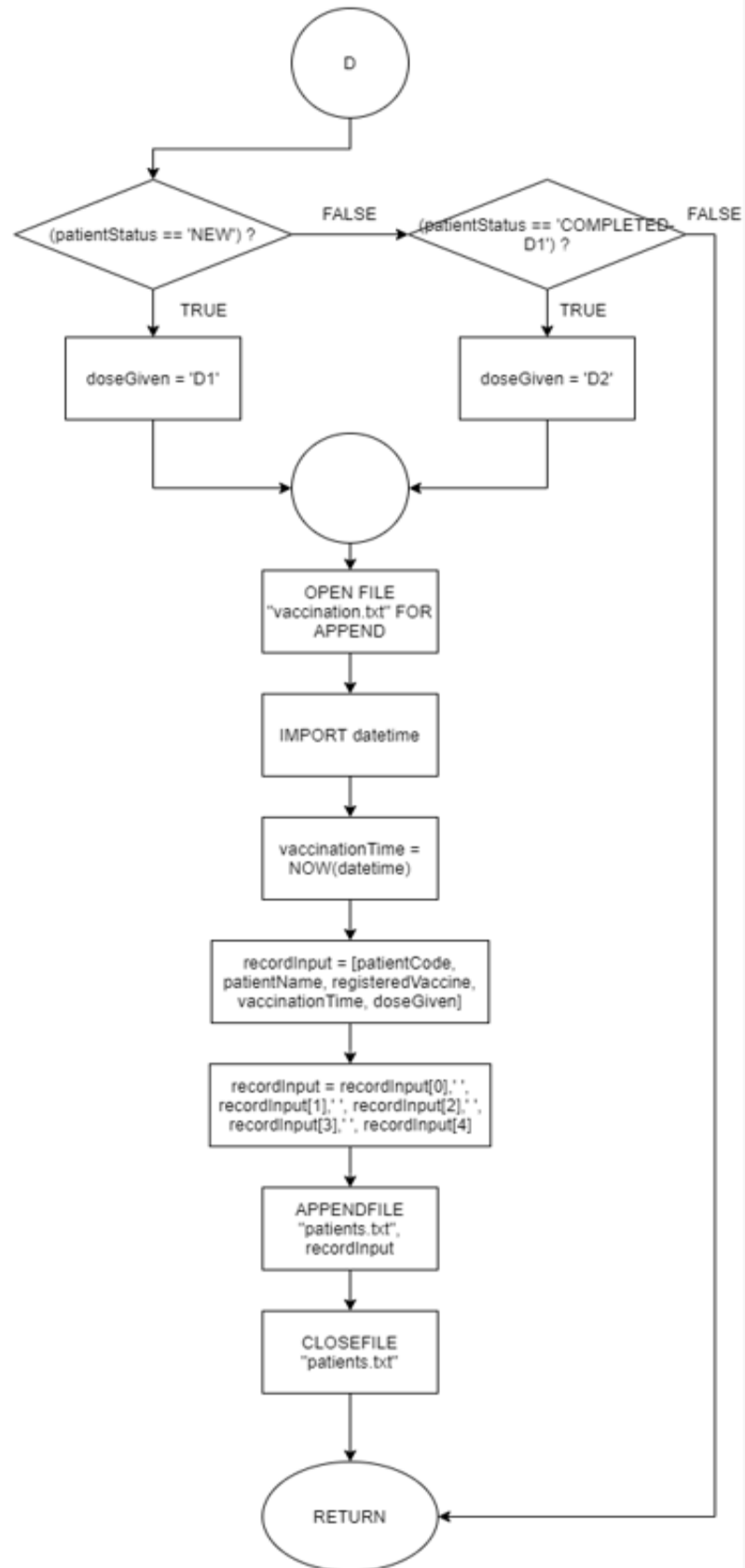




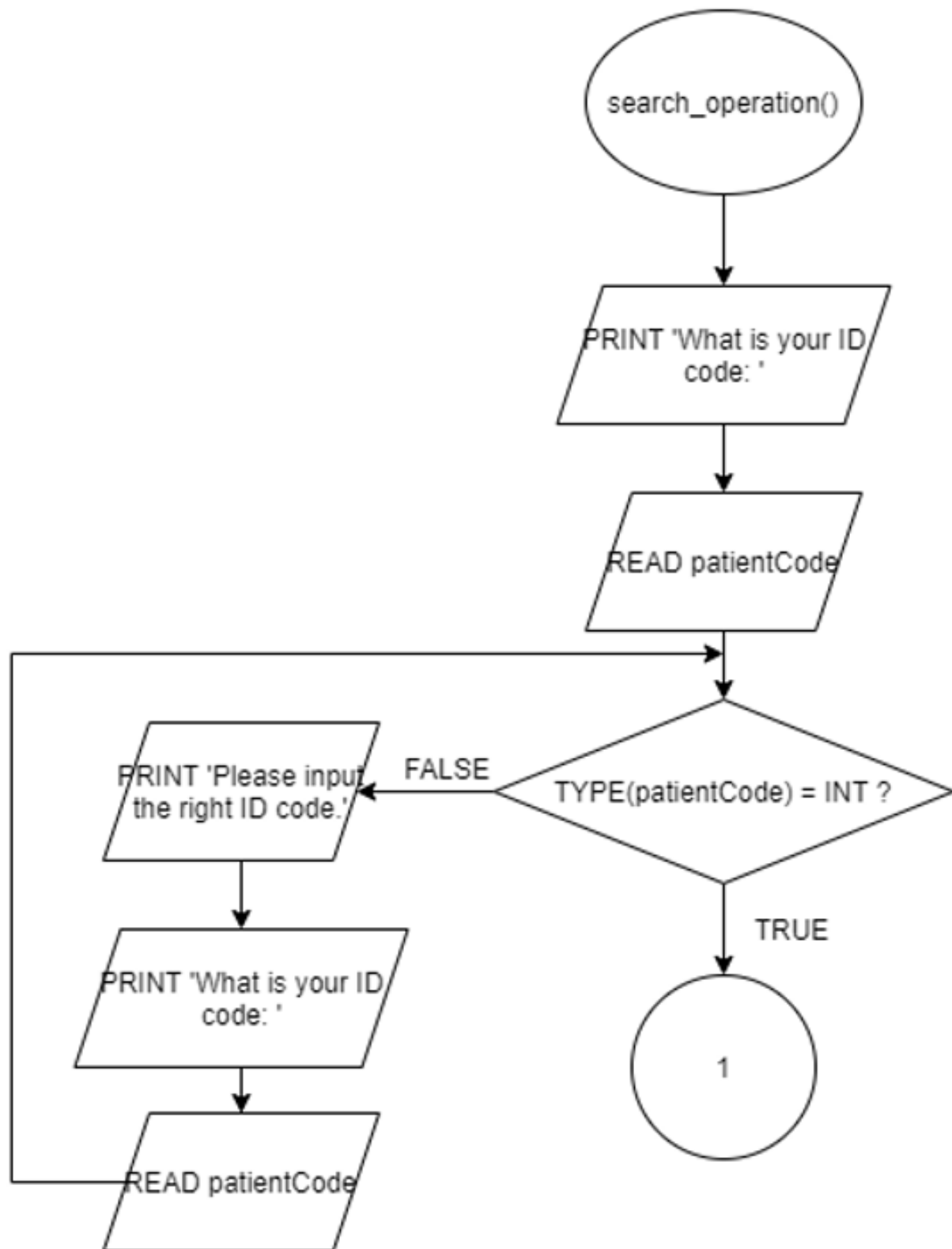


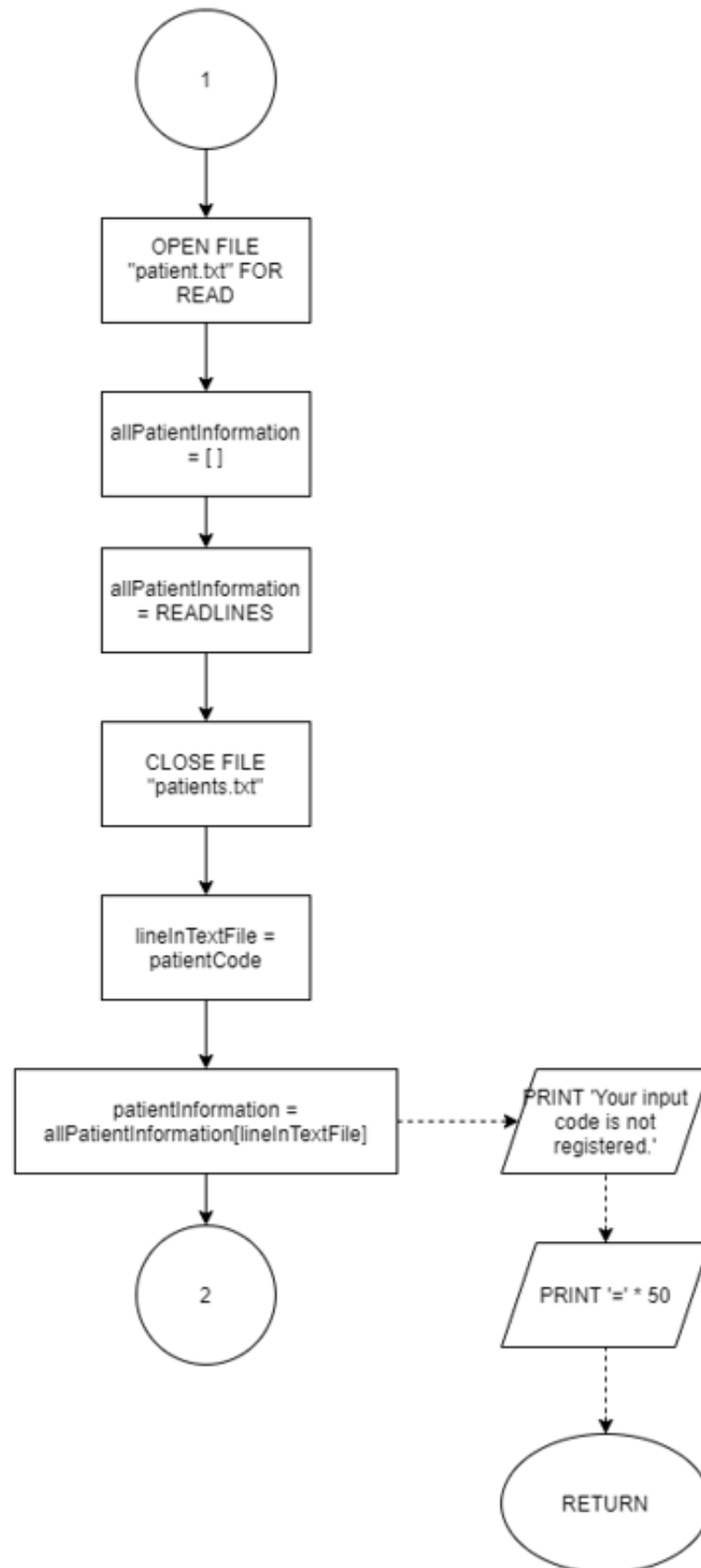


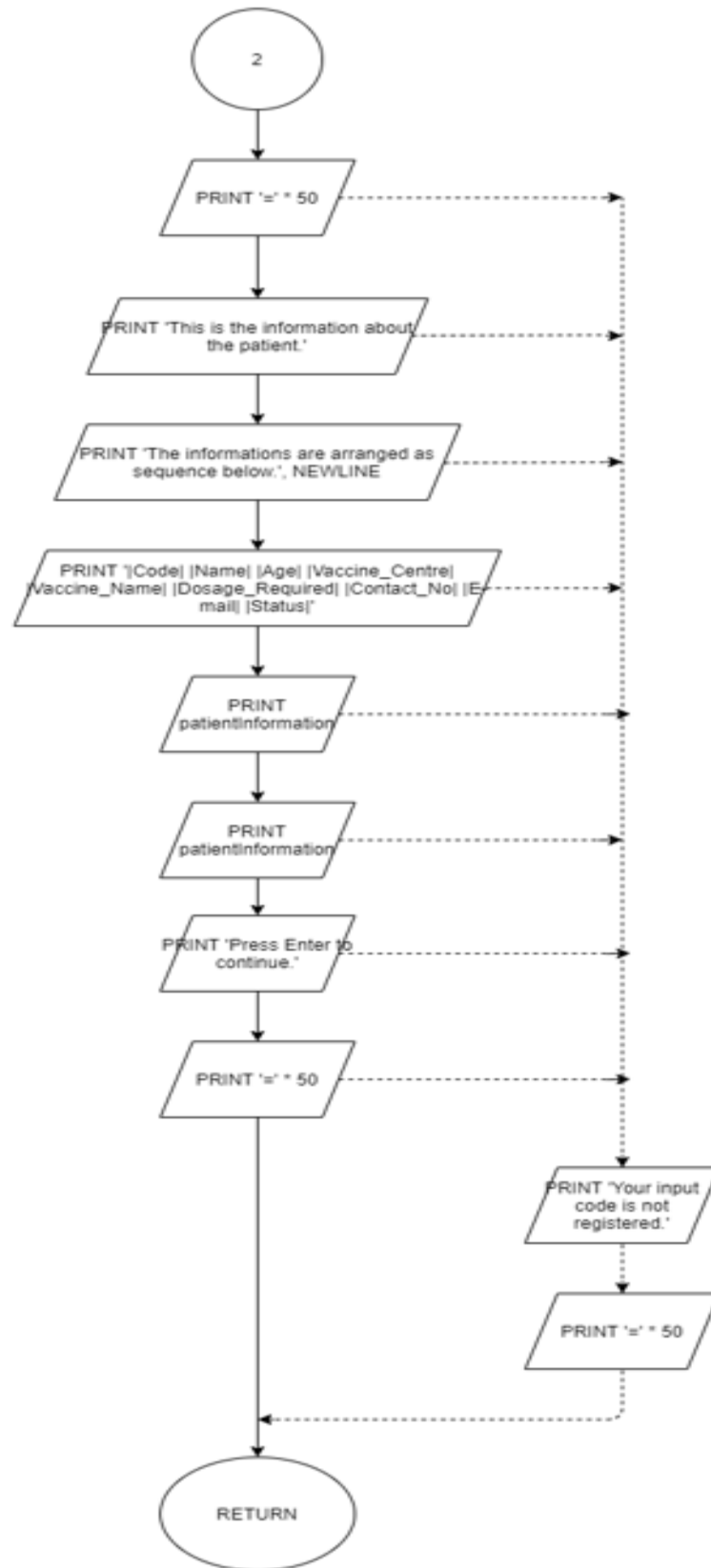




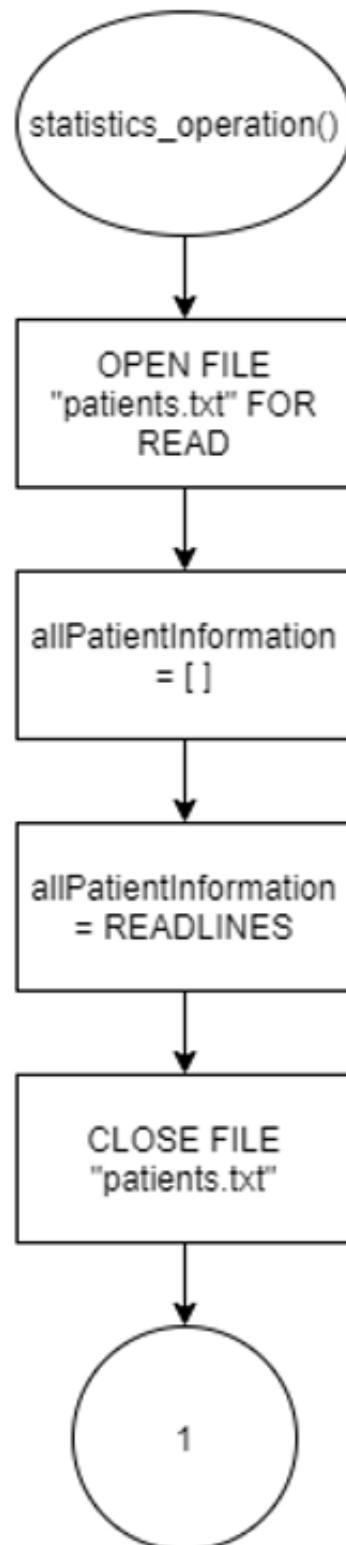
Search Function

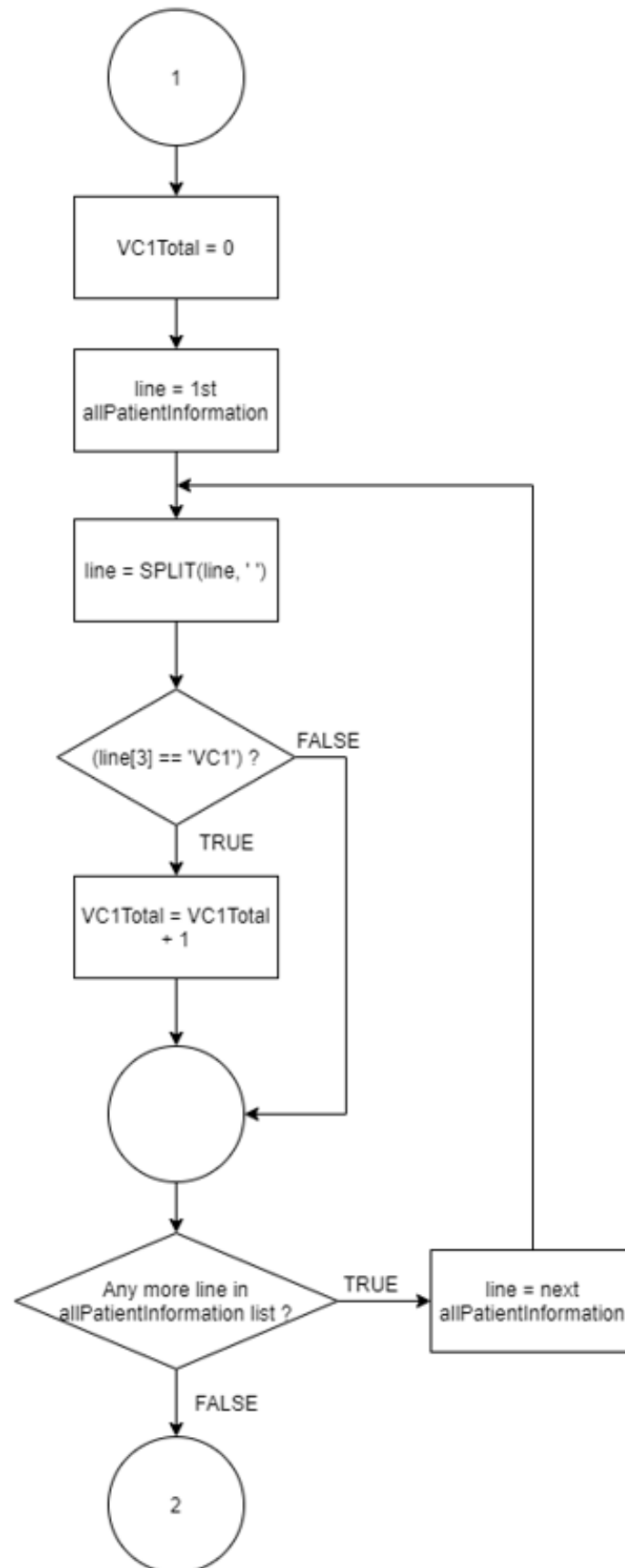


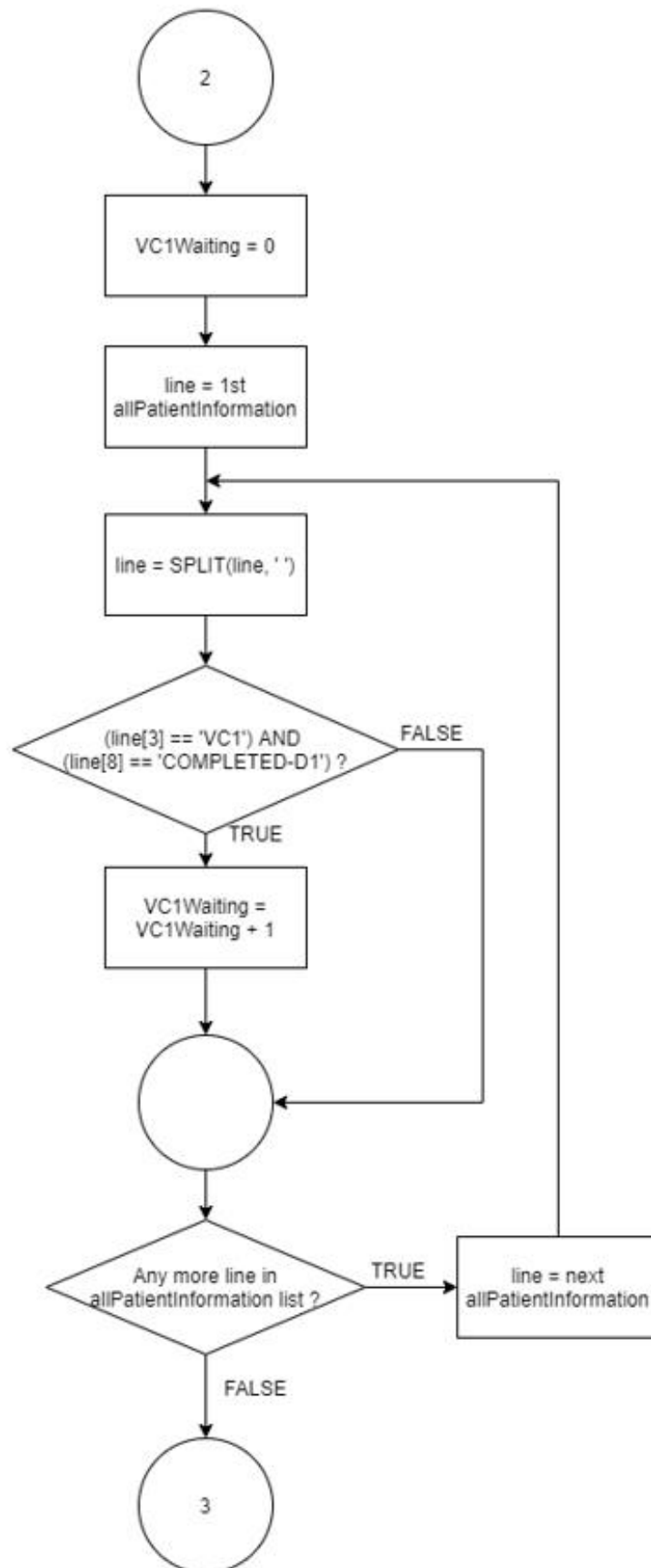


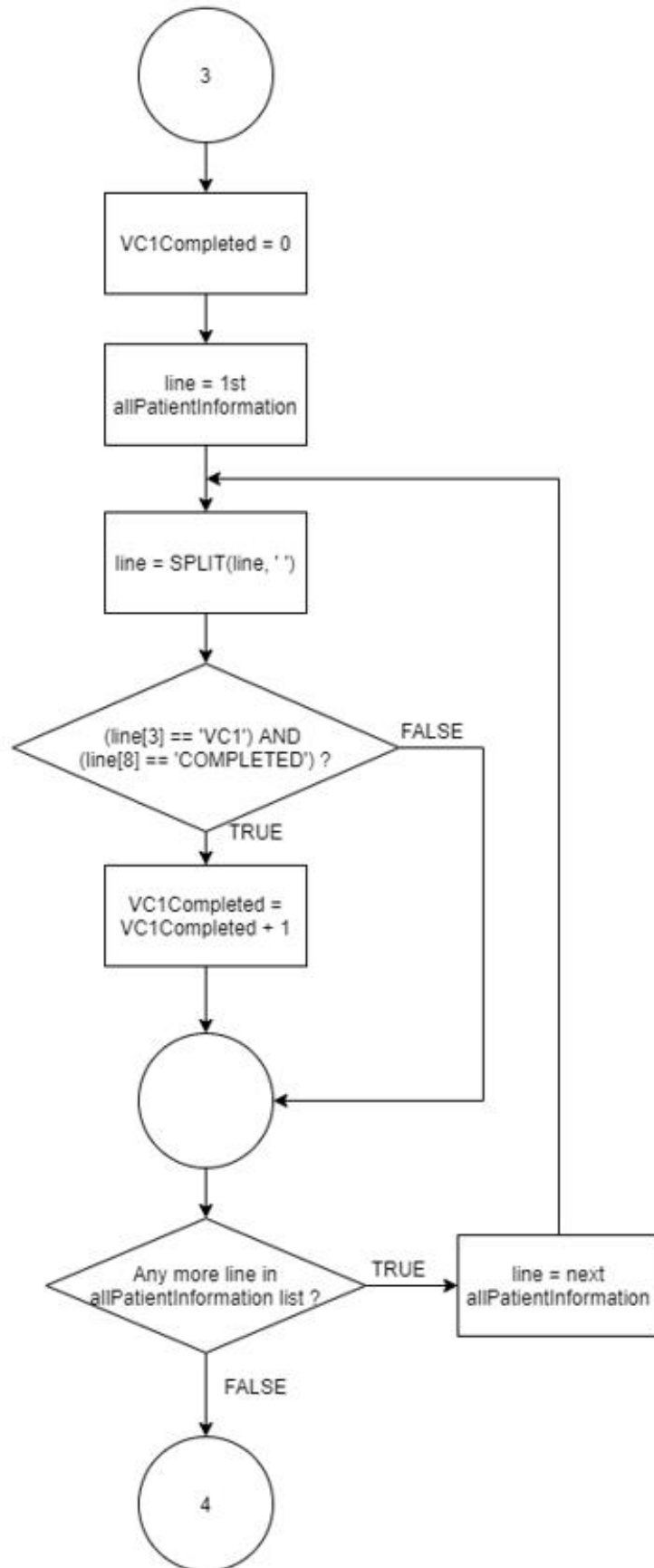


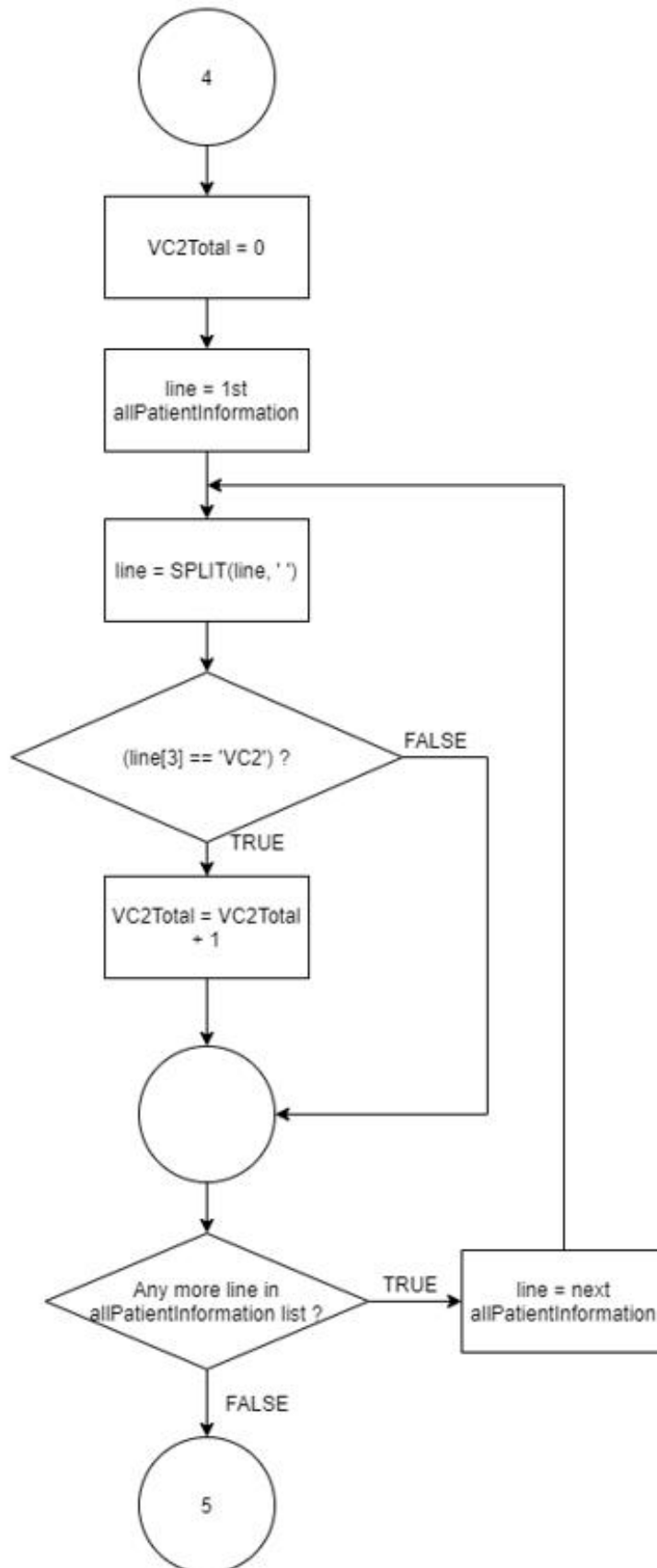
Statistics Function

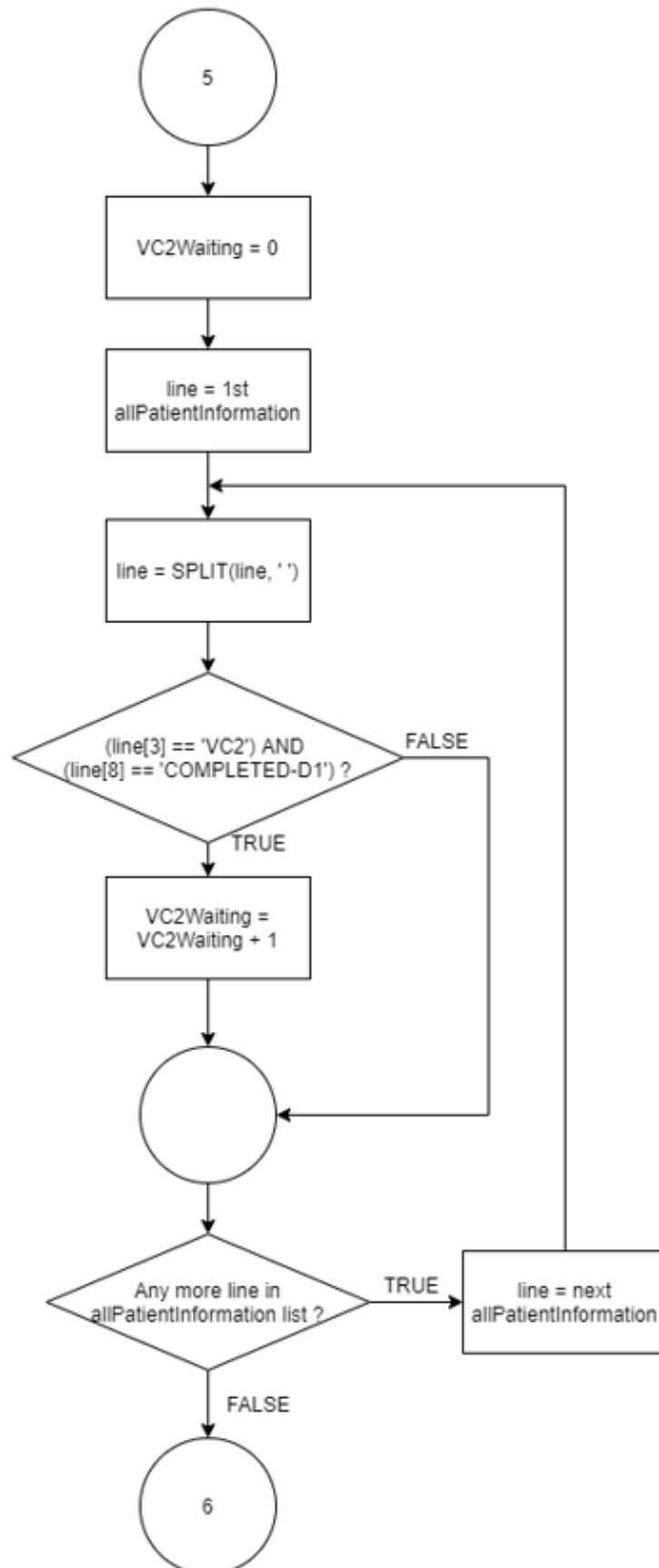


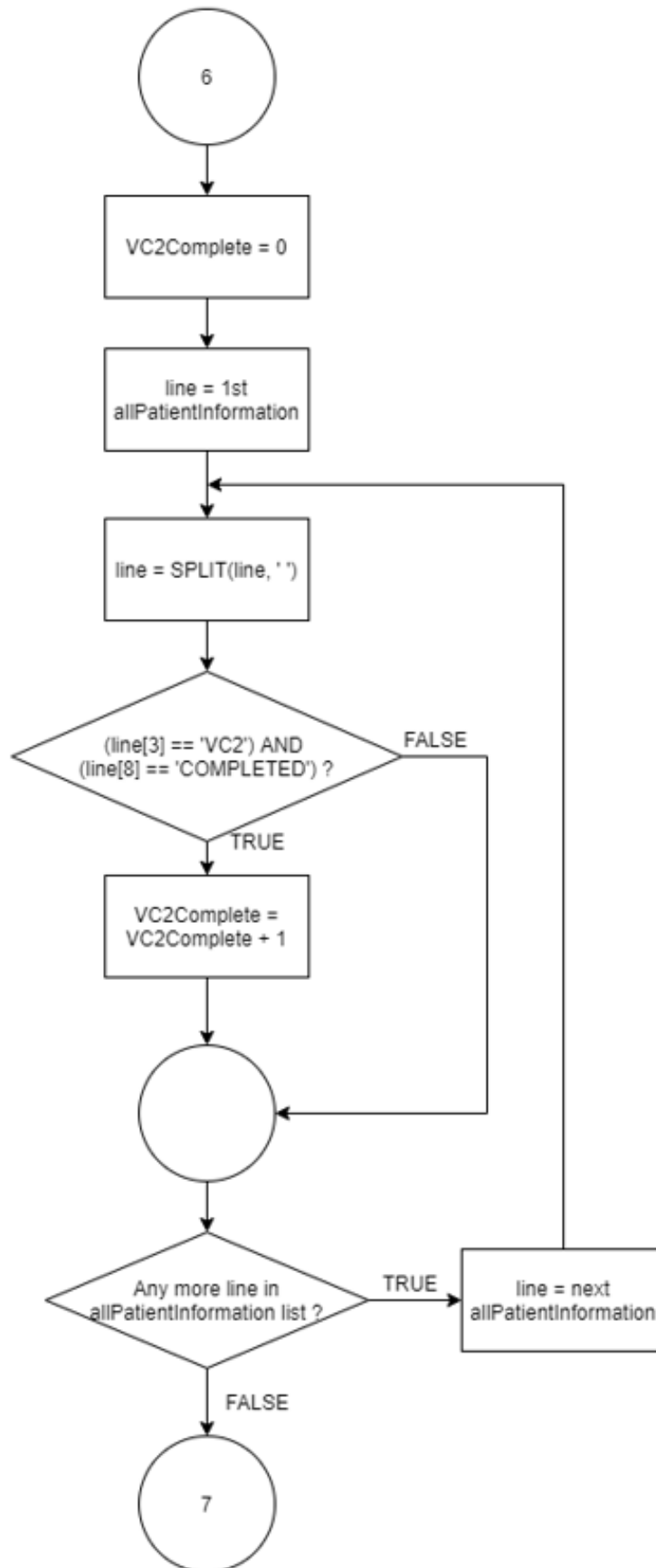


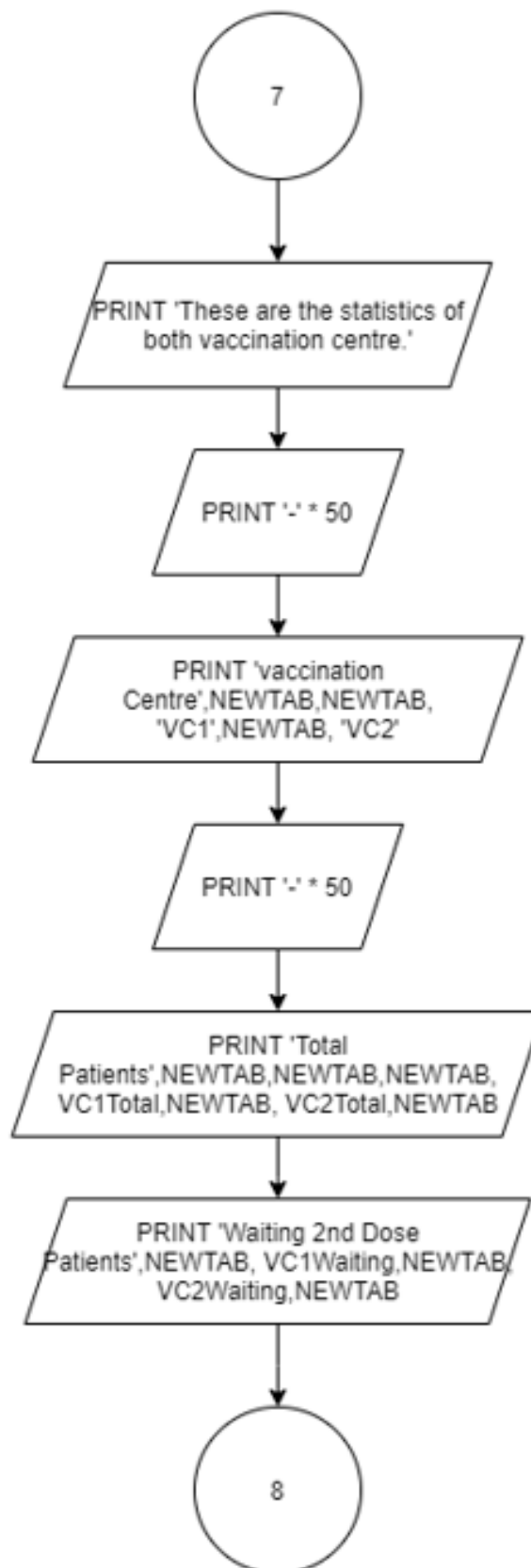


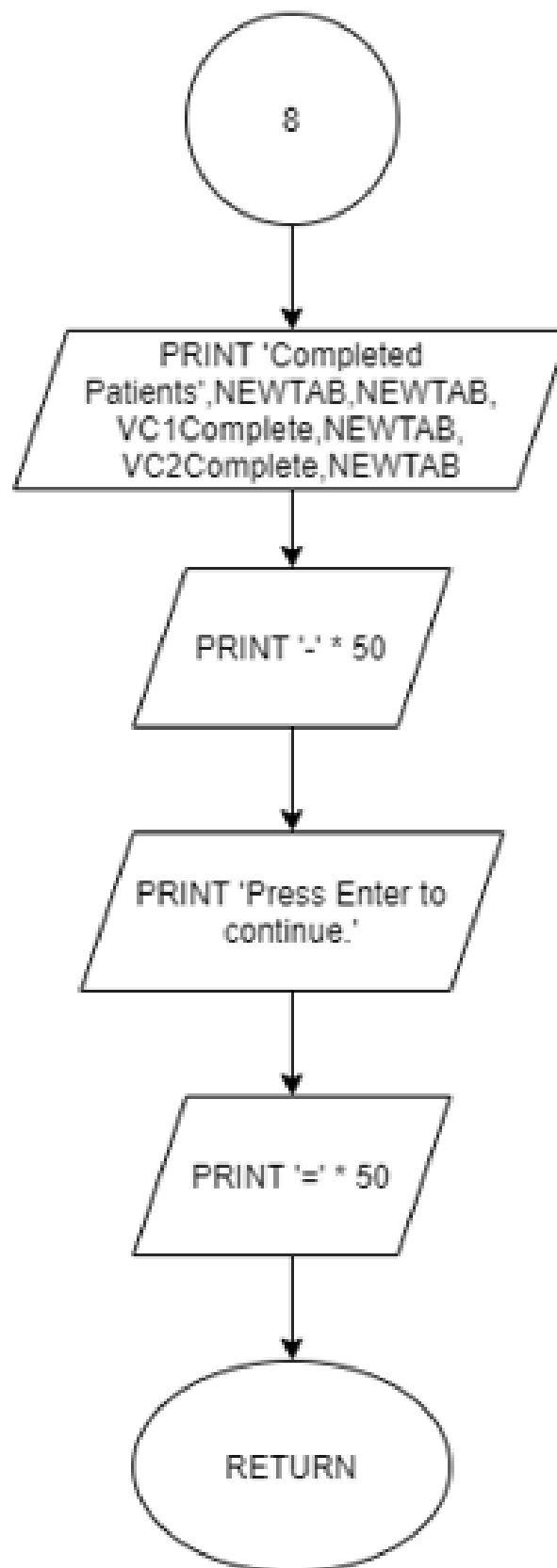




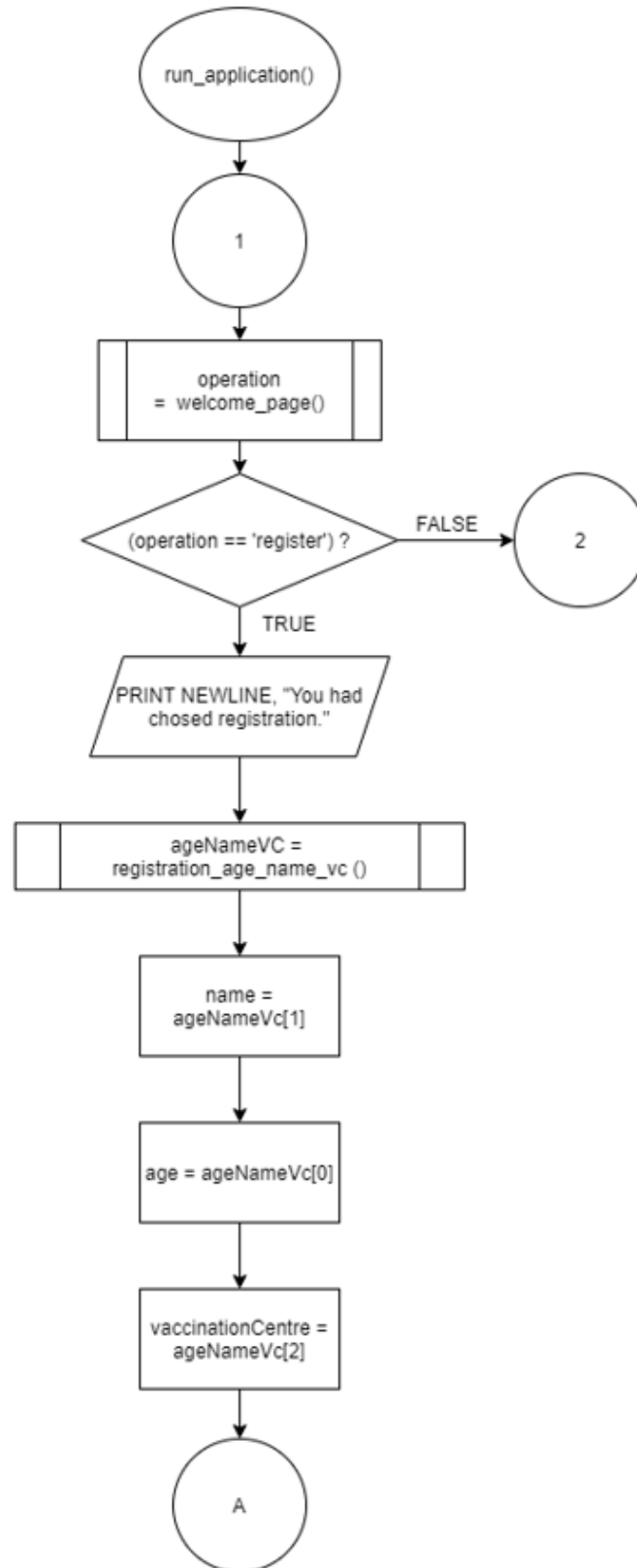


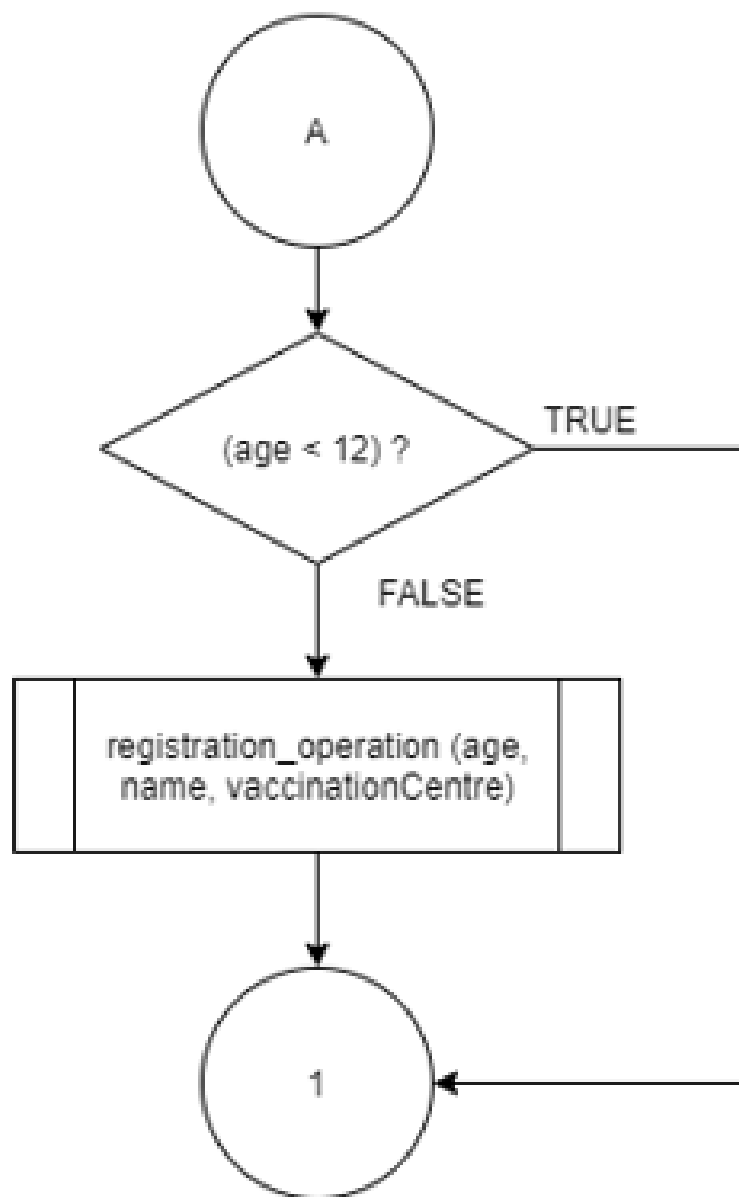


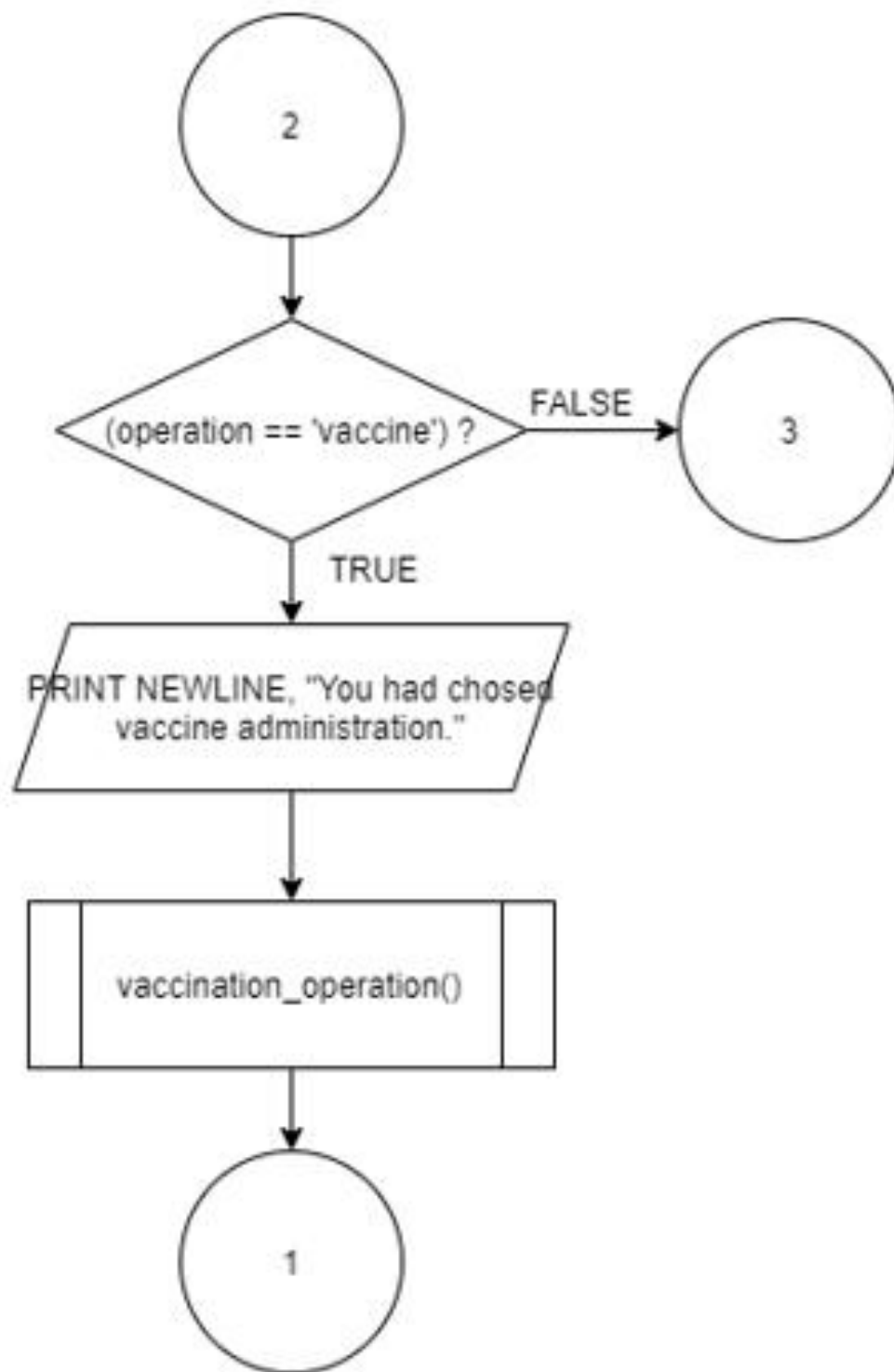


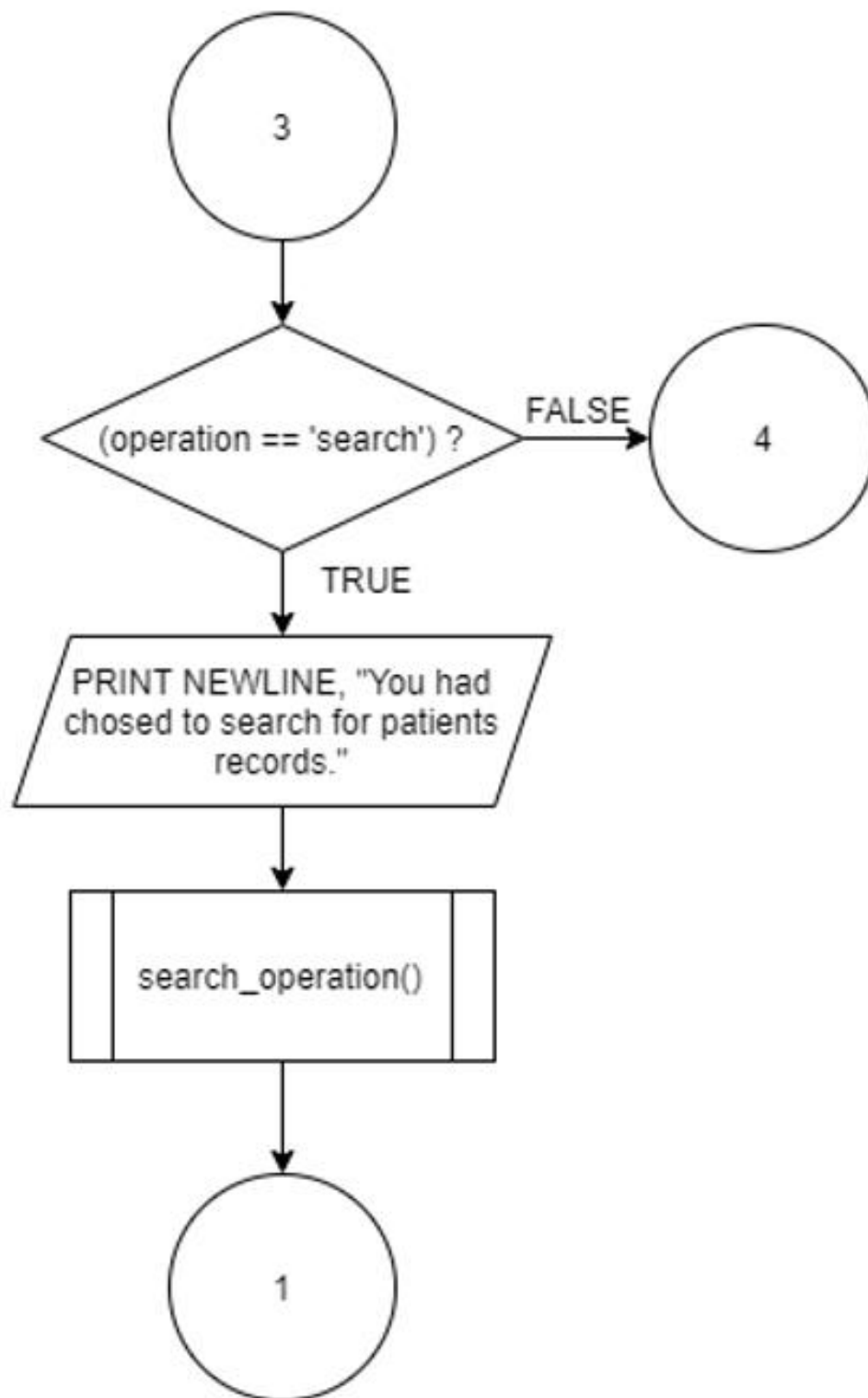


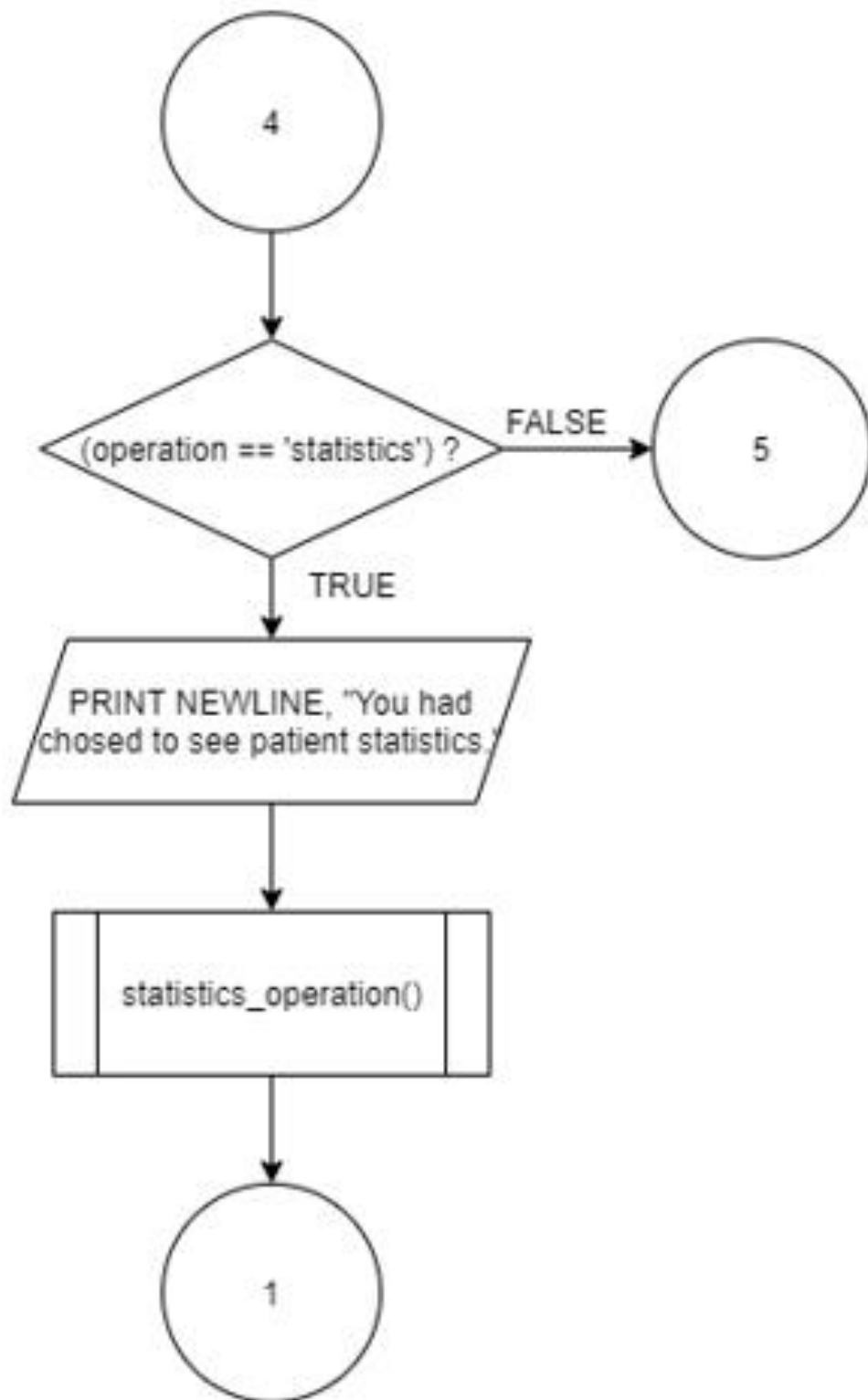
Run Application Function

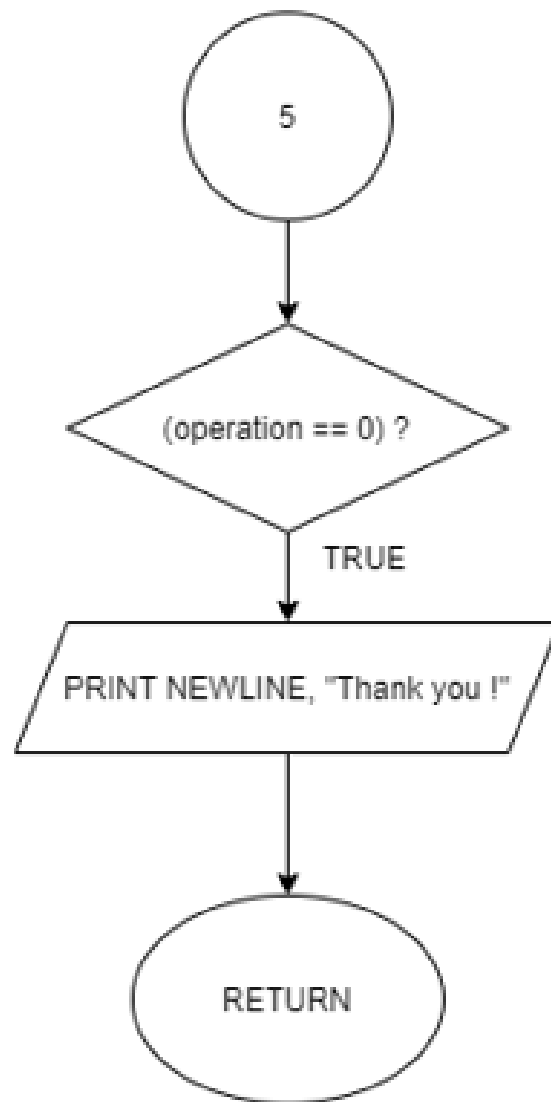




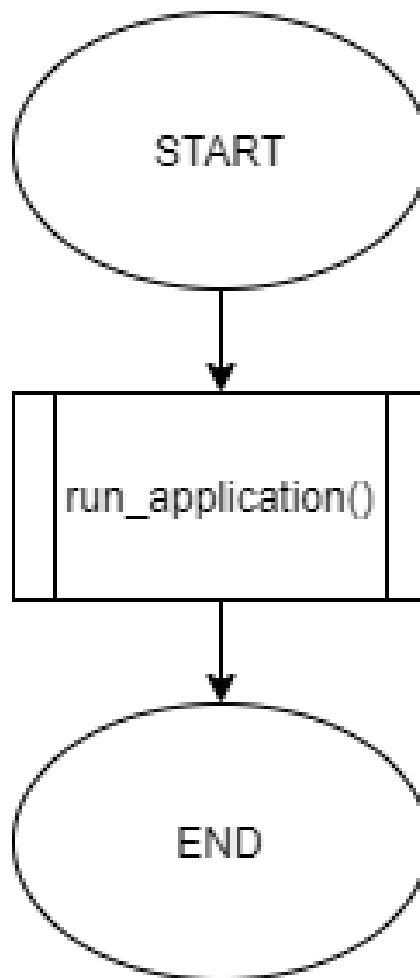








Main Code



Program Source Code and Explanation

The variable in this program is named using Camel Casing while the name of the function is named using Snake Case. They are named using separated method so that it is easier to be identified what are them. There will be 2 text files used in this program which are patients.txt and vaccination.txt. The purpose of patients.txt is to record the information of the patients including patient code, patient name, age, vaccine, dose amount, vaccination centre, email, contact number and status. While the function of vaccination.txt is to record patient code, patient name, vaccine, date, time and dose given.

Welcome Page Function

```
def welcome_page():
    operation = ''
    print('Welcome, please choose your operation.')
    print()
    print('Enter "register" to register new patient.')
    print('Enter "vaccine" for vaccine administration.')
    print('Enter "search" to search for patient record.')
    print('Enter "statistics" to see for patients statistics.')
    print('Enter "0" to quit.')
    print()
    while True: #Ensure no wrong input of operation
        operation = input('Please enter your operation here: ')
        if (operation != 'register') and (operation != 'vaccine') and (operation != 'search') and (operation != 'statistics') and (operation != '0'):
            print('Please input the right operation.')
            continue
        else:
            break
    return operation
```

This is the source code of the Welcome Page Function which is named as welcome_page(). First and foremost, the variable operation is set to be empty to remove the string that is stored when the program is looped again. Then the choice of operation is printed to tell the user what to input to choose the function that the user wants to run. The input function is placed inside a WHILE TRUE loop and the string contained inside is being checked using IF ELSE block to ensure that the user did not input anything other than the one is stated. The user will have to input again if his input is not one of the things that is stated. If it is, the user will break out of the loop and the string contained in the variable operation will be returned.

Registration Function

The whole Registration Function in this program is separated into 2 functions. Which are named as `registration_age_name_vc ()` and `registration_operation (age, name, vaccinationCentre)`. The second function will require the output of the first function.

```
def registration_age_name_vc ():  
    while True: #Ensure no wrong input of vaccine centre  
        vaccinationCentre = input ('Please choose your desired vaccination centre ["VC1" or "VC2"]: ')  
        if (vaccinationCentre != "VC1") and (vaccinationCentre != "VC2"):  
            print ('Please input the right code for vaccination centre.')  
            print()  
        else:  
            break  
  
    name = input ('Please input your name (without space, use "_" to indicate space): ')  
    name = name.replace(' ', '_')  
    while True: #Ensure no wrong input of age  
        try:  
            age = int (input('Please input your age: '))  
            break  
        except:  
            print ('Please input your age in numerical form.')  
  
    if (age<12):  
        print ('Sorry you are too young for vaccine.\n')  
        print ('=' * 50)  
    return age, name, vaccinationCentre
```

The name of the first function in the whole Registration Function is named as `registration_age_name_vc ()`. This function is only responsible to ask for the input of vaccination centre, age and name only. The input function of asking the input for vaccination centre is also put inside a WHILE TRUE loop and filtered with an IF ELSE block same like what is done in the Welcome Page Function. This is also to ensure no any inputs other than VC1 and VC2. The program will repeat asking for input for vaccination centre if it did not get a desired input.

Then the program will ask for name input, it is recommended to the user to input the name without space as later on in the program, the string that contain all the information of a patient will be broken down into a list by separating them when there is a space to run their function. Hence, to prevent the name will be separated into 2 elements, it is recommended to use underscore to replace space. Even if the user input name with space, the next line will replace the space with underscore using the replace function.

Then the program will ask for the user age and keep it in the 'age' variable. The input of the user is converted into integer and this process is also put inside a WHILE TRUE loop and inside a TRY

EXCEPT block to prevent the program from crashing and ensure that the input user gave is an integer. According to the question paper, there is no vaccine that is suitable for patient below 12 years old. Hence, when the user input an age that is lower than 12, the program will tell the user that he is not eligible to any vaccine.

After all of these the function will return 3 data which are 'age', 'name' and 'vaccinationCentre'.

```
def registration_operation (age, name, vaccinationCentre):  
    eMail = input('Please input your email: ')  
    contactNumber = input('Please input your contact number [without space]: ')  
  
    if (age >= 18) and (age <= 45):  
        while True: #Ensure no wrong input of vaccine name  
            print ('The vaccine suitable for you are "AF", "BV" "CZ", "DM" and "EC".')  
            vaccine = input('Please input the vaccine code that you want to have: ')  
            if (vaccine != "AF") and (vaccine != "BV") and (vaccine != "CZ") and (vaccine != "DM") and (vaccine != "EC"):  
                print ('Please input the right code for the vaccine.')  
                print ()  
            else:  
                break  
  
    if (age > 45):  
        while True: #Ensure no wrong input of vaccine name  
            print ('The vaccine suitable for you are "AF", "BV", "DM" and "EC".')  
            vaccine = input('Please input the vaccine code that you want to have: ')  
            if (vaccine != "AF") and (vaccine != "BV") and (vaccine != "DM") and (vaccine != "EC"):  
                print ('Please input the right code for the vaccine.')  
                print ()  
            else:  
                break  
  
    if (age >= 12) and (age < 18):  
        while True: #Ensure no wrong input of vaccine name  
            print ('The vaccine suitable for you are "AF", "CZ" and "DM".')  
            vaccine = input('Please input the vaccine code that you want to have: ')  
            if (vaccine != "AF") and (vaccine != "CZ") and (vaccine != "DM"):  
                print ('Please input the right code for the vaccine.')  
                print ()  
            else:  
                break
```

This is the second function in the whole Registration Function which is named as registration_operation (). This function needs 3 inputs from the last function which are 'age', 'name' and 'vaccinationCentre'. Firstly, the program will ask for the user's information which are the email and their contact number which will be stored inside the variable 'eMail' and 'contactNumber'. Then the age of the user that is stored inside the 'age' variable will go through 3 IF ELSE block to determine which vaccine is suitable for the patient according to the information given in the question paper. The suitable vaccines will be displayed to the user to let them make their choice and store it into the variable 'vaccine'. This process is also placed in a WHILE TRUE loop and then in an IF ELSE block to ensure that the vaccine is input by the user is what is suitable for them and not any other random string.

```

print ('=' *50)
if (vaccine == 'EC'):
    dosageRequired = '1Dose'
else:
    dosageRequired = '2Dose'
read_patient_information = open("patients.txt", "r")
patient = read_patient_information.readlines()
patientAmount = len(patient) - 1
read_patient_information.close()
patientCode = str(patientAmount + 1).zfill(4)
print ('Your code is', patientCode, 'Please remember your code.')

append_patient_information = open("patients.txt", "a")
informationInput = '\n' + str(patientCode) + ' ' + name + ' ' + str(age) + ' ' + vaccinationCentre + ' '
+ vaccine + ' ' + dosageRequired + ' ' + contactNumber + ' ' + eMail + ' ' + 'NEW'
append_patient_information.write(informationInput)
append_patient_information.close()

print ('Thanks for registering', name, '.', 'Your registered vaccine is', vaccine)
print ('Your code is ', patientCode)
input ('Press Enter to continue.') #This is to let the user to have time to look at the information
print ('=' * 50)

```

Then the program will check the vaccine choice of the user and determine how many doses should be given to be recorded into patients.txt file. According to the question paper vaccine EC only need 1 dose. Hence, an IF ELSE block is used to check the string in the variable 'vaccine' and if the string contained is EC the variable 'dosageRequired' will be 1Dose. Else it will be 2Dose. Then the file patients.txt is opened to read. Then the READLINES function is used to read everything in the file and every line in the file is an element of a list. The list is named as 'patient', Then the total amount of patient will be calculated using the LEN function to calculate how many elements are there in the 'patient' list. The number will be minus by 1 because the first line in the text file is not patient's information while it is telling the arrangement of the information.

1	[Code]	[Name]	[Age]	[Vaccine_Centre]	[Vaccine_Name]	[Dosage_Required]	[Contact_No]	[E-mail]	[Status]
2	0001	Adam	18	VC2	EC	1Dose	0122367886	Adam@gmail.com	COMPLETED
3	0002	Lucas	20	VC1	CZ	2Dose	0122367347	Lucas@gmail.com	COMPLETED-D1

Then the file is closed. Then the patient code is generated by adding 1 to the 'patientAmount' variable and converted into a string of 4 digit using ZFILL function. As an example, if the number is 4, then after using the function zfill(4), it will become 0004. This will be used as the patient code. Then the file patients.txt is opened again for append. All of the information of the patient will be combined into a string and stored inside a variable called 'informationInput', the reason of having \n at the beginning is to make sure that the information is recorded in the new line in the file. Every line in the file should only contain 1 patient's information. Then the variable 'informationInput' is appended into the file and the

file is closed. Then the patient code is displayed to the user and he is reminded to remember the patient code as it is needed during vaccination function and search function.

Vaccination Function

```
def vaccination_operation ():  
    while True:  
        registration = input('Had you done your registration? ["yes" if yes, "no" if no]: ')  
        if (registration != 'yes') and (registration != 'no'): #Ensure no wrong input  
            print ('Please input the right answer.')  
            continue  
        elif (registration == 'no'):  
            ageNameVc = registration_age_name_vc ()  
            name = ageNameVc[1]  
            age = ageNameVc[0]  
            vaccinationCentre = ageNameVc[2]  
            if (age < 12):  
                break  
            registration_operation(age, name, vaccinationCentre)  
            continue  
        elif (registration == 'yes'):  
            try: #This is to prevent user enter code that is not registered and crash the program  
                while True:  
                    try:  
                        patientCode = int(input('what is your ID code: '))  
                        break  
                    except:  
                        print('Please input the right ID code.')
```

The function is named as `vaccination_operation ()`. According to the question paper the patient that want to get vaccinated doesn't necessarily have to been registered before. So, when the user chooses the vaccination function, the program will ask the user if he had registered or not. The input is also placed in a `WHILE TRUE` loop and checked by `IF ELSE` block to ensure no any input other than yes and no. The user has to input yes if he had registered before and no if he had not.

If the input is no, the program will call the Registration Function which are `registration_age_name_vc ()` and `registration_operation ()`. The `registration_age_name_vc ()` will be called first and the list 'ageNameVc' will contain the age, name and vaccination centre of the patient that is returned. Then the element in the list then transferred to 3 new variables to contain each element which are 'name', 'age' and 'vaccinationCentre'. Then the integer contained in 'age' is checked if it is lesser than 12. If the age is lesser than 12, the program will exit from the loop and go back to the Welcome Page Function as there are no suitable vaccine for patients younger than age 12. If the age is more than 12, then the program will proceed to `registration_function ()`. After done registering the program will go back to ask the user had registered or not. Then the user will input yes as he already registered just now.

Then when the user input yes, the program will proceed to the vaccination process. The whole code that run after the user input yes is placed inside a TRY EXCEPT block to prevent the user input a patient code that is not available in the patients.txt file and causes the program to crash. The program will first ask for the patient's code and this process is also placed in a WHILE TRUE loop and checked with IF ELSE block to ensure that the input is an integer. If not, the program will ask again.

```
read_patient_information = open("patients.txt", "r")
allPatientInformation = read_patient_information.readlines() #
read_patient_information.close
lineInTextFile = int(patientCode) #lineInTextFile is the line
patientInformation = allPatientInformation[lineInTextFile] #The
patientNewInformation = patientInformation #The same informati
patientInformation = patientInformation.split() #The informati
patientName = patientInformation [1] #The information in the l
print ('Hello', patientName)
registeredVaccine = patientInformation [4]
print ('Your vaccine registered is', registeredVaccine)
patientStatus = patientInformation [8]
print ('Your vaccine status was', patientStatus)
```

Then the file patient.txt is opened and READLINES function is used and every line in the file is an element of a list and it is store in 'allPatientInformation'. Then the integer stored in the 'patientCode' is also copied to another variable which is 'lineInTextFile'. This variable will store the location of the patient in the 'allPatientInformation' list. Then the particular patient's information is extracted out from the 'allPatientInformation' list and stored in the variable 'patientInformation'. The string in 'patientInformation' is also stored in a new variable 'patientNewInformation' to be edited and update or replace the old information of the particular patient later. Then the string variable 'patientInformation' is split into a list by separating them when there is a space in the string. As an example, if the list is 'XinHong 18', it will be split into ['XinHong', 18]. This will produce a nested list. Then all the patient's information is extracted from the list and stored in different variables such as 'patientName', 'registeredVaccine' and 'patientStatus' to be displayed to the patient and be used later on.

```

patientNewInformation = patientNewInformation.split()
#The patient vaccine status were updated
if (registeredVaccine == 'EC'):
    patientNewInformation [8] = 'COMPLETED'
elif (patientStatus == 'COMPLETED-D1'):
    patientNewInformation [8] = 'COMPLETED'
elif (patientStatus == 'NEW'):
    patientNewInformation [8] = 'COMPLETED-D1'
elif (patientStatus == 'COMPLETED'):
    patientNewInformation [8] = 'COMPLETED'

read_patient_information = open("patients.txt", "r")
patient = read_patient_information.readlines()
patientAmount = len(patient) - 1
read_patient_information.close()
print()
#This is to prevent 2 patients information to be continued after another in the text file
if (patientAmount != int(patientNewInformation [0])) and (patientNewInformation [8] == 'COMPLETED'):
    patientNewInformation [8] = 'COMPLETED\n'
elif (patientAmount != int(patientNewInformation [0])) and (patientNewInformation [8] == 'COMPLETED-D1'):
    patientNewInformation [8] = 'COMPLETED-D1\n'

allPatientInformation[lineInTextFile] = patientNewInformation #A nested list is created
allPatientInformation[lineInTextFile] = ' '.join(allPatientInformation[lineInTextFile]) #The list is joined to let user have a clearer view in the file
write_patient_information = open("patients.txt", "w")
write_patient_information.writelines(allPatientInformation)
write_patient_information.close()
print ('='*50)

```

Then the string variable 'patientNewInformation' is also split into a list so that the program can edit the information that is recorded. The program will check the original condition of the patient and update it. As an example, if the patient condition is NEW then the program will update it to COMPLETED if the patient's registered vaccine is EC and to COMPLETED-D1 if it is other vaccine. If the condition is COMPLETED-D1, the program will update it to COMPLETED.

Then the file patients.txt is opened again to read how many patients is recorded in the file. In order to prevent 2 patients' information is stored in one line in the file, a \n or newline have to be added at the last element in the list. However, this is not needed if the patient the last in the file because it will make an empty line if \n is added to the last patient recorded in the file and causing the program to have bug and produce patient code that is not in sequence. Hence, the program will check if the patient is the last one or not with IF ELSE block.

Then the list 'patientNewInformation' is joined together with space to become a string using the JOIN function. As an example, if the list is ['XinHong', 18], it will become 'XinHong 18'.

Then the 'patientNewInformation' string is used to replace the old element that contain the old patient's information that is contained in the 'allPatientInformation' list. This will create a nested list. In order to make the user be able to read clearer in the text file without brackets and commas, the nested list removed by joining the elements using space using JOIN function. Then the file patients.txt is opened and the

whole list is used to overwrite the old information in the file. After all of this the file is closed and a line of 50 '=' is printed to make the interface clearer.

```
if (patientStatus == 'COMPLETED-D1'):
    print ("congratulations, you had completed your vaccine, you don't have to come again.", '\n')
    input ('Press Enter to continue.') #This is to let the user to have time to look at the information
    print ('='*50)
elif (patientStatus == 'NEW') and (registeredVaccine == 'EC'):
    print ("congratulations, you had completed your vaccine, you don't have to come again.", '\n')
    input ('Press Enter to continue.') #This is to let the user to have time to look at the information
    print ('='*50)
elif (patientStatus == 'NEW') and ((registeredVaccine == 'BV') or (registeredVaccine == 'CZ')):
    print ('You had completed your first vaccine, please come back in 3 weeks (21 days) to get your second vaccine.', '\n')
    input ('Press Enter to continue.') #This is to let the user to have time to look at the information
    print ('='*50)
elif (patientStatus == 'NEW') and (registeredVaccine == 'AF'):
    print ('You had completed your first vaccine, please come back in 2 weeks (14 days) to get your second vaccine.', '\n')
    input ('Press Enter to continue.') #This is to let the user to have time to look at the information
    print ('='*50)
elif (patientStatus == 'NEW') and (registeredVaccine == 'DM'):
    print ('You had completed your first vaccine, please come back in 4 weeks (28 days) to get your second vaccine.', '\n')
    input ('Press Enter to continue.') #This is to let the user to have time to look at the information
    print ('='*50)
elif (patientStatus == 'COMPLETED'):
    print ('You had already completed your vaccine, why you came?', '\n')
    input ('Press Enter to continue.') #This is to let the user to have time to look at the information
    print ('='*50)
```

Then the patient is given information about the interval between first and second dose according to what is their vaccine and what is their status. This process is run using nested if. This will increase the program's efficiency. Their vaccine and status are checked by IF ELSE block and the corresponding instruction will be given according to it. The patient that had completed 1 dose which is not EC vaccine were advised to come back after the interval that is given in the question paper. And if the patient completed their dose or get 1 dose for EC is advised to not come again as they already finish their vaccination. Lastly to prevent the program to have bugs the patient that had come after they finish their vaccination will also be informed that they had already finish their vaccination and they do not need to come again.

```
if (patientStatus == 'NEW'):
    doseGiven = 'D1'
elif (patientStatus == 'COMPLETED-D1'):
    doseGiven = 'D2'

if (patientStatus == 'NEW') or (patientStatus == 'COMPLETED-D1'):
    record_patient_information = open("vaccination.txt", "a")
    from datetime import datetime
    vaccinationTime = datetime.now()
    vaccinationTime = vaccinationTime.strftime("%d/%m/%Y | %H:%M:%S")
    patientCode = str(patientCode).zfill(4)
    recordInput = ['\n', patientCode, patientName, registeredVaccine, vaccinationTime, doseGiven]
    recordInput = ' '.join(recordInput)
    record_patient_information.write(recordInput)
    read_patient_information.close()
    break
except:
    print ('Your input code is not registered.')
    print ('='*50)
    break
```

The dose given to the patient will be recorded in the file vaccination.txt. This also include their code, name, vaccine, time and which dose is given. At first the program will check what is the status of the patient and determine which dose should be given. Then the file vaccination.txt is opened for append. DATETIME function is imported to generate date and time when the vaccine is given. The integer that is contained in 'patientCode' is also converted into string and ZFILL function is used again to make the code 4 digits. Then all of the information is combined into a list and contained in 'recordInput' then they are also joined with space using JOIN function. Lastly, the string 'recordInput' will be appended into vaccination.txt file.

The EXCEPT block is to catch the program when the program crashes when the user input the code that is not registered in patients.txt file that will crashes the program. The user is also informed that the code input is not registered in the patients.txt file. When this happens, the program will exit the loop in Vaccination Function and go back to Welcome Page Function.

Search Function

```
def search_operation ():  
    while True:  
        try:  
            patientCode = int(input('What is your ID code: '))  
            break  
        except:  
            print('Please input the right ID code.')  
  
        try: #This is to prevent user enter code that is not registered and crash the program  
            read_patient_information = open("patients.txt", "r")  
            allPatientInformation = read_patient_information.readlines()  
            read_patient_information.close()  
            lineInTextFile = int(patientCode)  
            patientInformation = allPatientInformation[lineInTextFile]  
            print ('='*50)  
            print ('This is the information about the patient.')  
            print ('The informations are arranged as sequence below.\n')  
            print ('|Code| |Name| |Age| |Vaccine_Centre| |Vaccine_Name| |Dosage_Required| |Contact_No| |E-mail| |Status|')  
            print (patientInformation)  
            input('Press Enter to continue.') #This is to let the user to have time to look at the information  
            print ('='*50)  
        except:  
            print ('Your input code is not registered.')  
            print ('='*50)
```

This function is named as `search_operation ()`. The main task of this function is to check the status of the patient which their code is input in the program. Firstly, the program will ask for the patient code that the user wants to check. Same as what is done in the previous functions, this process is placed in a `WHILE TRUE` loop and checked with `IF ELSE` block to ensure that the input is an integer. If not, the program will keep on asking for another input.

Then the process of searching the line where the patient's information locates is placed in `TRY EXCEPT` block to prevent the user to input a code that is not registered and out of range which will cause the program to crash. Firstly, the `patients.txt` file is opened for read and `READLINES` function is used to transfer the content into 'allPatientInformation' in a list form. Each line in the file is an element of the list. Then the input code by the user is converted into integer and stored in 'lineInTextFile' which contain the location of the patient's information in the list. Then the patient's information is extracted from the list. Then the arrangement of the information is displayed to the user and so does the information of the patient. The input function after that is to let user have time to read the information before the program continue the loop and go back to Welcome Page Function. If the input code is not registered, the program will crash and go to the except block and inform the user that the code is not registered.

Statistics Function

```
def statistics_operation ():  
    read_patient_information = open("patients.txt", "r")  
    allPatientInformation = read_patient_information.readlines()  
    read_patient_information.close()  
  
    VC1Total = 0  
    for line in allPatientInformation:  
        line = line.split() #nested list is created  
        try:  
            if (line[3] == 'VC1'):  
                VC1Total = VC1Total + 1  
        except:  
            continue
```

The name of this function is `statistics_operation ()`. To analyze all the patient's information, the content in the `patients.txt` file has to be extracted and stored as list in 'allPatientInformation' using `READLINES` function. Every line in the file is an element of the list. The first thing that will be calculated in this program is total number of registered patients in vaccination centre 1. The value will be stored in 'VC1Total'. 'VC1Total' has to be set to 0 before the counting process to reset the counter. The process is run using enhanced for loop. The elements in 'allPatientInformation' are a string. For every element in 'allPatientInformation' they will be split using `split` function into a list as what is done in Vaccination Function. The example is from 'XinHong 18' to ['XinHong', 18]. This will make the list 'allPatientInformation' into a nested list to increase the efficiency of the program.

Then since the arrangement of the information is the same in every line. Hence, the program doesn't need to check for every element in all lines or elements in 'allPatientInformation' while only need to check the particular location in which store the information that is needed. In this case is VC1. Thus, when there is a VC1 found in the location [3] in each line, the counter will add 1 to the counter 'VC1Total'. This is a very good application of nested list to increase the program's efficiency. The enhanced for loop will end when there is no more element or line in 'allPatientInformation'. This process is also placed in `TRY EXCEPT` block to prevent the program from crashing.

```
VC1Waiting = 0
for line in allPatientInformation:
    line = line.split() #nested list is created
    try:
        if (line[3] == 'VC1') and (line[8] == 'COMPLETED-D1'):
            VC1Waiting = VC1Waiting + 1
    except:
        continue

VC1Complete = 0
for line in allPatientInformation:
    line = line.split() #nested list is created
    try:
        if (line[3] == 'VC1') and (line[8] == 'COMPLETED'):
            VC1Complete = VC1Complete + 1
    except:
        continue

VC2Total = 0
for line in allPatientInformation:
    line = line.split() #nested list is created
    try:
        if (line[3] == 'VC2'):
            VC2Total = VC2Total + 1
    except:
        continue

VC2Waiting = 0
for line in allPatientInformation:
    line = line.split() #nested list is created
    try:
        if (line[3] == 'VC2') and (line[8] == 'COMPLETED-D1'):
            VC2Waiting = VC2Waiting + 1
    except:
        continue
```

The process is all the same for the rest just the requirements for their counter to add 1 is different. As an example, when the program wants to count the total patient waiting for second dose in vaccination centre 2, the program will check for VC2 at location [3] and COMPLETED-D1 at location [8]. When they both exist, the counter 'VC2Waiting' will add 1.

```
VC2Complete = 0
for line in allPatientInformation:
    line = line.split() #nested list is created
    try:
        if (line[3] == 'VC2') and (line[8] == 'COMPLETED'):
            VC2Complete = VC2Complete + 1
    except:
        continue

print ('These are the statistics of both vaccinaiton centre.')
print ('-' * 50)
print ('vaccination Centre\t\t', 'VC1\t', 'VC2')
print ('-' * 50)
print ('Total Patients\t\t\t', VC1Total, '\t', VC2Total, '\t')
print ('Waiting 2nd Dose Patients\t', VC1Waiting, '\t', VC2Waiting, '\t')
print ('Completed Patients\t\t', VC1Complete, '\t', VC2Complete, '\t')
print ('-' * 50)
input('Press Enter to continue.') #This is to let the user to have time to look at the table
print ('=' * 50)
```

After all of the counting process, the program will print all the information that are stored in the variables which are 'VC1Total', 'VC2Total', 'VC1Waiting', 'VC2Waiting', 'VC1Completed', and 'VC2Completed'. All of the information will be displayed in a table form to let the user to be able to read it easily. \t and the minus sign are used to produce a neat table. The input function is also let the user to have time to read the table before the program continues to the Welcome Page Function. The user has to press enter to let the program continue.

Run Application Function

```
def run_application():  
    while True:  
        operation = welcome_page()  
  
        if (operation == 'register'):  
            print ('\nYou had chosed registration.')  
            ageNameVc = registration_age_name_vc ()  
            name = ageNameVc[1]  
            age = ageNameVc[0]  
            vaccinationCentre = ageNameVc[2]  
            if (age < 12):  
                continue  
            registration_operation(age, name, vaccinationCentre)  
  
        if (operation == 'vaccine'):  
            print ('\nYou had chosed vaccine administration.')  
            vaccination_operation()  
  
        if (operation == 'search'):  
            print ('\nYou had chosed to search for patients record.')  
            search_operation()  
  
        if (operation == 'statistics'):  
            print ('\nYou had chosed to see patient statistics.')  
            statistics_operation()  
  
        if (operation == '0'):  
            print ('\nThank you !')  
            break
```

The name of this function is `run_application ()`. This function basically combines all the functions that were mentioned above and arrange their flow. All the functions were placed in a `WHILE TRUE` loop so that the user doesn't need to run the program again when he wants to user another function after finish using the recent function. When the user wants to quit, he will have to input 0 at the Welcome Page Function to end the program. The function will run according to what is returned from the Welcome Page Function which is stored in the variable 'operation' and is checked using IF ELSE block. Before any of the function runs, the program will inform the user what function they had chosen. When the function end, they will all go back to Welcome Page Function because they are in `WHILE TRUE` loop unless they input 0. All of the function are run almost the same except for the Registration Function

because there are 2 functions in it. To eliminate the patient that is under age of 12, the function is separated into 2 and the age, name and vaccination centre is returned from the first function. The returned data are stored in a list. Hence the program will need to extract them out and store it in another variable to contain each of them. The age is checked with an IF ELSE block before going to the second function in Registration Function. If the age is lower than 12, the program will skip the next function and straight go to the next iteration to go back to Welcome Page Function since there are no vaccine for patient age under 12. If the age is higher or equal to 12, the program will continue to the second function in Registration Function.

Main Code

```
run_application()
```

This is the only main code for the program is to run the `run_application ()` function. This will prevent any global variable in the program and make information of the patients more secured.

Screenshots of Sample Input/Output and Explanation

Text Files

There will be 2 text files that generated to cooperate with this program which are patients.txt and vaccination.txt file. It is important to note that user is not allowed to edit both of the text file as it may cause the program to have bugs or crashes. Especially creating an empty line after the last line in patients.txt file will cause bugs in the program.

patients.txt

```
|Code||Name||Age||Vaccine_Centre||Vaccine_Name||Dosage_Required||Contact_No||E-mail||Status|
0001 Adam 18 VC2 EC 1Dose 0122367886 Adam@gmail.com COMPLETED
0002 Lucas 20 VC1 CZ 2Dose 0122367347 Lucas@gmail.com COMPLETED-D1
0003 Luna 60 VC2 DM 2Dose 0153658516 Luna@gmail.com COMPLETED-D1
0004 Thomas 21 VC1 AF 2Dose 0125544455 Thomas@gmail.com COMPLETED-D1
0005 Kierra 18 VC2 BV 2Dose 02135461321 Kierra@yahoo.com NEW
0006 Ali 16 VC2 DM 2Dose 0154442254 Ali@hotmail.com COMPLETED-D1
0007 Siti 17 VC1 CZ 2Dose 0115554468 Siti@yahoo.com NEW
0008 Muthu 35 VC2 CZ 2Dose 0154443355 Muthu@gmail.com COMPLETED-D1
0009 Nicholas 44 VC2 BV 2Dose 0154488794 Nicholas@hotmail.com COMPLETED
0010 EeLoong 65 VC2 DM 2Dose 0113215496 Loong@yahoo.com COMPLETED
0011 Andrew 43 VC1 CZ 2Dose 0114498946 Andrew@gmail.com NEW
0012 Chris 22 VC1 BV 2Dose 0116468765 Chris@yahoo.com NEW
0013 Anna 24 VC2 DM 2Dose 0147741132 Anna@yahoo.com COMPLETED-D1
0014 Calvin 26 VC1 EC 1Dose 0115454132 Calvin@gmail.com COMPLETED
0015 Eric 33 VC2 EC 1Dose 0132546512 Eric@hotmail.com NEW
0016 Kenneth 38 VC2 DM 2Dose 0112356487 Kenneth@hotmail.com NEW
0017 YiCong 55 VC2 BV 2Dose 02145455541 YiCong@hotmail.com COMPLETED-D1
0018 TzeHong 69 VC1 AF 2Dose 01544412111 TzeHong@yahoo.com COMPLETED
0019 SuePeow 41 VC2 CZ 2Dose 0147778854 SuePeow@hotmail.com COMPLETED-D1
0020 Batolamio 55 VC1 DM 2Dose 0196332255 Batolamio@yahoo.com NEW
0021 XinHong 22 VC2 EC 1Dose 127823983 XinHong@gmail.com NEW
0022 Anthony 70 VC2 BV 2Dose 123456789 Anthony@gmail.com NEW
0023 ZhongYi 56 VC2 EC 1Dose 0125554444 Zhongyi@gmail.com COMPLETED
0024 Raju 25 VC1 AF 2Dose 0122225554 Raju@gmail.com COMPLETED-D1
0025 Agrula 60 VC2 AF 2Dose 0215554444 Agru@gmail.com COMPLETED-D1
0026 LucasSu 15 VC1 AF 2Dose 0122225555 LucasSu@gmail.com COMPLETED
```

This is what it looks like in the file. All the arrangement of information of all patient are the same and is stated at the first line.

vaccination.txt

```
|Code| |Patient_Name| |Vaccine_Name| |Given_Date_And_Time| |Dose_Given|
0006 Ali DM |20/08/2021| |17:13:40| D1
0025 Agrula AF |19/09/2021| |19:13:42| D1
0026 LucasSu AF |20/09/2021| |22:57:00| D1
0026 LucasSu AF |20/09/2021| |23:03:00| D2
0024 Raju AF |20/09/2021| |23:05:36| D1
```

This is what it looks like in the file same as the patient.txt file, the arrangement of information is all the same and is stated at the first line

Welcome Page Function

```
Welcome, please choose your operation.  
  
Enter "register" to register new patient.  
Enter "vaccine" for vaccine administration.  
Enter "search" to search for patient record.  
Enter "statistics" to see for patients statistics.  
Enter "0" to quit.  
  
Please enter your operation here: ABC  
Please input the right operation.  
Please enter your operation here: 123  
Please input the right operation.  
Please enter your operation here: █
```

This is the thing that user will see when they run the program. As you can see when the input is not one of the choices of the functions, the program will ask the user for another input.

```
Welcome, please choose your operation.  
  
Enter "register" to register new patient.  
Enter "vaccine" for vaccine administration.  
Enter "search" to search for patient record.  
Enter "statistics" to see for patients statistics.  
Enter "0" to quit.  
  
Please enter your operation here: register  
  
You had chosed registration.  
Please choose your desired vaccination centre ["VC1" or "VC2"]: █
```

If the user inputs the correct string, the program will run the function that the user wants. Diagram above is an example of user calling Registration Function by input 'register'.

```
Welcome, please choose your operation.

Enter "register" to register new patient.
Enter "vaccine" for vaccine administration.
Enter "search" to search for patient record.
Enter "statistics" to see for patients statistics.
Enter "0" to quit.

Please enter your operation here: vaccine

You had chosed vaccine administration.
Had you done your registration? ["yes" if yes, "no" if no]:
```

This is an example of calling Vaccination Function by input 'vaccine'.

```
Welcome, please choose your operation.

Enter "register" to register new patient.
Enter "vaccine" for vaccine administration.
Enter "search" to search for patient record.
Enter "statistics" to search for patient record.
Enter "0" to quit.

Please enter your operation here: search

You had chosed to search for patient conditions.
What is your ID code:
```

This is an example of calling Search Function by input 'search'.

```
Welcome, please choose your operation.

Enter "register" to register new patient.
Enter "vaccine" for vaccine administration.
Enter "search" to search for patient record.
Enter "statistics" to search for patient record.
Enter "0" to quit.

Please enter your operation here: statistics

You had chosed to see patient statistics.
```

This is an example of calling Statistics Function by input 'statistics'.

```
Welcome, please choose your operation.  
  
Enter "register" to register new patient.  
Enter "vaccine" for vaccine administration.  
Enter "search" to search for patient record.  
Enter "statistics" to search for patient record.  
Enter "0" to quit.  
  
Please enter your operation here: 0  
  
Thank you !
```

This is when the user chooses to quit the program by input '0'.

Registration Function

```
Please enter your operation here: register

You had chosed registration.
Please choose your desired vaccination centre ["VC1" or "VC2"]: ABC
Please input the right code for vaccination centre.

Please choose your desired vaccination centre ["VC1" or "VC2"]: █
```

When the user enters Registration Function, the program will first ask the desired vaccination centre. If the input is not VC1 or VC2, the program will ask again just like what is shown in the picture above.

```
Please choose your desired vaccination centre ["VC1" or "VC2"]: VC1
Please input your name (without space, use "_" to indicate space): LucasSu
Please input your age: ABC
Please input your age in numerical form.
Please input your age: 15
Please input your email: LucasSu@gmail.com
Please input your contact number [without space]: 0122225555
```

Then the user will have to input their information according to what is asked by the program following the instructions. As you can see if when the user input 'ABC' when the program is asking age, the program will sense that it is not an integer and ask for the age again and remind the user to input in numerical form. While for contact number space is also not allowed as it will cause malfunction in the program. Same as with the input of email but it is not reminded because it is not possible to have space in an email.

```
The vaccine suitable for you are "AF" ,"CZ" and "DM".
Please input the vaccine code that you want to have: ABC
Please input the right code for the vaccine.

The vaccine suitable for you are "AF" ,"CZ" and "DM".
Please input the vaccine code that you want to have: EC
Please input the right code for the vaccine.

The vaccine suitable for you are "AF" ,"CZ" and "DM".
Please input the vaccine code that you want to have: █
```

After that the program will display the suitable vaccine for them and ask the user which vaccine they want. As you can see in the diagram is the user input any random string such as 'ABC' or other vaccine name that is not suitable for them such as EC, the program will ask for input again so that the user choose the right vaccine.

```
The vaccine suitable for you are "AF" ,"CZ" and "DM".
Please input the vaccine code that you want to have: AF
=====
Your code is 0026 Please remember your code.
Thanks for registering LucasSu . Your registered vaccine is AF
Your code is 0026
Press Enter to continue.█
```

When the user chooses the right vaccine code, the program will generate and display the patient code and the user is reminded to remember the code.

```
24    0023 ZhongYi 56 VC2 EC 1Dose 0125554444 Zhongyi@gmail.com COMPLETED
25    0024 Raju 25 VC1 AF 2Dose 0122225554 Raju@gmail.com NEW
26    0025 Agrula 60 VC2 AF 2Dose 0215554444 Agru@gmail.com COMPLETED-D1
27    0026 LucasSu 15 VC1 AF 2Dose 0122225555 LucasSu@gmail.com NEW
```

Then the information is registered and appended into patients.txt file.

```
Please enter your operation here: register

You had chosed registration.
Please choose your desired vaccination centre ["VC1" or "VC2"]: VC1
Please input your name (without space, use "_" to indicate space): LucasSu
Please input your age: 10
Sorry you are too young for vaccine.

=====
Welcome, please choose your operation.

Enter "register" to register new patient.
Enter "vaccine" for vaccine administration.
Enter "search" to search for patient record.
Enter "statistics" to search for patient record.
Enter "0" to quit.

Please enter your operation here: █
```


If the user input age lower than 12, the program will inform that he is too young for vaccine and go back to Welcome Page Function and nothing will be stored in the patients.txt file.

Vaccination Function

```
Please enter your operation here: vaccine

You had chosed vaccine administration.
Had you done your registration? ["yes" if yes, "no" if no]: no
Please choose your desired vaccination centre ["VC1" or "VC2"]: no
Please input the right code for vaccination centre.

Please choose your desired vaccination centre ["VC1" or "VC2"]:
```

When the user enters the Vaccination Function, the program will ask if the user had registered or not if the user input no the program will call Registration Functions and the process will be the same as what is explained just now.

After finishing the registration, the user will come back to this question and the user will input yes and proceed to the real vaccination process.

```
Please enter your operation here: vaccine

You had chosed vaccine administration.
Had you done your registration? ["yes" if yes, "no" if no]: yes
What is your ID code: 0026
Hello LucasSu
Your vaccine registered is AF
Your vaccine status was NEW

=====
You had completed your first vaccine, please come back in 2 weeks (14 days) to get your second vaccine.
Press Enter to continue.
```

After input yes, the program will ask for the patient code of the user. Then the program will read the information from the patients.txt and display the patient's name, registered vaccine and what is their status before the vaccination. Then the program will assume that the vaccine is given and update the information in the patients.txt file and record the time and the dose given in the vaccination.txt file. Besides that, the patient will also display instruction to the patient to come back after how many days for the next vaccine if needed according to the information provided in the question paper.

```
24 0023 ZhongYi 56 VC2 EC 1Dose 0125554444 Zhongyi@gmail.com COMPLETED
25 0024 Raju 25 VC1 AF 2Dose 0122225554 Raju@gmail.com NEW
26 0025 Agrula 60 VC2 AF 2Dose 0215554444 Agru@gmail.com COMPLETED-D1
27 0026 LucasSu 15 VC1 AF 2Dose 0122225555 LucasSu@gmail.com COMPLETED-D1|
```

As you can see the information is updated in patients.txt file.

Then after the user press enter to proceed, the record will also be recorded into the vaccination.txt file.

7	0023 ZhongYi EC 24/08/2021 14:01:08 D1
8	0025 Agrula AF 19/09/2021 19:13:42 D1
9	0026 LucasSu AF 20/09/2021 22:57:00 D1

This is what is recoded in the vaccination.txt file and the D1 indicates first dose is given while D2 is dose 2.

```
You had completed your first vaccine, please come back in 2 weeks (14 days) to get your second vaccine.
Press Enter to continue.
=====
Welcome, please choose your operation.

Enter "register" to register new patient.
Enter "vaccine" for vaccine administration.
Enter "search" to search for patient record.
Enter "statistics" to search for patient record.
Enter "0" to quit.

Please enter your operation here: █
```

After that the program will go back to the Welcome Page Function.

If the patient come again for the second dose or he had already registered, he can input yes when the program asks if he had done registration or not. Then the program will straight lead the user to the vaccination process.

```
Please enter your operation here: vaccine

You had chosed vaccine administration.
Had you done your registration? ["yes" if yes, "no" if no]: yes
What is your ID code: 0026
Hello LucasSu
Your vaccine registered is AF
Your vaccine status was COMPLETED-D1

=====
Congratulations, you had completed your vaccine, you don't have to come again.
Press Enter to continue. █
```

This is the output if the user come again for second dose. The information will also be updated in patients.txt file and the record will also be in vaccination.txt file. The program will also inform the patient that he had completed his vaccination and does not need to come again.

```
24 0023 ZhongYi 56 VC2 EC 1Dose 0125554444 Zhongyi@gmail.com COMPLETED
25 0024 Raju 25 VC1 AF 2Dose 0122225554 Raju@gmail.com NEW
26 0025 Agrula 60 VC2 AF 2Dose 0215554444 Agru@gmail.com COMPLETED-D1
27 0026 LucasSu 15 VC1 AF 2Dose 0122225555 LucasSu@gmail.com COMPLETED
```

```
8 0025 Agrula AF |19/09/2021| |19:13:42| D1
9 0026 LucasSu AF |20/09/2021| |22:57:00| D1
10 0026 LucasSu AF |20/09/2021| |23:03:00| D2
```

If the user come again after he completed his vaccination.

```
Please enter your operation here: vaccine

You had chosed vaccine administration.
Had you done your registration? ["yes" if yes, "no" if no]: yes
What is your ID code: 0026
Hello LucasSu
Your vaccine registered is AF
Your vaccine status was COMPLETED

=====
You had already completed your vaccine, why you came?

Press Enter to continue.█
```

This is the output that he will get and no record or update will be made.

```
Please enter your operation here: vaccine

You had chosed vaccine administration.
Had you done your registration? ["yes" if yes, "no" if no]: yes
What is your ID code: 0055
Your input code is not registered.
=====
Welcome, please choose your operation.

Enter "register" to register new patient.
Enter "vaccine" for vaccine administration.
Enter "search" to search for patient record.
Enter "statistics" to search for patient record.
Enter "0" to quit.

Please enter your operation here: █
```

If the user input a code that is not registered, the program will display a message to tell that the code is not registered and go back to Welcome Page Function.

Search Function

```
Please enter your operation here: search

You had chosed to search for patient conditions.
What is your ID code: █
```

When the user chooses Search Function, the program will ask for the code that the user wants to search for.

```
What is your ID code: 0026
=====
This is the information about the patient.
The informations are arranged as sequence below.

|Code| |Name| |Age| |Vaccine_Centre| |Vaccine_Name| |Dosage_Required| |Contact_No| |E-mail| |Status|
0026 LucasSu 15 VC1 AF 2Dose 0122225555 LucasSu@gmail.com COMPLETED
Press Enter to continue.█
```

Then the program will display an output of the arrangement of the information and also the information of the patient. The user will have to press enter after he finish reading the information.

```
|Code| |Name| |Age| |Vaccine_Centre| |Vaccine_Name| |Dosage_Required| |Contact_No| |E-mail| |Status|
0026 LucasSu 15 VC1 AF 2Dose 0122225555 LucasSu@gmail.com COMPLETED
Press Enter to continue.
=====
Welcome, please choose your operation.

Enter "register" to register new patient.
Enter "vaccine" for vaccine administration.
Enter "search" to search for patient record.
Enter "statistics" to search for patient record.
Enter "0" to quit.

Please enter your operation here: █
```

Then the program will go back to Welcome Page Function.

Statistics Function

```
Please enter your operation here: statistics

You had chosed to see patient statistics.
These are the statistics of both vaccinaiton centre.
-----
vaccination Centre          VC1      VC2
-----
Total Patients              10       16
Waiting 2nd Dose Patients   3        7
Completed Patients          3        4
-----
Press Enter to continue.█
```

Once the user chooses the Statistics Function, the program will immediately calculate and display the information to the user. This includes the total amount of patients in each vaccination centre and the amount of patient that had completed dose 1 and completed vaccination.

```
-----
vaccination Centre          VC1      VC2
-----
Total Patients              10       16
Waiting 2nd Dose Patients   3        7
Completed Patients          3        4
-----
Press Enter to continue.
=====
Welcome, please choose your operation.

Enter "register" to register new patient.
Enter "vaccine" for vaccine administration.
Enter "search" to search for patient record.
Enter "statistics" to search for patient record.
Enter "0" to quit.

Please enter your operation here: █
```

After the user finish reading and press enter, the program will go back to Welcome Page Function.

```
Please enter your operation here: search

You had chosed to search for patient conditions.
What is your ID code: 0055
Your input code is not registered.
=====
Welcome, please choose your operation.

Enter "register" to register new patient.
Enter "vaccine" for vaccine administration.
Enter "search" to search for patient record.
Enter "statistics" to search for patient record.
Enter "0" to quit.

Please enter your operation here: █
```

When the user input a code that is registered, he will be informed that the code is not registered and the program will go back to Welcome Page Function.

Quit Program

```
Welcome, please choose your operation.  
  
Enter "register" to register new patient.  
Enter "vaccine" for vaccine administration.  
Enter "search" to search for patient record.  
Enter "statistics" to search for patient record.  
Enter "0" to quit.  
  
Please enter your operation here: 0  
  
Thank you !  
PS C:\Users\User\Desktop\APU Study\Degree Y1S1\Python Programming\Assignment>
```

To quit the program, the user will have to input 0 and the program will display thank you and the program will be ended.

Conclusion

In conclusion this report included all of the information about the program created to manage patients' vaccination record or progress. The program created successfully fulfill all the requirements to record all the vaccination information. This program even exceeds what is required in the question paper. By default, the patients were required to input what dose of vaccine that they are going to receive by their own. However, this program will record what is their status and generate what is the dose that should be given to the patient. This will effectively reduce the chances of program crashing when the user input something that is not related or something random. Besides that, patients also don't have to remember what dose had they get and how many dose are left. What they need to remember only is their code and that will let them be able to access all the functions in the program. Furthermore, many codes were also implemented in this program to prevent the random inputs from the user to crash the program. Instead, the program will continuously ask the user until a desired input is obtained. This will effectively reduce the chance of program crashing.

However, there are still many improvements that can be done in this program to increase its efficiency and reduce the code length. Moreover, some of the inputs are also not checked and this may cause some of the user that does not following the instruction and give random or not desired input such as the input for name, email and contact number. Further improvement can be done on that to make sure that what is input by the user is what is desired by the program. Besides, the nested list can be further applied in this program other than only in the statistics function to reduce the steps of extracting the particular patients out from the list that contain all patients' information. The variable that is used to contain the information that is extracted can be removed. This will reduce the variables that are used in the program and alightly reduce its complexity.

The effort of lecturer Ts. Sivaguru Subarmaniyan in providing such quality classes and deliver such important knowledge on coding is also much appreciated. Without the knowledge delivered this program would not be created. Besides, he also emphasizes on doing our own research on some extra knowledge on creating pseudocode, flowchart and other inbuilt functions in Python that will help in completing and improving the program as this will increase our understandings on coding overall.

In a nutshell, during the process of creating this program, many things were learned as precious knowledges. First and foremost, the importance of having a plan or an algorithm before start coding is understood. If you fail to plan, you are planning to fail. The process of coding will be much faster and

easier when there is a plan. Pseudocode and flowchart to follow. Besides, by having a pseudocode and flowchart, the program can be transferred by someone else in other language such as C++ and other more. Pseudocode and flowchart are basically the universal language that everyone will know once looking at it. Moreover, I was also trained to think logically and critically to produce an algorithm and create the suitable flow for the program. Furthermore, some ideas were also developed to help in add some extra features or improve the program and increase its efficiency. Secondly, extensive research is also done to discover some inbuilt function in Python that is helpful in completing the program. Those inbuilt functions were fully utilized in this program. This assignment enabled me to have a deeper understanding on Python.

References

- GeeksforGeeks. (2021). *How to write a Pseudo Code?* Retrieved from: <https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/> [Accessed Date: 17 August 2020]
- Anon. (n.d.). *Pseudocode Quickref.* Retrieved from: https://tools.withcode.uk/quickref/string_manipulation [Accessed Date: 18 August 2020]
- Anc 9608. (2017). *Text File Processing Pseudocode.* [Video]. YouTube. Retrieved from: <https://www.youtube.com/watch?v=snFPNd13XyA> [Accessed Date: 19 August 2020]
- Programiz. (n.d.). *Flowchart In Programming.* Retrieved from: <https://www.programiz.com/article/flowchart-programming> [Accessed Date: 20 August 2020]
- W3schools. (n.d.). *Python String split() Method.* Retrieved from: https://www.w3schools.com/python/ref_string_split.asp [Accessed Date: 22 August 2020]
- GeeksforGeeks. (2021). *Python String join() Method.* Retrieved from: <https://www.geeksforgeeks.org/python-string-join-method/> [Accessed Date: 24 August 2020]
- Programiz. (n.d.). *How to get current date and time in Python?* Retrieved from: <https://www.programiz.com/python-programming/datetime/current-datetime> [Accessed Date: 26 August 2020]