ASIA PACIFIC UNIVERSITY TECHNOLOGY & INNOVATION

CT108-3-1-PYP PYTHON PROGRAMMING

TITLE	FINAL ASSIGNMENT
NAME & STUDENT ID	MOHAMMAD FAWZAN ALIM TP064501
INTAKE	APD1F2106CE
LECTURER	TS. SIVAGURU SUBARMANIYAN
DATE	27 SEPTEMBER 2021

TABLE OF CONTENTS

INTRODUCTION

Coronavirus disease (COVID-19) is an infectious disease caused by Coronavirus. The virus can spread through small liquid particles coming from an infected person's mouth or nose. Due to its infectious nature, there is a worldwide pandemic since the beginning of the last year. As of now, there is no specific drug or medicine to cure this disease but scientists are working hard to develop one. Recently researchers from different scientific society have made a breakthrough by inventing multiple types of vaccines. The objective of this assignment is to implement a COVID-19 vaccination record management system using the programming knowledge we have gained in our Python Programming module.

ASSUMPTION

For this assignment, we have assumed that we have two vaccination centers; VC1 and VC2. A patient must complete all the dosage of the vaccination from the same center they have registered. For example: If a patient registers at VC1, he/she cannot go to VC2 to get vaccinated. We have also assumed that there are 5 vaccines available. All the vaccines are available at both the centers with unlimited stock. The record management system will be a python program which will be run by trained operators at the vaccination centers.

DESIGN

Pseudocode

```
PROGRAM VaccineManagementSystem
BEGIN
      Print "Welcome to COVID-19 Vaccination Record Management System"
      Print newline
      call menu()
END
```

```
FUNCTION menu()
      DOWHILE(1==1)
            Print "HOME MENU:"
            Print newline

            Print"  1. New Patient Registration"
            Print"  2. Vaccine Administration"
            Print"  3. Search Patient Record and Vaccination Status"
            Print"  4. Statistical Information on Patients Vaccinated"
            Print"  0. Exit"

            DOWHILE(1==1)
                  Print newline
                  Print "  Choose an option: "

                  Read homeChoice

                  IF homeChoice == '0'
                        BREAK OUT OF LOOP
                  ENDIF

                  IF homeChoice == '1'
                        BREAK OUT OF LOOP
                  ENDIF

                  IF homeChoice == '2'
                        BREAK OUT OF LOOP
                  ENDIF

                  IF homeChoice == '3'
                        BREAK OUT OF LOOP
                  ENDIF
```

```
                IF homeChoice == '4'
                      BREAK OUT OF LOOP
                ENDIF

                Print "  Invalid input. Input must be a number between
                0 and 4. Try again."
          ENDDO

          IF homeChoice == '0'
                Print newline
                Print "  Exiting the program...."

                RETURN
          ENDIF

          IF homeChoice == '1'
                call registration()
                GO TO NEXT ITERATION
          ENDIF

          IF homeChoice == '2'
                call administration()
                GO TO NEXT ITERATION
          ENDIF

          IF homeChoice == '3'
                call search()
                GO TO NEXT ITERATION
          ENDIF

          IF homeChoice == '4'
                call statistics()
                GO TO NEXT ITERATION
          ENDIF
     ENDDO
ENDFUNCTION
```

```
FUNCTION registration()
      Print newline
      Print "NEW PATIENT REGISTRATION MENU:"

      patient = call regInput()

      IF patient == "None"
            RETURN
      ENDIF

      patient[0] = call generateId()

      call writePatientData(patient)
ENDFUNCTION
```

```
FUNCTION regInput()
      DOWHILE(1==1)
            Print "  Vaccination center [1/2]: "
            Read center

            IF center == '1' OR center == '2'
                  BREAK OUT OF LOOP
            ENDIF

            Print "  Invalid input. Input must be '1' or '2'. Try again."
            Print newline
      ENDDO

      DOWHILE(1==1)
            Print newline
            Print "  Name:"

            IF name != ""
                  BREAK OUT OF LOOP
            ENDIF

            Print "  Please provide a name"
      ENDDO

      DOWHILE(1==1)
            TRY
                  Print newline
                  Print "  Age (in years): "
                  Read age
                  Try to convert age to float
```

```
        EXCEPT
            Print "  Invalid input. Input must be a number. Try
            again."
            NEXT ITERATION
        ENDTRY

        IF age <= 0
            Print "  Invalid input. Try again."
        ELSE
            BREAK OUT OF LOOP
        ENDIF
    ENDDO

    Print newline
    Print "  Available vaccines: "
    IF age <12
        Print "    Sorry. No Vaccines available for this age."
        Print newline
        RETURN "None"
    ENDIF

    IF age >= 12
        Print "    AF [2 Dosage with 2 weeks interval]"
    ENDIF

    IF age >= 18
        Print "    BV [2 Dosage with 3 weeks interval]"
    ENDIF

    IF age >= 12 AND age <= 45
        Print "    CZ [2 Dosage with 3 weeks interval]"
    ENDIF

    IF age >= 12
        Print "    DM [2 Dosage with 4 weeks interval]"
    ENDIF

    IF age >= 18
        Print "    EC [1 Dosage]"
    ENDIF
    DOWHILE(1==1)
        Print newline
        Print "  Vaccine code: "
        Read code
```

```
            IF code == "AF" AND age >= 12
                  BREAK OUT OF LOOP
            ENDIF

            IF code == "BV" AND age >= 18
                  BREAK OUT OF LOOP
            ENDIF

            IF code == "CZ" AND age >= 12 AND age <= 45
                  BREAK OUT OF LOOP
            ENDIF

            IF code == "DM" AND age >= 12
                  BREAK OUT OF LOOP
            ENDIF

            IF code == "EC" AND age >= 18
                  BREAK OUT OF LOOP
            ENDIF

            Print "  Invalid input. Try again."
      ENDDO

      DOWHILE (1==1)
            Print newline
            Print "  Contact Number: "
            Read contactNumber

            IF contactNumber != ""
                  BREAK OUT OF LOOP
            ENDIF

            Print "  Please provide a contact number"
      ENDDO
      DOWHILE (1==1)
            Print newline
            Print "  Email: "
            Read email

            IF email != ""
                  BREAK OUT OF LOOP
            ENDIF
      ENDDO
      RETURN ['ID', center, age, code, 'NEW', name, conotactNumber,
      email]
ENDFUNCTION
```

```
FUNCTION generateId()
      TRY
            filePatient = OPEN "patients.txt" for reading

            Read the last line from filePatient
            lastId = Read the first word of line

            CLOSE filePatient

      EXCEPT
            lastId = 0


      Id = lastId + 1
      Id = convert Id to string
      MAKE Id a six digit by filling with zeros

      RETURN Id
ENDFUNCTION
```

```
FUNCTION writePatientData(patient)
      filePatient = OPEN "patients.txt" for appending

      IF patient[0] == "000001"
            Write on filePatient "ID"
            Write on filePatient tab
            Write on filePatient "Center"
            Write on filePatient tab
            Write on filePatient "Age"
            Write on filePatient tab
            Write on filePatient "Vaccine"
            Write on filePatient tab
            Write on filePatient "Status"
            Write on filePatient tab
            Write on filePatient "Name"
            Write on filePatient tab
            Write on filePatient "Contact Number"
            Write on filePatient tab
            Write on filePatient "Email"
            Write on filePatient newline
      ENDIF

      LOOP info IN patient
            convert info to string
            Write on filePatient info
            Write on filePatient tab
            NEXT info
      ENDLOOP

      Write on filePatient newline
      CLOSE filePatient

      Print newline
      Print "  New patient registered successfully"
      Print newline
      Print "  ID:", patient[0]
      Print "  Name:", patient[5]
      Print "  Age:", str(patient[2]), "Y"
      Print "  Center: VC" + patient[1]
      Print "  Vaccine:", patient[3]
      Print "  Status:", patient[4]
      Print "  Contact Number:", patient[6]
      Print "  Email:", patient[7]
      Print newline

ENDFUNCTION
```

```
FUNCTION administration()
      Print newline
      Print "VACCINE ADMINISTRATION MENU:"
      Print newline

      patients = call readAllPatientData()

      IF patients = "None"
            RETURN
      ENDIF

      Id = call getPatientId(patients)

      call printPatientInfo(patients, Id)

      CONVERT Id to integer
      IF patients[Id][4] == 'COMPLETED'
            Print newline
            Print "  Vaccination Completed already"
            Print newline
            RETURN
      ENDIF

      patients[Id] = call updatePatientStatus(patients[Id])

      call writeAllPatientData(patients)

      call updateVaccineData(patients[Id])

ENDFUNCTION
```

```
FUNCTION getPatientId(patients)
      DOWHILE(1==1)
            Print "  Enter patient ID: "
            IF length of Id == 6
                  TRY
                        IF (Id > 0 AND Id < length of patients)
                              BREAK OUT OF LOOP
                        ELSE
                              Print "  Invalid ID. ID does not exist. Try
                              again."
                              Print newline
                  EXCEPT
                        Print "  Invalid ID. ID should be a six digit
                        number. Try again."
                        Print newline
                  ENDTRY
            ELSE
                  Print "  Invalid ID. ID should be a six digit number.
                  Try again."
                  Print newline
            ENDIF

      ENDDO

      RETURN Id
ENDFUNCTION
```

```
FUNCTION readAllPatientData()
      TRY
            filePatient = OPEN "patients.txt" for reading
      EXCEPT
            Print "  Zero patient registered so far."
            RETURN "None"

      patients = []

      LOOP line IN filePatient
            patient = []

            make line into list of words

            LOOP info IN line
                  APPEND info into patient list
            NEXT LOOP
            ENDLOOP

            APPEND patient list into patients list

      NEXT LOOP
      ENDLOOP

      CLOSE filePatient

      RETURN patients
ENDFUNCTION
```

```
FUNCTION printPatientInfo(patients, Id)
      Convert Id to integer

      Print newline

      Print "  Patient Information:"
      Print "    ID: " + patient[0]
      Print "    Name: " + patient[5]
      Print "    Age: " + patient[2] + " Y"
      Print "    Vaccine: " + patient[3]
      Print "    Current Status: " + patient[4]

ENDFUNCTION
```

```
FUNCTION updatePatientStatus(patient)
     IF patient[4] == 'COMPLETED-D1'
          patient[4] = 'COMPLETED'
          Print newline
          Print "  Status Updated to 'COMPLETED'"
          Print newline
          RETURN patient
     ENDIF

     IF patient[4] == 'NEW' AND patient[3] == 'AF'
          patient[4] = 'COMPLETED-D1'
          Print newline
          Print "  Status Updated to 'COMPLETED-D1'"
          Print "  Please come back after 2 weeks for second dose"
          Print newline
          RETURN patient
     ENDIF

     IF patient[4] == 'NEW' AND patient[3] == 'BV'
          patient[4] = 'COMPLETED-D1'
          Print newline
          Print "  Status Updated to 'COMPLETED-D1'"
          Print "  Please come back after 3 weeks for second dose"
          Print newline
          RETURN patient
     ENDIF

     IF patient[4] == 'NEW' AND patient[3] == 'CZ'
          patient[4] = 'COMPLETED-D1'
          Print newline
          Print "  Status Updated to 'COMPLETED-D1'"
          Print "  Please come back after 3 weeks for second dose"
          Print newline
          RETURN patient
     ENDIF

     IF patient[4] == 'NEW' AND patient[3] == 'DM'
          patient[4] = 'COMPLETED-D1'
          Print newline
          Print "  Status Updated to 'COMPLETED-D1'"
          Print "  Please come back after 4 weeks for second dose"
          Print newline
          RETURN patient
     ENDIF
```

```
      IF patient[4] == 'NEW' AND patient[3] == 'EC'
            patient[4] = 'COMPLETED'
            Print newline
            Print "  Status Updated to 'COMPLETED'"
            Print newline
            RETURN patient
      ENDIF
ENDFUNCTION
```

```
FUNCTION writeAllPatientData(patients)
      filePatient = OPEN "patients.txt" for writing

      LOOP patient IN patients
            LOOP info IN patient
                  CONVERT info to string
                  Write on filePatient info
                  Write on filePatient tab
            NEXT LOOP
            ENDLOOP

            Write on filePatient newline

      NEXT LOOP
      ENDLOOP

      CLOSE filePatient
ENDFUNCTION
```

```
FUNCTION updateVaccineData(patient)
      fileVaccine = OPEN "vaccination.txt" for appending

      Write on fileVaccine patient[1]
      Write on fileVaccine tab
      Write on fileVaccine patient[3]
      Write on fileVaccine tab
      Write on fileVaccine patient[0]
      Write on fileVaccine tab
      Write on fileVaccine patient[4]
      Write on fileVaccine newline

      CLOSE fileVaccine


ENDFUNCTION
```

```
FUNCTION search()
      Print newline
      Print "SEARCH MENU:"
      Print newline

      TRY
            filePatient = OPEN "patients.txt" for reading
            Go to NEXT LINE in filePatient
      EXCEPT
            Print "  Zero patient registered so far"
            Print newline
            RETURN
      ENDTRY

      Print "  Enter Search Keyword: "
      Read searchKey
      Print newline
      Print "  ID" + tab
      Print "Center" + tab
      Print "Age" + tab
      Print "Vaccine" + tab
      Print "Status" + tab
      Print "Name" + tab
      Print "Contact Number" + tab
      Print "Email"

      matchFound = 0

      LOOP line IN filePatient
            CONVERT searchKey to lowercase
            CONVERT line to lowercase
            IF searchKey IN line
                  Print "  "
                  Print line
                  matchFound = matchFound + 1
            ENDIF
      NEXT LOOP
      ENDLOOP

      Print newline
      Print "  Total Match Found = ", matchFound
      Print newline
      CLOSE filePatient
ENDFUNCTION
```

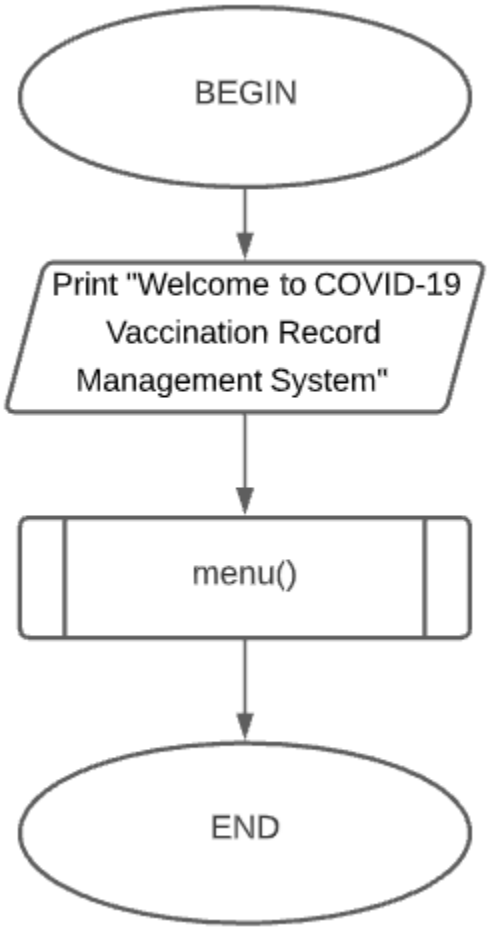
```
FUNCTION statistics()
      Print newline
      Print "STATISTICAL INFORMATION:"
      Print newline

      vaccinations = call readAllVaccinationData()

      IF vaccinations == "None"
            RETURN
      ENDIF

      call PrintStat(vaccinations, 'VC1')
      call PrintStat(vaccinations, 'VC2')
      call PrintStat(vaccinations, 'Total')

ENDFUNCTION
```

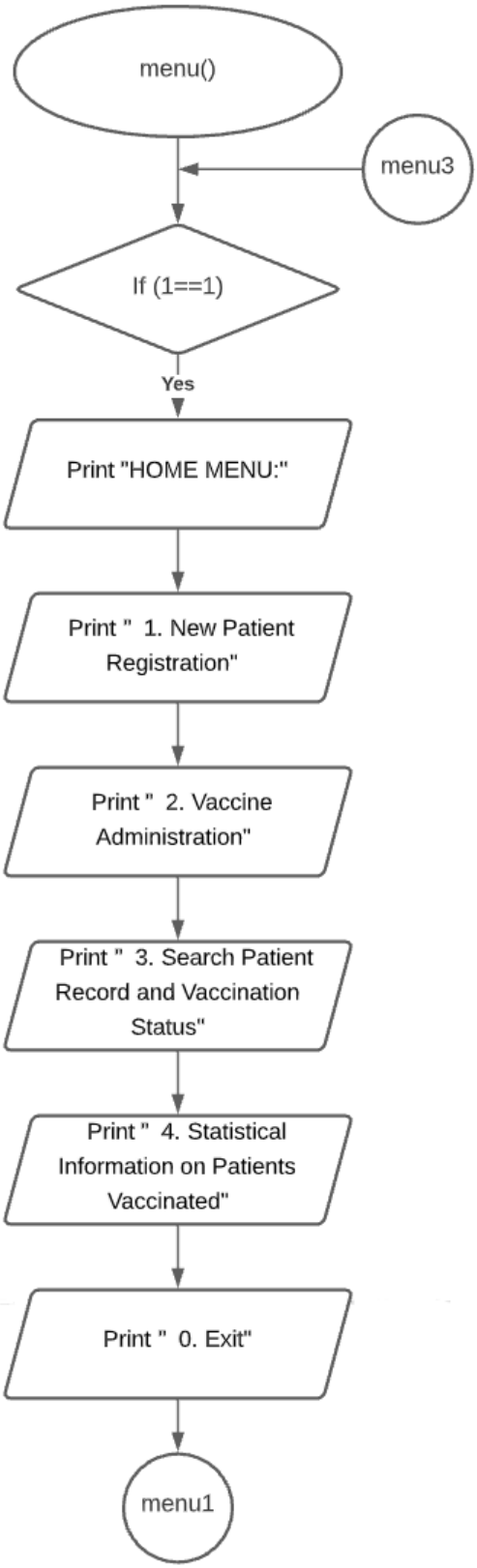
```
FUNCTION readAllVaccinationData()
      vaccinations = []

      TRY
            fileVaccine = OPEN "vaccination.txt" for reading
      EXCEPT
            Print "  Zero patient vaccinated so far"
            RETURN "None"
      ENDTRY

      LOOP line IN fileVaccine
            vaccination = []

            split line into list using tab
            LOOP info IN line
                  APPEND info in vaccination list
            NEXT LOOP
            ENDLOOP

            APPEND vaccination in vaccinations list
      NEXT LOOP
      ENDLOOP

      CLOSE fileVaccine
ENDFUNCTION
```

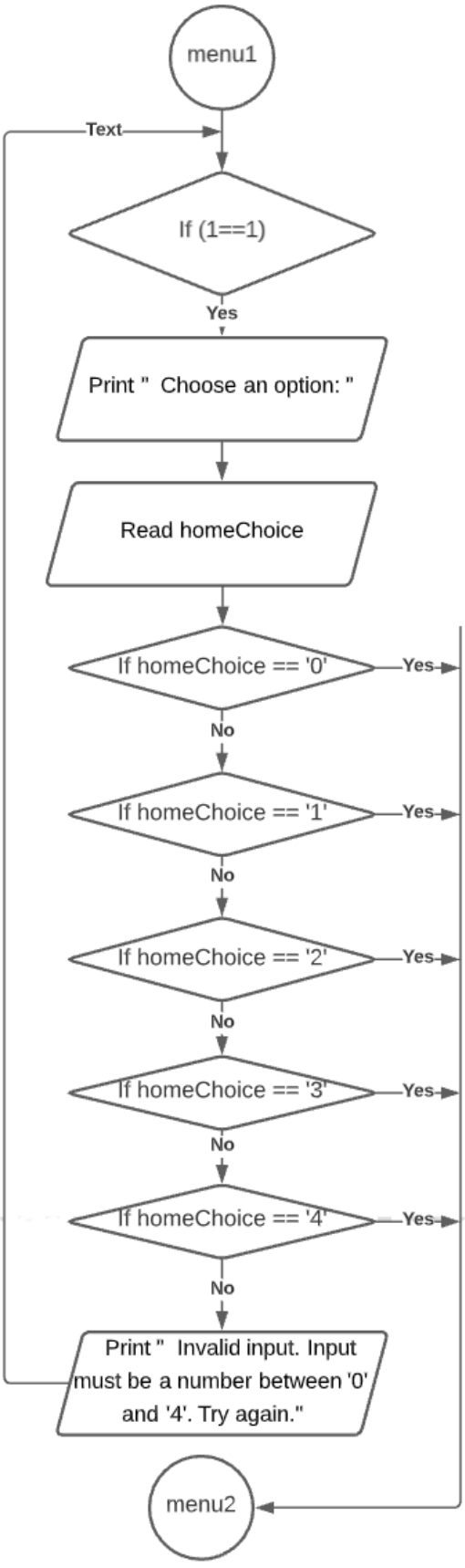
```
FUNCTION printStat(vaccinations, center)
     data = []

     APPEND ["\t", "AF", "BV", "CZ", "DM", "EC"] in data list
     APPEND ["COMPLETED-D1", 0, 0, 0, 0, 0] in data list
     APPEND ["COMPLETED", 0, 0, 0, 0, 0] in data list
     APPEND [center + "\t", 0, 0, 0, 0, 0] in data list

     LOOP vaccination IN vaccinations
          IF center[2] == vaccination[0] OR center == 'TOTAL'
               IF vaccination[1] == 'AF'
                    IF vaccination[3] == 'COMPLETED-D1'
                         data[1][1] = data[1][1] + 1
                    ELSE
                         data[2][1] = data[2][1] + 1
                         data[1][1] = data[1][1] - 1
                    ENDIF
               ENDIF

               IF vaccination[1] == 'BV'
                    IF vaccination[3] == 'COMPLETED-D1'
                         data[1][2] = data[1][2] + 1
                    ELSE
                         data[2][2] = data[2][2] + 1
                         data[1][2] = data[1][2] - 1
                    ENDIF
               ENDIF

               IF vaccination[1] == 'CZ'
                    IF vaccination[3] == 'COMPLETED-D1'
                         data[1][3] = data[1][3] + 1
                    ELSE
                         data[2][3] = data[2][3] + 1
                         data[1][3] = data[1][3] - 1
                    ENDIF
               ENDIF

               IF vaccination[1] == 'DM'
                    IF vaccination[3] == 'COMPLETED-D1'
                         data[1][4] = data[1][4] + 1
                    ELSE
                         data[2][4] = data[2][4] + 1
                         data[1][4] = data[1][4] - 1
                    ENDIF
               ENDIF
```

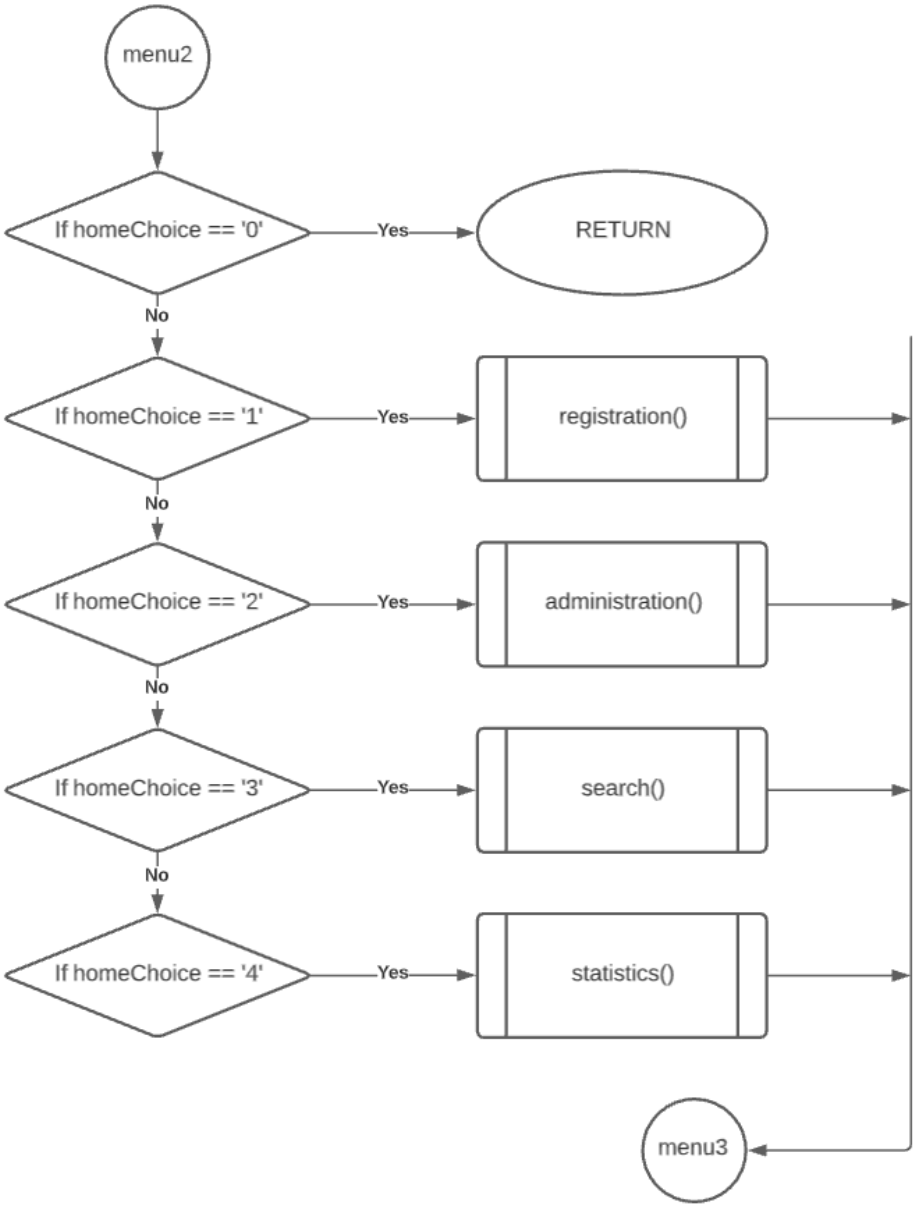
```
                IF vaccination[1] == 'EC'
                     data[2][5] = data[2][5] + 1
                ENDIF
           ENDIF

     NEXT LOOP
     END LOOP

ENDFUNCTION
```

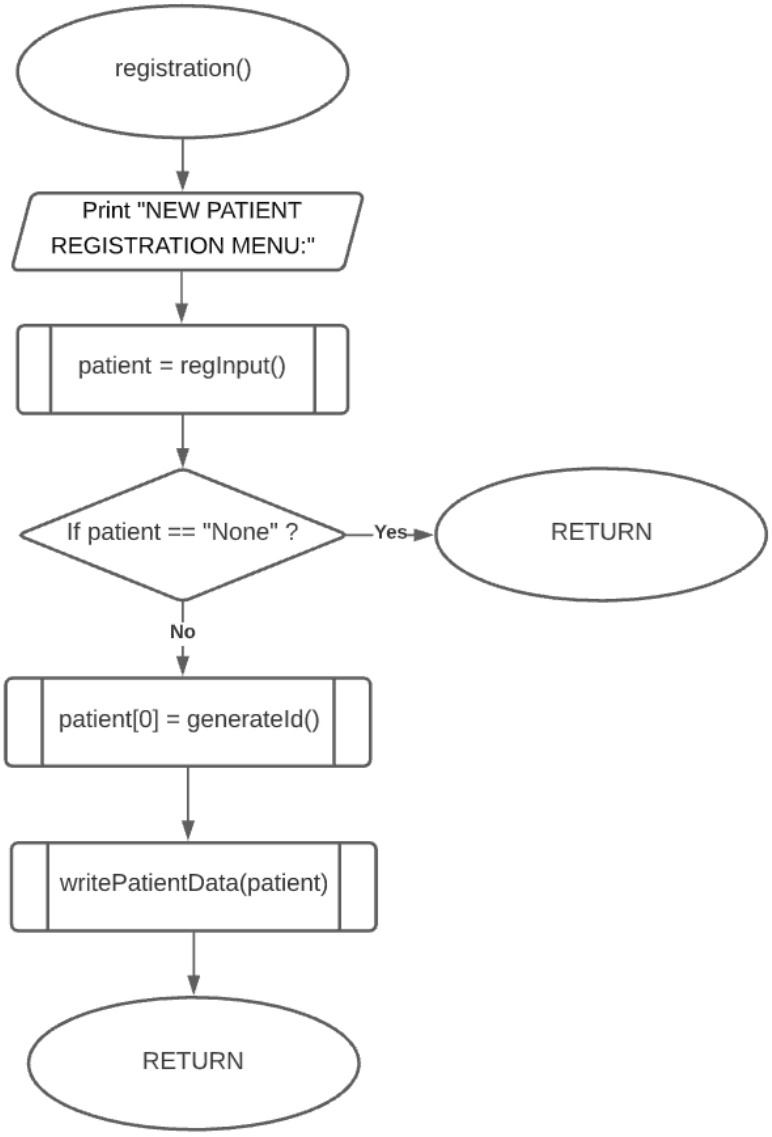
Flowchart

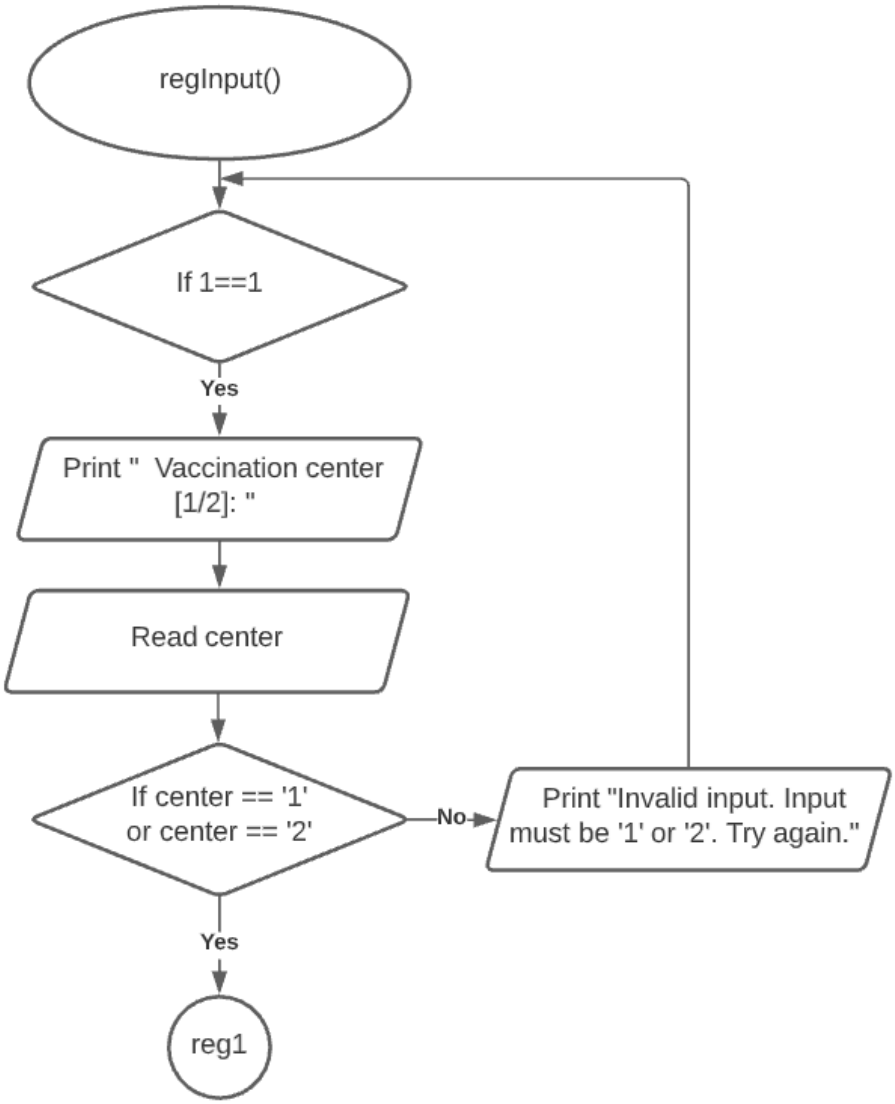

BEGIN
Print "Welcome to COVID-19 Vaccination Record Management System"
menu()
END

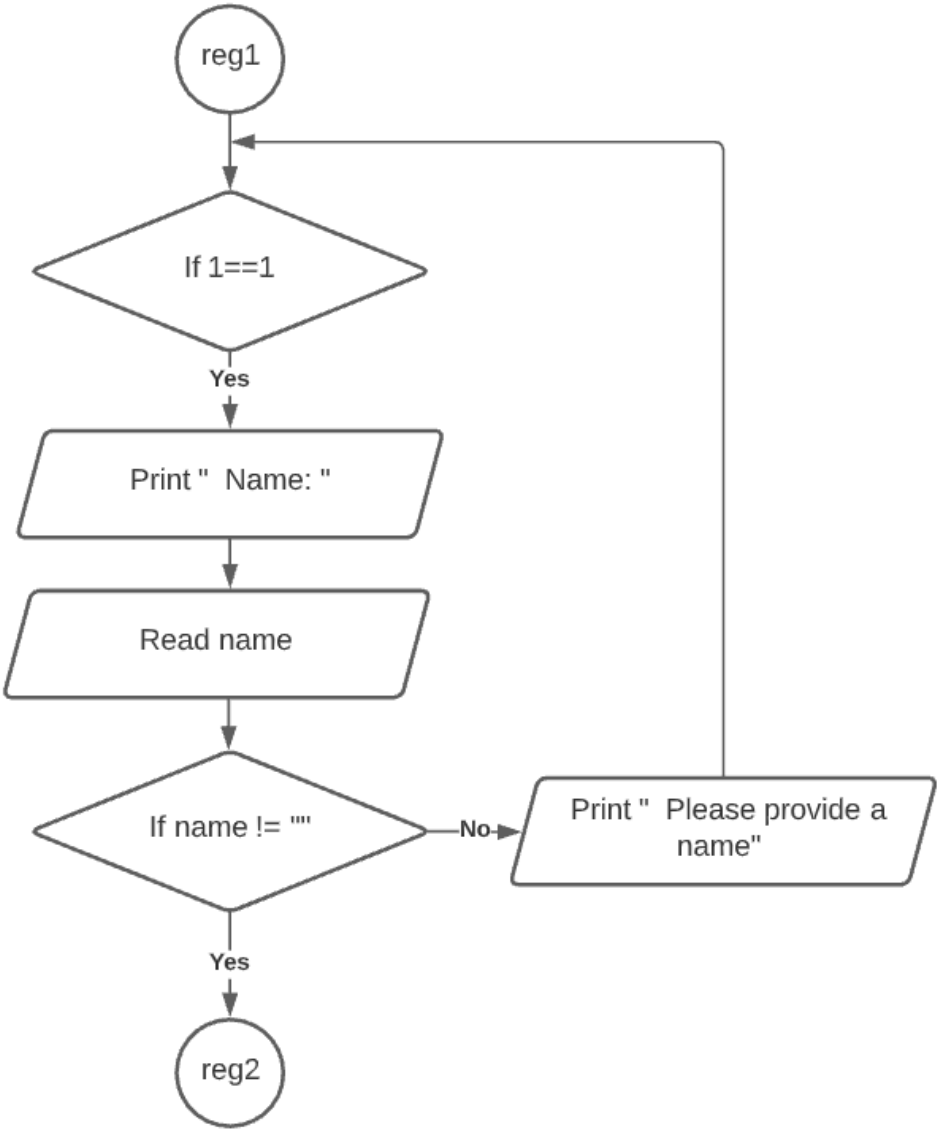

menu()
menu3
If (1==1)
Yes
Print "HOME MENU:"
Print " 1. New Patient Registration"
Print " 2. Vaccine Administration"
Print " 3. Search Patient Record and Vaccination Status"
Print " 4. Statistical Information on Patients Vaccinated"
Print " 0. Exit"
menu1

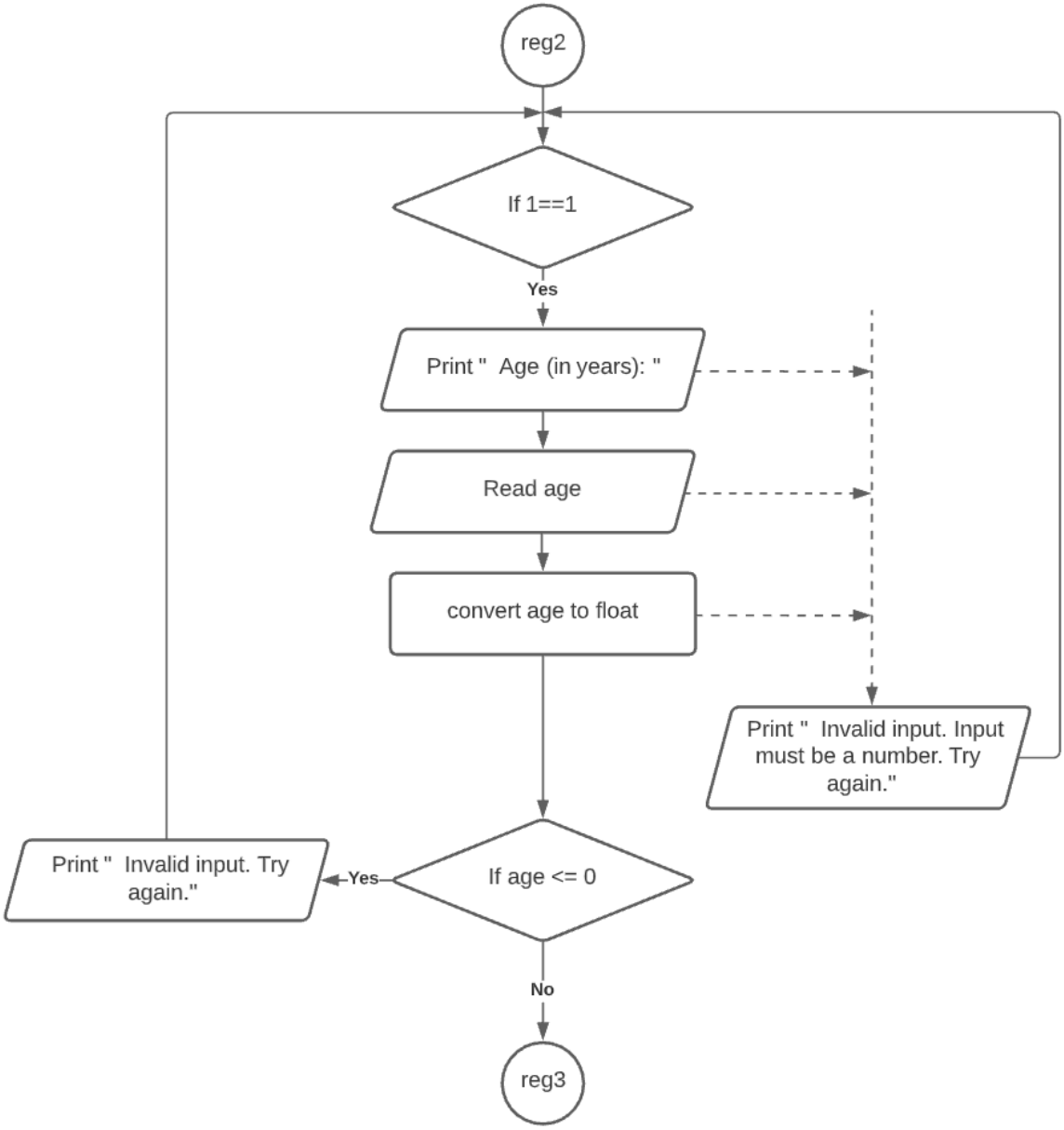

menu1
Text
If (1==1)
Yes
Print " Choose an option: "
Read homeChoice
If homeChoice == '0'
Yes
No
If homeChoice == '1'
Yes
No
If homeChoice == '2'
Yes
No
If homeChoice == '3'
Yes
No
If homeChoice == '4'
Yes
No
Print " Invalid input. Input must be a number between '0' and '4'. Try again."
menu2

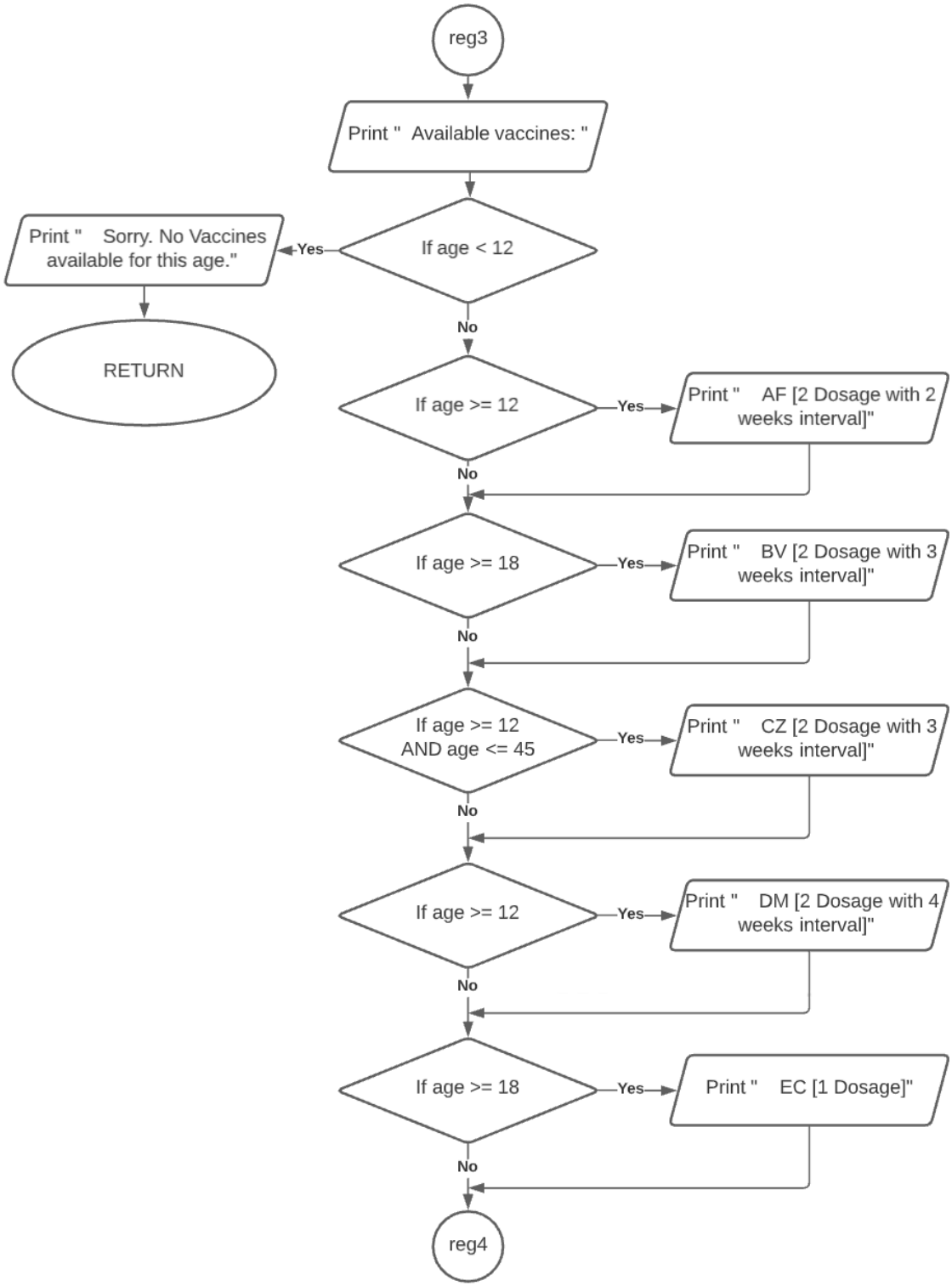

menu2
If homeChoice == '0'
Yes
RETURN
No
If homeChoice == '1'
Yes
registration()
No
If homeChoice == '2'
Yes
administration()
No
If homeChoice == '3'
Yes
search()
No
If homeChoice == '4'
Yes
statistics()
menu3

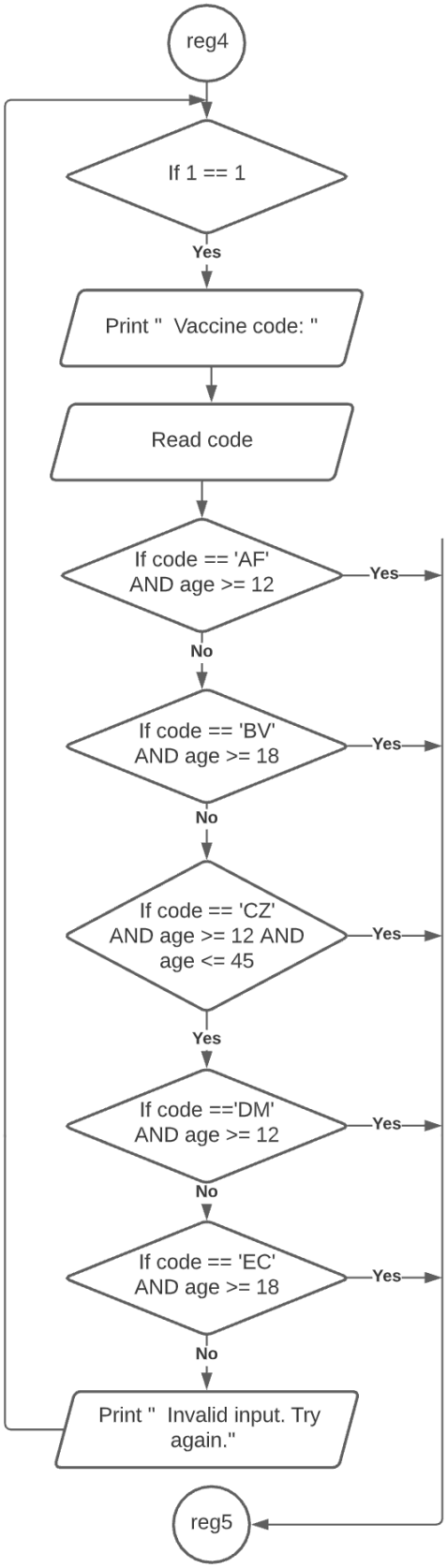

registration()
Print "NEW PATIENT REGISTRATION MENU:"
patient = regInput()
If patient == "None" ?
Yes
RETURN
No
patient[0] = generateId()
writePatientData(patient)
RETURN

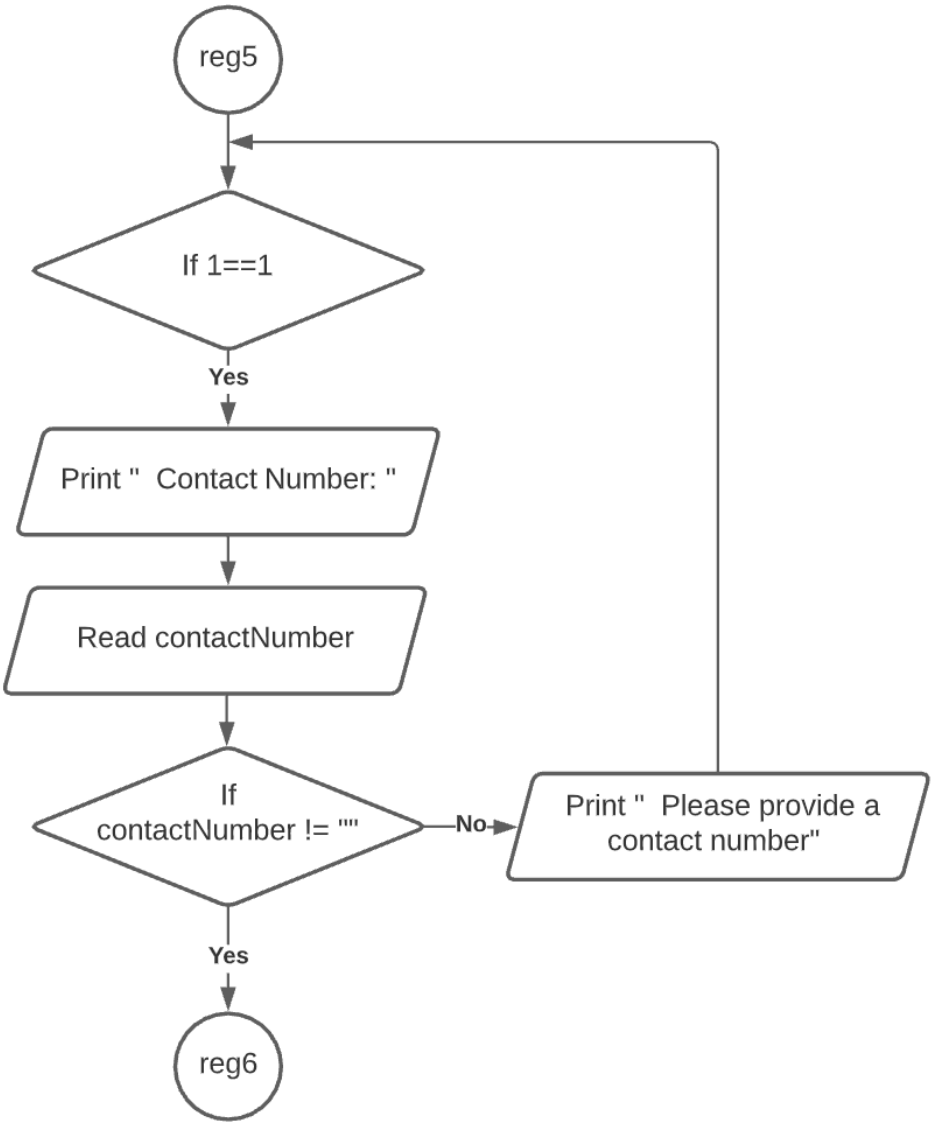

regInput()
If 1==1
Yes
Print " Vaccination center [1/2]: "
Read center
If center == '1' or center == '2'
No
Print "Invalid input. Input must be '1' or '2'. Try again."
Yes
reg1

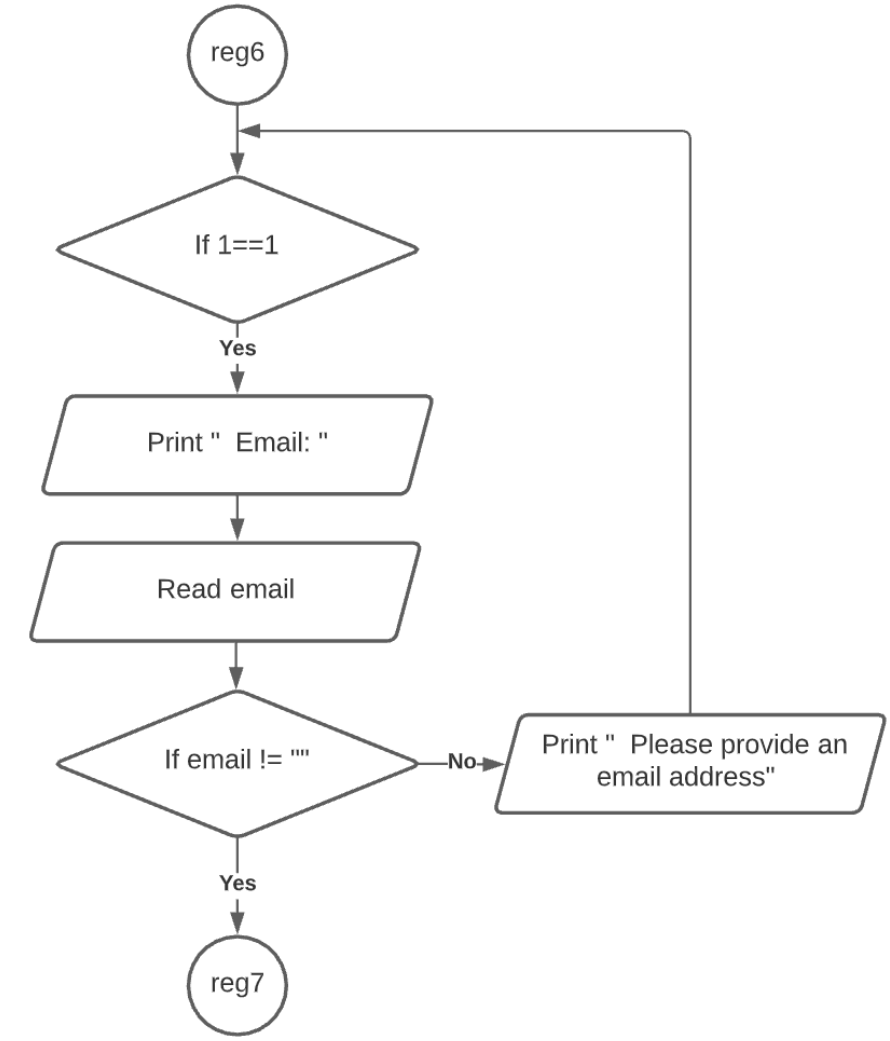

reg1
If 1==1
Yes
Print " Name: "
Read name
If name != ""
No
Print " Please provide a name"
Yes
reg2

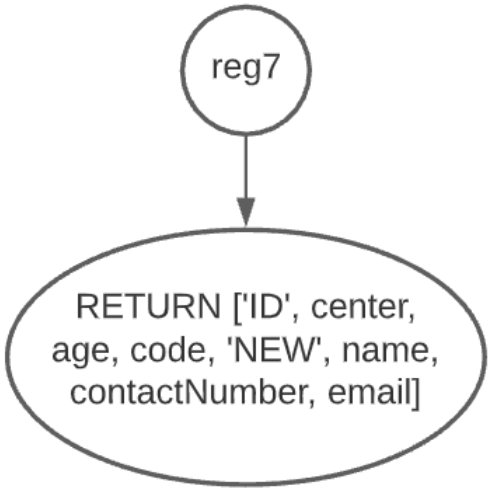

reg2
If 1==1
Yes
Print " Age (in years): "
Read age
convert age to float
Print " Invalid input. Input must be a number. Try again."
If age <= 0
Yes
Print " Invalid input. Try again."
No
reg3

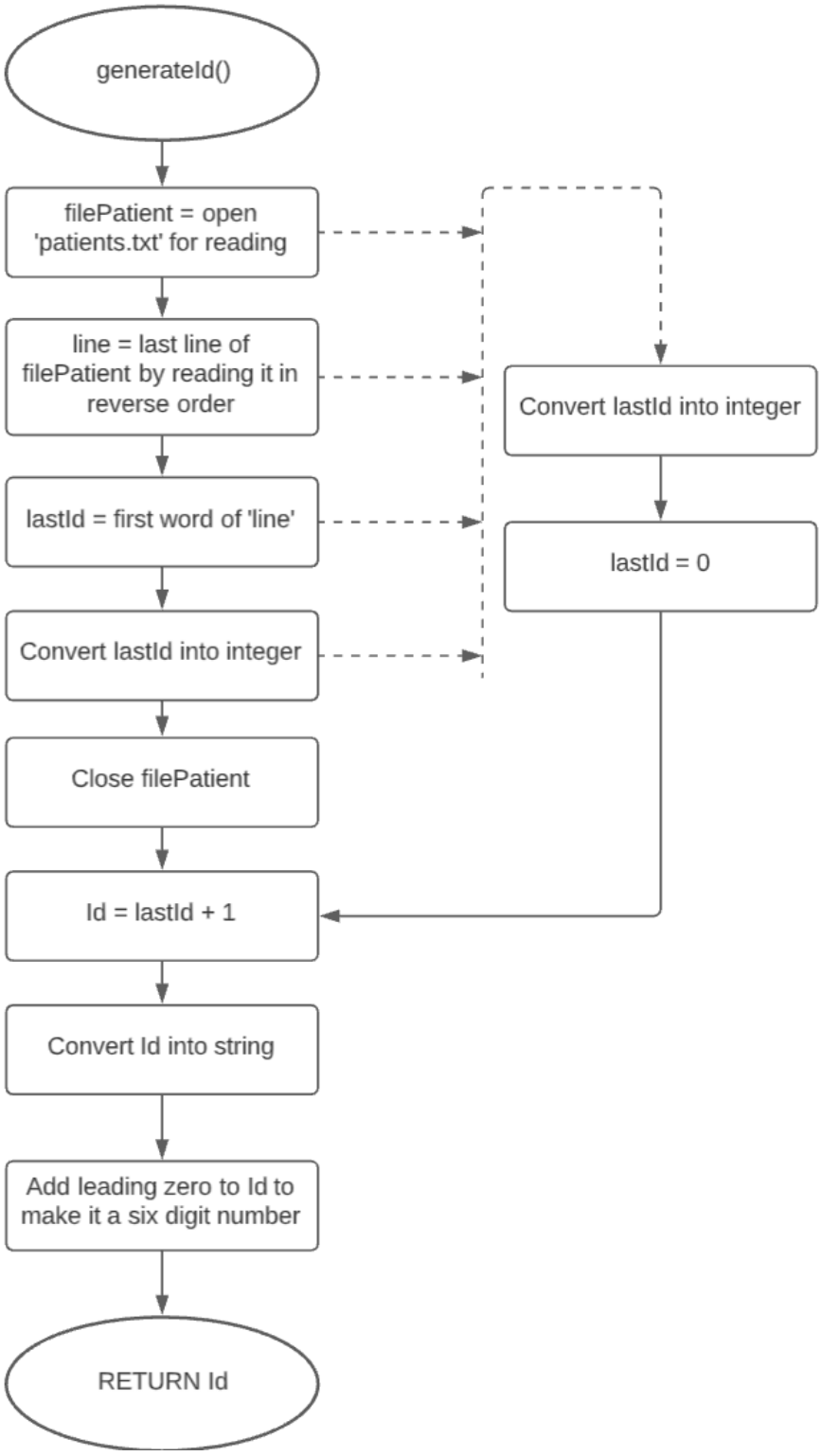

reg3
Print " Available vaccines: "
If age < 12
Yes
Print " Sorry. No Vaccines available for this age."
RETURN
No
If age >= 12
Yes
Print " AF [2 Dosage with 2 weeks interval]"
No
If age >= 18
Yes
Print " BV [2 Dosage with 3 weeks interval]"
No
If age >= 12 AND age <= 45
Yes
Print " CZ [2 Dosage with 3 weeks interval]"
No
If age >= 12
Yes
Print " DM [2 Dosage with 4 weeks interval]"
No
If age >= 18
Yes
Print " EC [1 Dosage]"
No
reg4

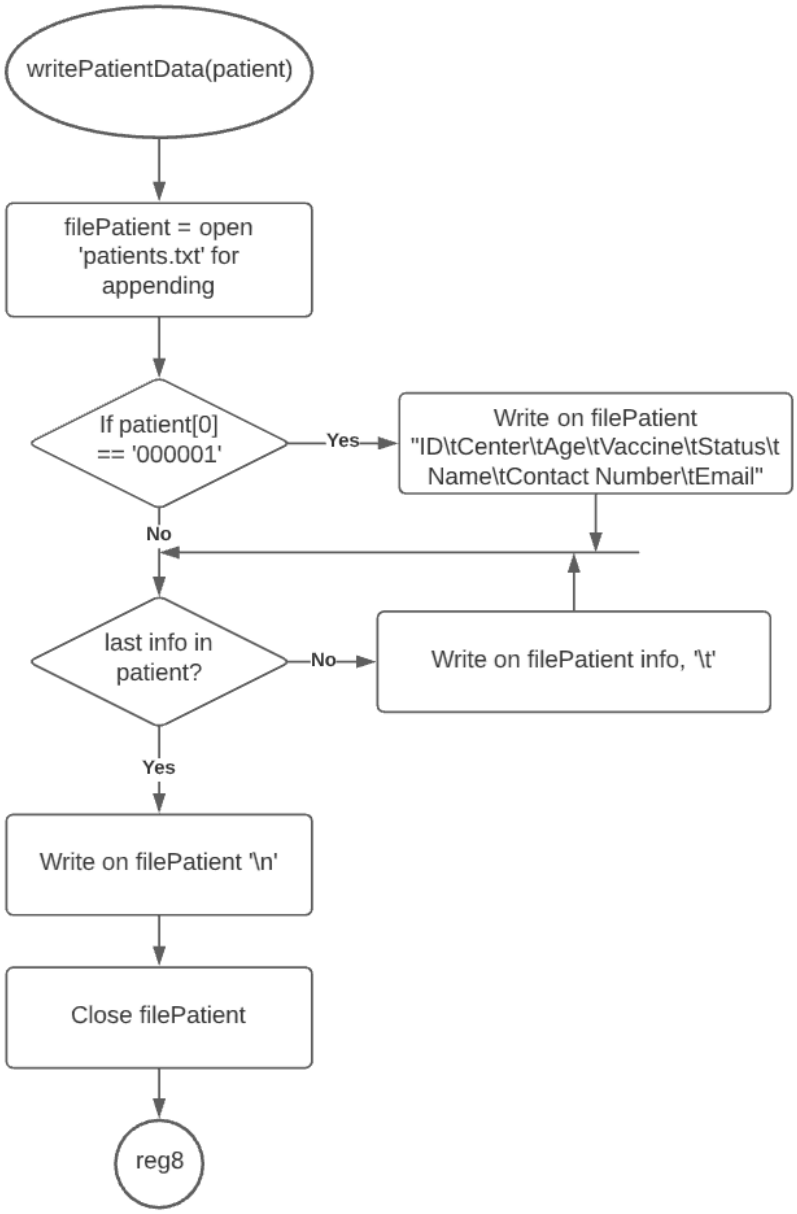

reg4
If 1 == 1
Yes
Print " Vaccine code: "
Read code
If code == 'AF'
AND age >= 12
Yes
No
If code == 'BV'
AND age >= 18
Yes
No
If code == 'CZ'
AND age >= 12 AND
age <= 45
Yes
Yes
If code =='DM'
AND age >= 12
Yes
No
If code == 'EC'
AND age >= 18
Yes
No
Print " Invalid input. Try
again."
reg5

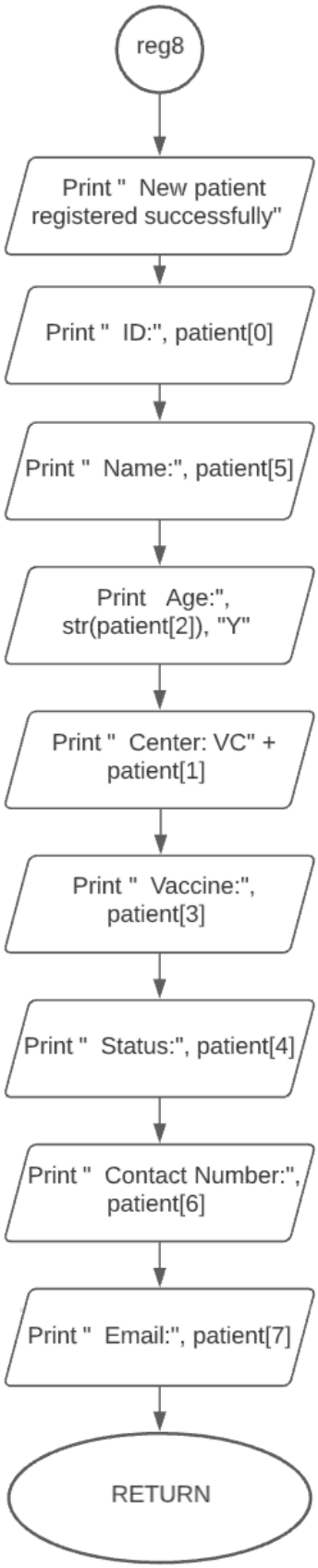

reg5
If 1==1
Yes
Print " Contact Number: "
Read contactNumber
If
contactNumber != ""
No
Print " Please provide a
contact number"
Yes
reg6

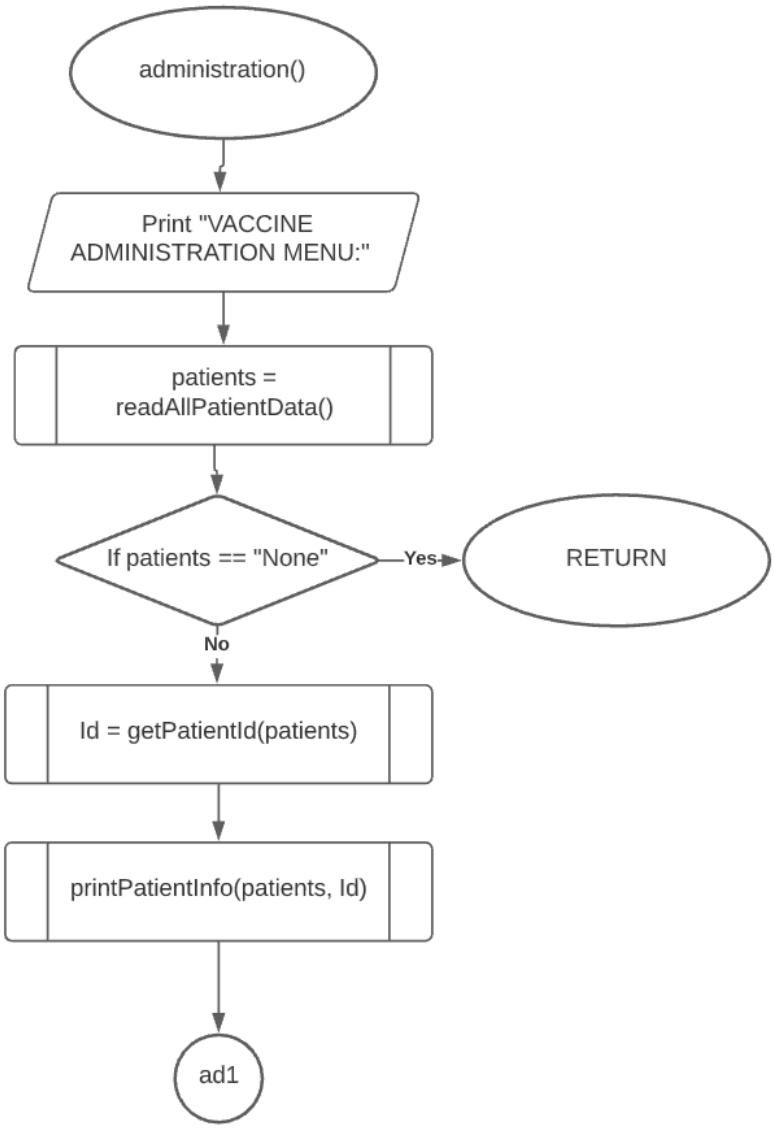

reg6
If 1==1
Yes
Print " Email: "
Read email
If email != ""
No
Print " Please provide an email address"
Yes
reg7

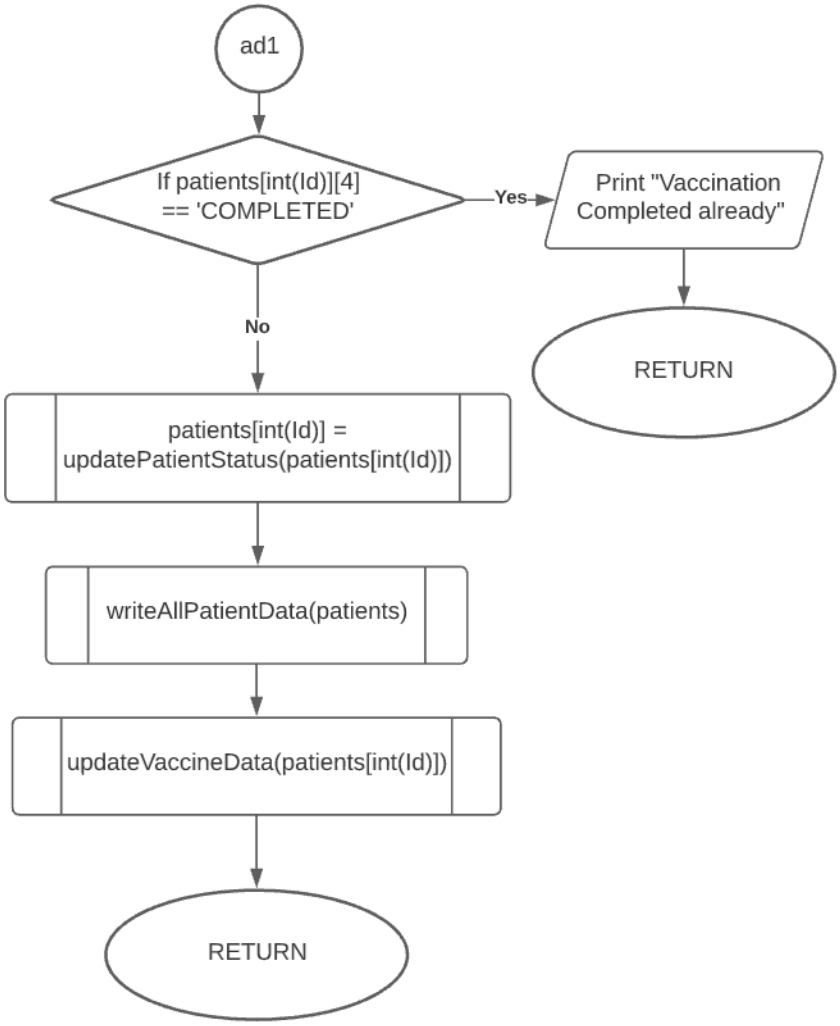

reg7
RETURN ['ID', center, age, code, 'NEW', name, contactNumber, email]

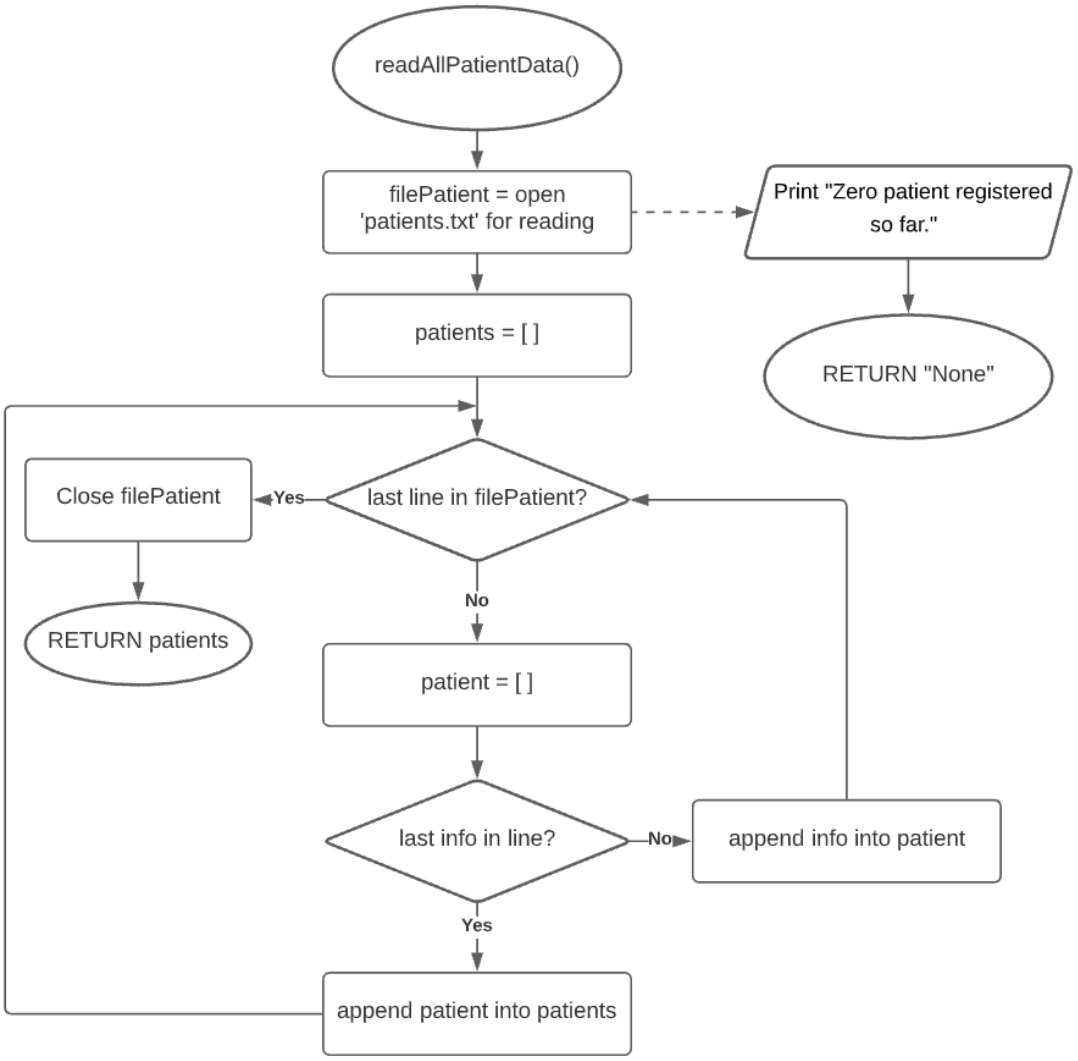

generateId()
filePatient = open 'patients.txt' for reading
line = last line of filePatient by reading it in reverse order
lastId = first word of 'line'
Convert lastId into integer
Close filePatient
Id = lastId + 1
Convert Id into string
Add leading zero to Id to make it a six digit number
RETURN Id
Convert lastId into integer
lastId = 0

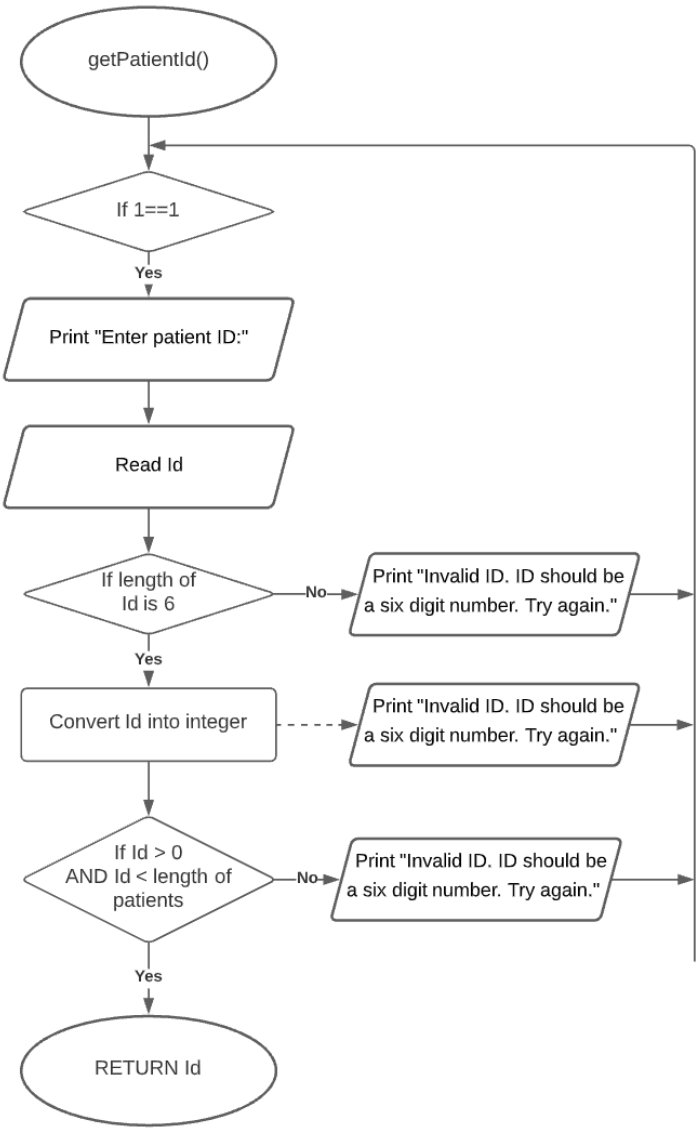

writePatientData(patient)
filePatient = open 'patients.txt' for appending
If patient[0] == '000001'
Yes
Write on filePatient "ID\tCenter\tAge\tVaccine\tStatus\t Name\tContact Number\tEmail"
No
last info in patient?
No
Write on filePatient info, '\t'
Yes
Write on filePatient '\n'
Close filePatient
reg8

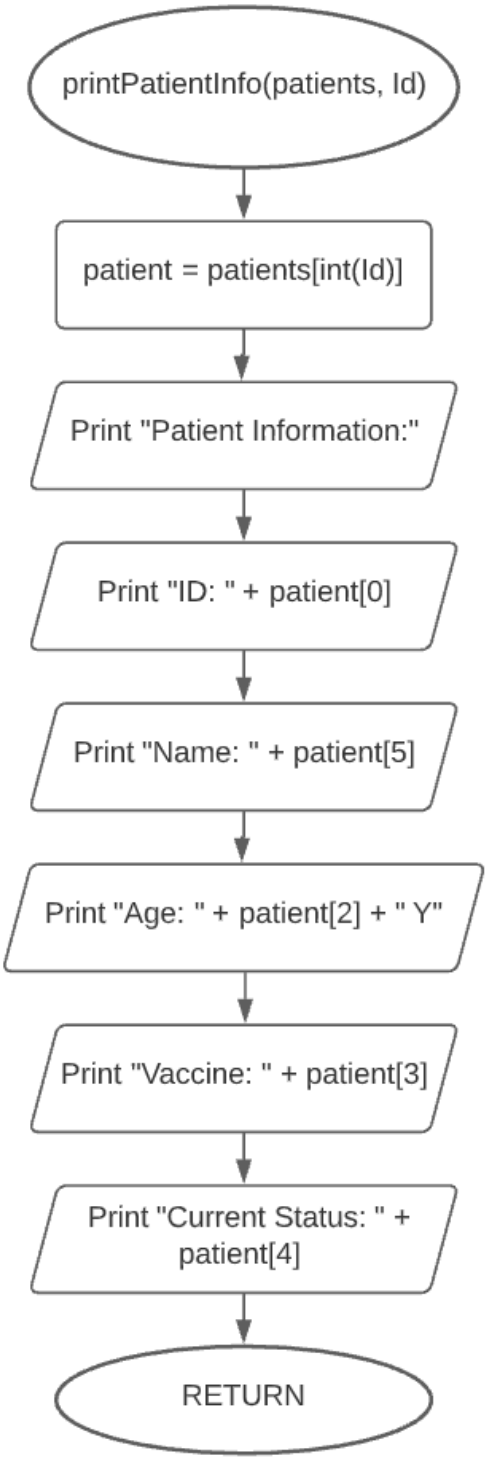

reg8
Print " New patient registered successfully"
Print " ID:", patient[0]
Print " Name:", patient[5]
Print Age:", str(patient[2]), "Y"
Print " Center: VC" + patient[1]
Print " Vaccine:", patient[3]
Print " Status:", patient[4]
Print " Contact Number:", patient[6]
Print " Email:", patient[7]
RETURN

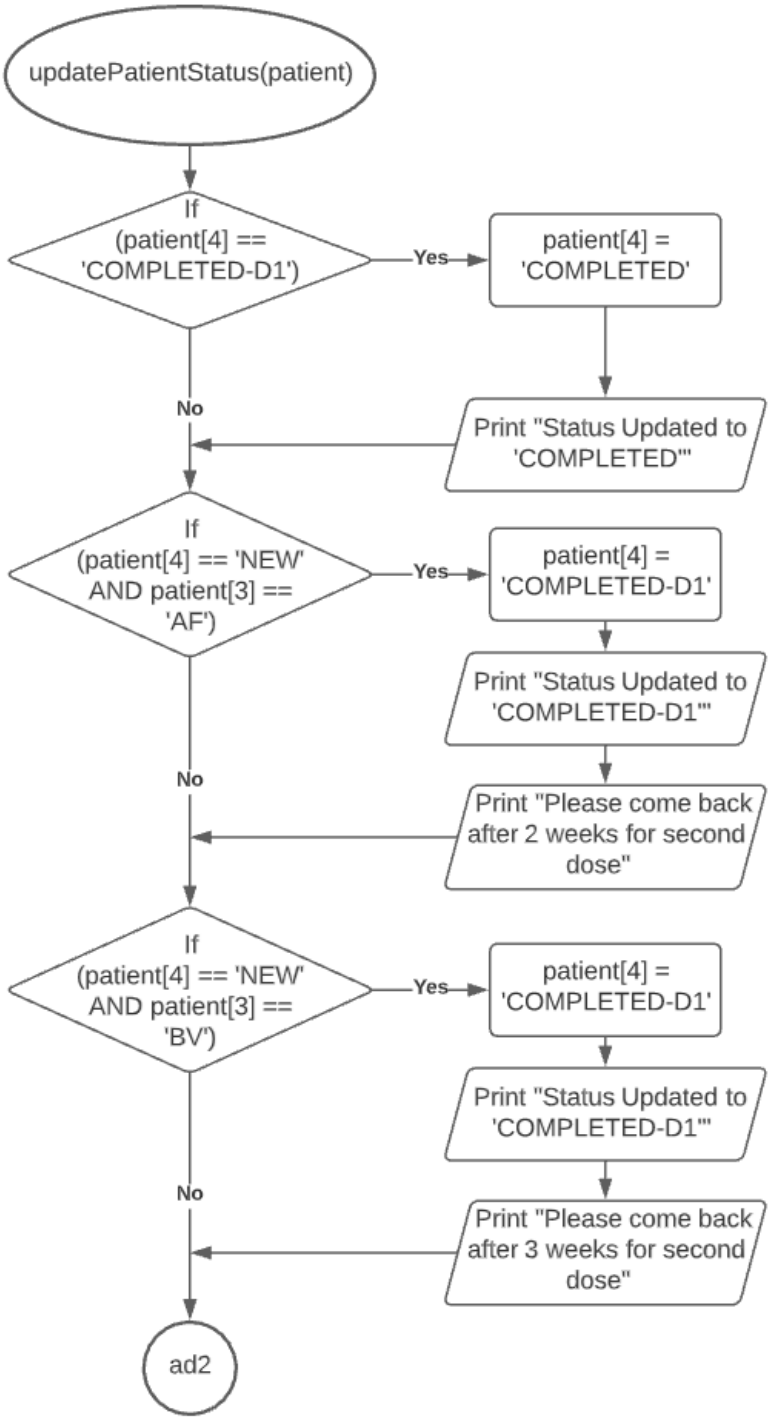

administration()
Print "VACCINE ADMINISTRATION MENU:"
patients = readAllPatientData()
If patients == "None"
Yes
RETURN
No
Id = getPatientId(patients)
printPatientInfo(patients, Id)
ad1

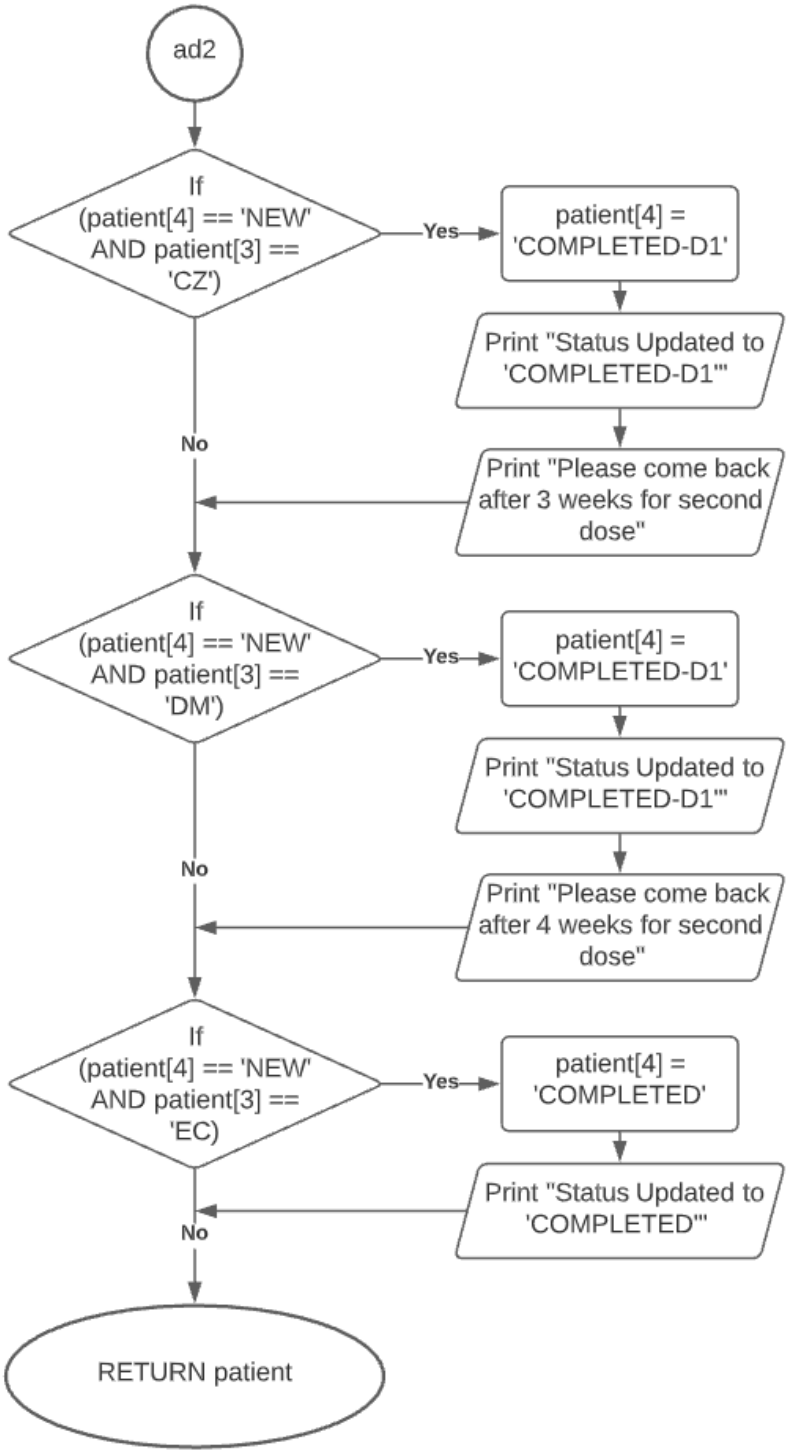

ad1
If patients[int(Id)][4]
== 'COMPLETED'
Yes
Print "Vaccination
Completed already"
RETURN
No
patients[int(Id)] =
updatePatientStatus(patients[int(Id)])
writeAllPatientData(patients)
updateVaccineData(patients[int(Id)])
RETURN

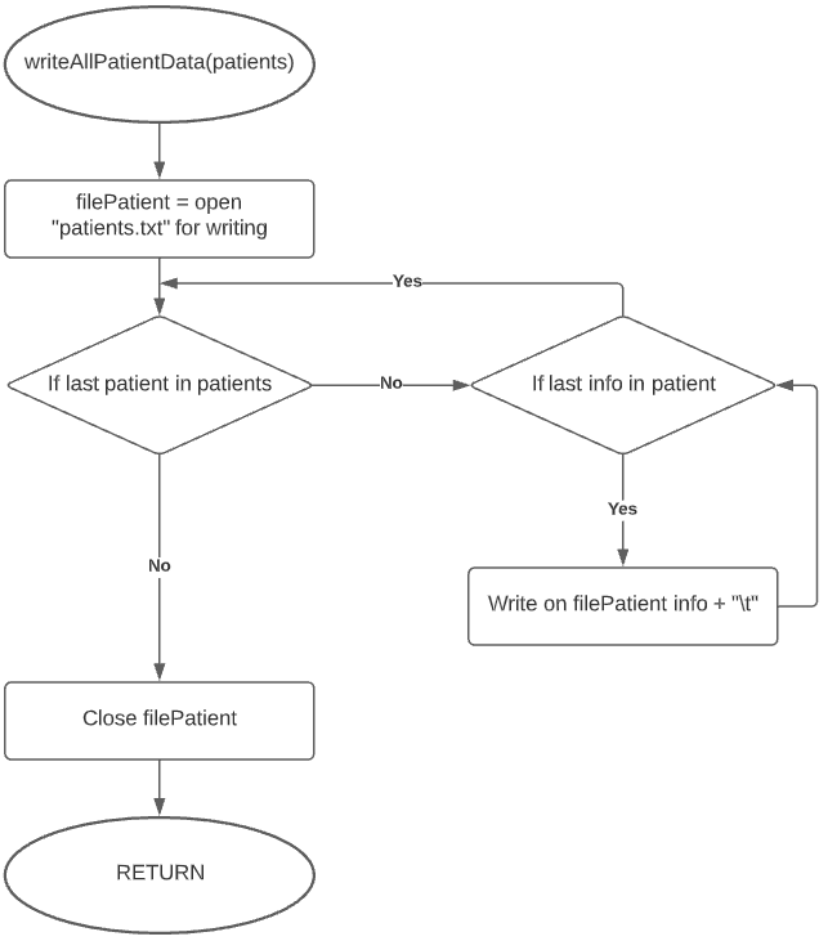

readAllPatientData()
filePatient = open 'patients.txt' for reading
Print "Zero patient registered so far."
RETURN "None"
patients = [ ]
Close filePatient
Yes
last line in filePatient?
RETURN patients
No
patient = [ ]
last info in line?
No
append info into patient
Yes
append patient into patients

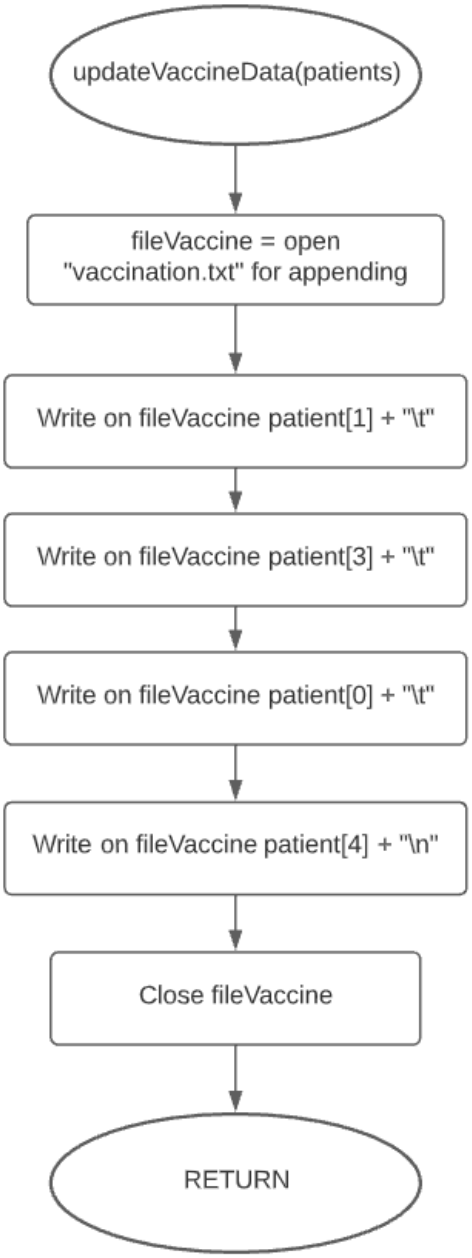

getPatientId()
If 1==1
Yes
Print "Enter patient ID:"
Read Id
If length of Id is 6
No
Print "Invalid ID. ID should be a six digit number. Try again."
Yes
Convert Id into integer
Print "Invalid ID. ID should be a six digit number. Try again."
If Id > 0 AND Id < length of patients
No
Print "Invalid ID. ID should be a six digit number. Try again."
Yes
RETURN Id

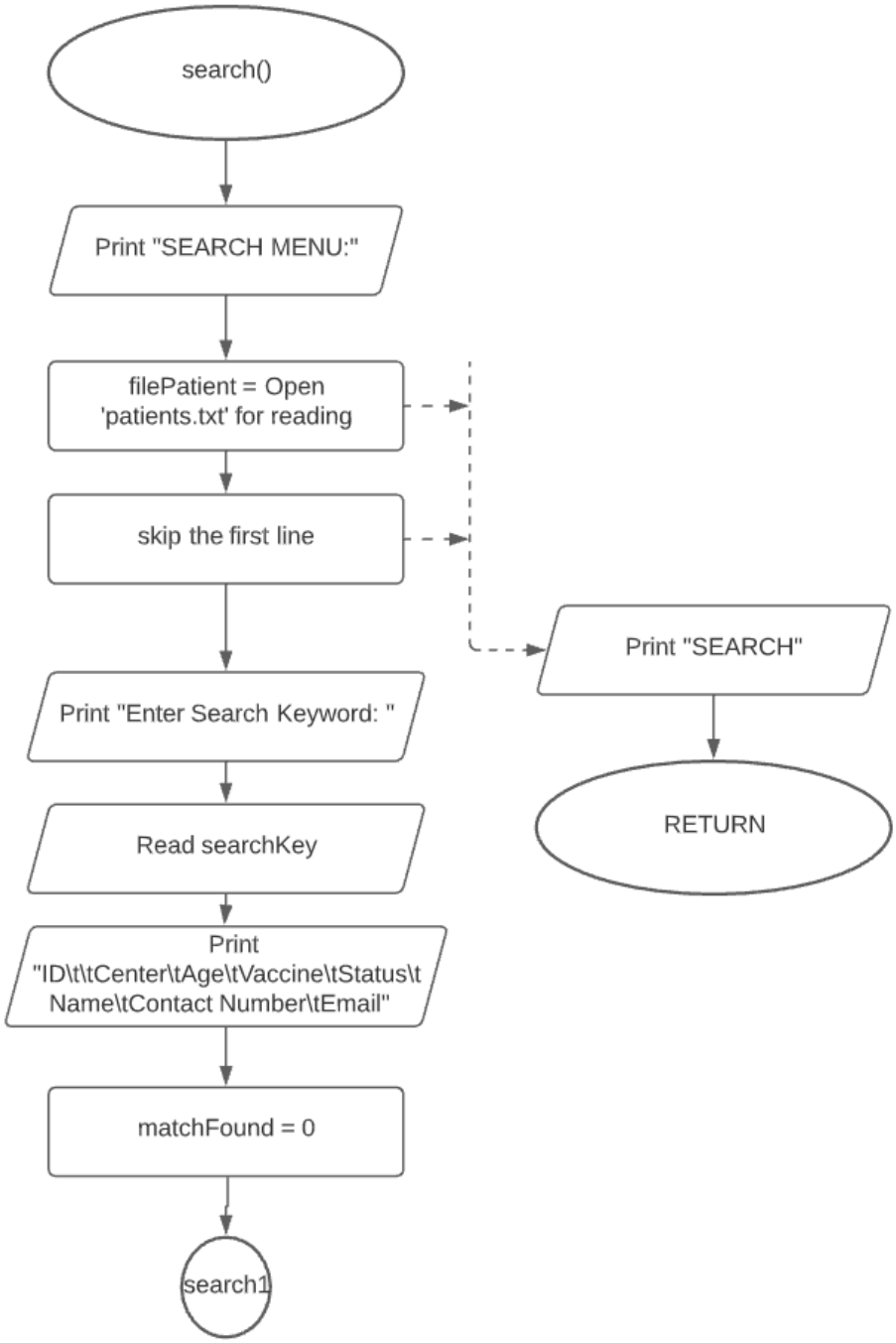

printPatientInfo(patients, Id)
patient = patients[int(Id)]
Print "Patient Information:"
Print "ID: " + patient[0]
Print "Name: " + patient[5]
Print "Age: " + patient[2] + " Y"
Print "Vaccine: " + patient[3]
Print "Current Status: " + patient[4]
RETURN

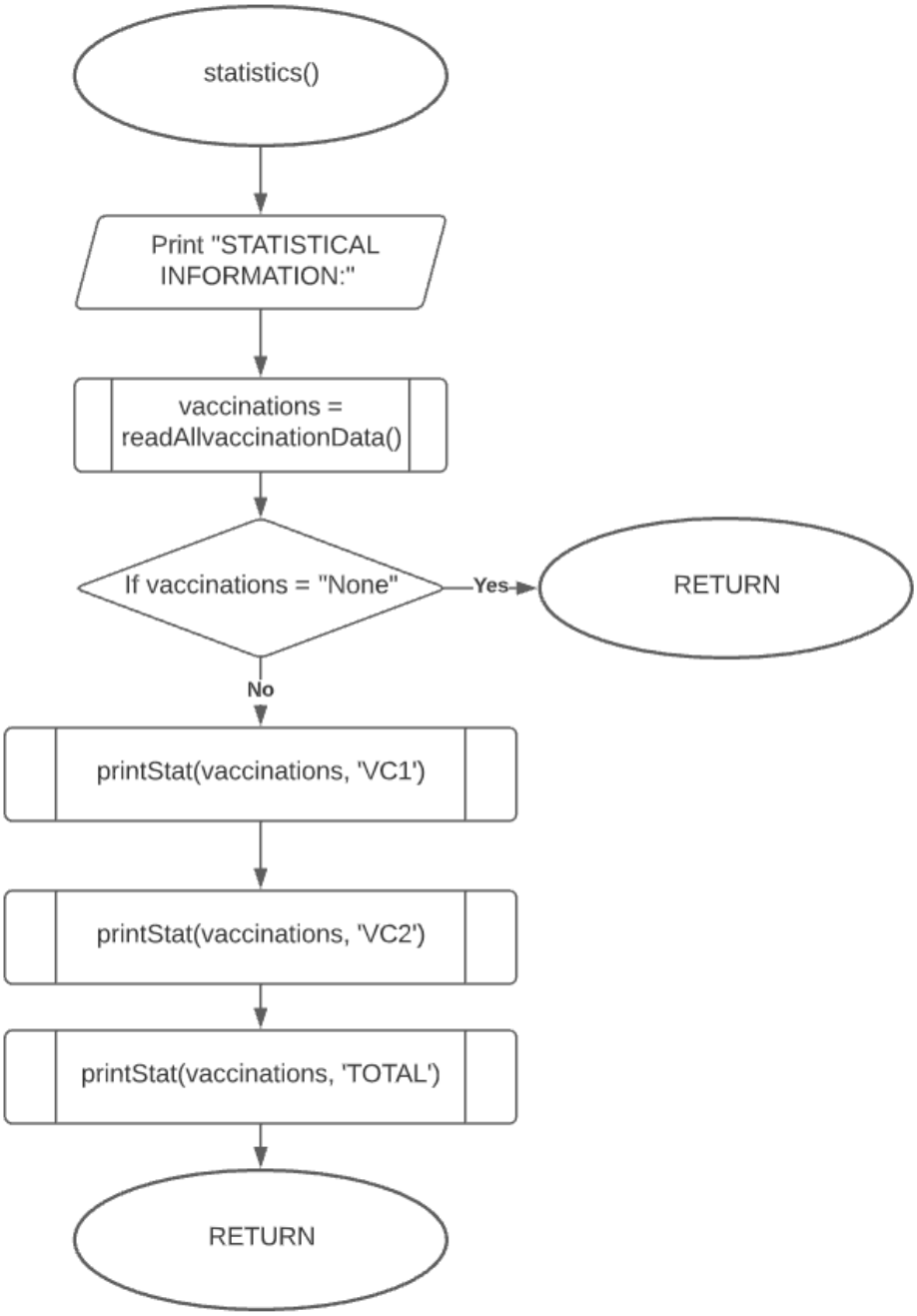

updatePatientStatus(patient)
If (patient[4] == 'COMPLETED-D1')
Yes
patient[4] = 'COMPLETED'
No
Print "Status Updated to 'COMPLETED'"
If (patient[4] == 'NEW' AND patient[3] == 'AF')
Yes
patient[4] = 'COMPLETED-D1'
Print "Status Updated to 'COMPLETED-D1'"
No
Print "Please come back after 2 weeks for second dose"
If (patient[4] == 'NEW' AND patient[3] == 'BV')
Yes
patient[4] = 'COMPLETED-D1'
Print "Status Updated to 'COMPLETED-D1'"
No
Print "Please come back after 3 weeks for second dose"
ad2

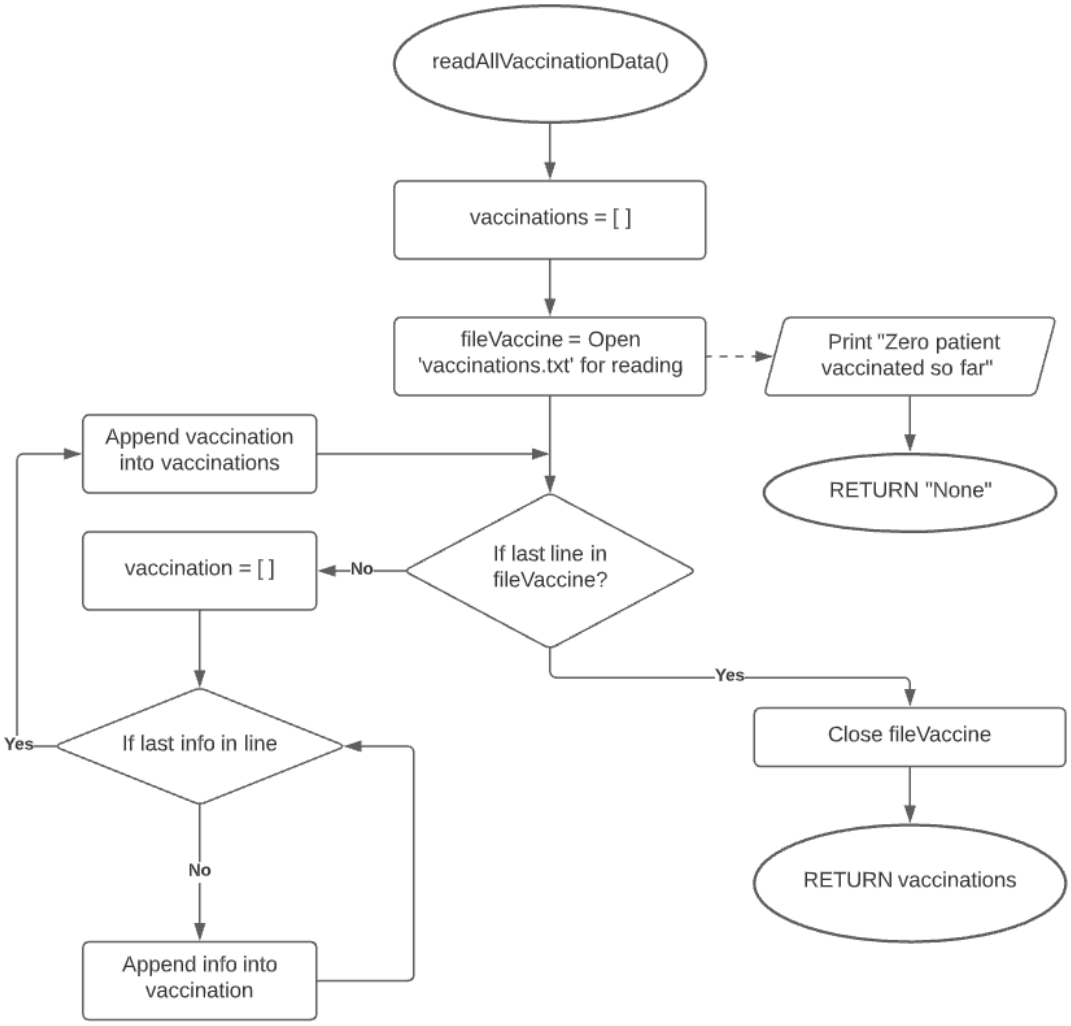

ad2
If (patient[4] == 'NEW' AND patient[3] == 'CZ')
Yes
patient[4] = 'COMPLETED-D1'
Print "Status Updated to 'COMPLETED-D1'"
No
Print "Please come back after 3 weeks for second dose"
If (patient[4] == 'NEW' AND patient[3] == 'DM')
Yes
patient[4] = 'COMPLETED-D1'
Print "Status Updated to 'COMPLETED-D1'"
No
Print "Please come back after 4 weeks for second dose"
If (patient[4] == 'NEW' AND patient[3] == 'EC)
Yes
patient[4] = 'COMPLETED'
Print "Status Updated to 'COMPLETED'"
No
RETURN patient

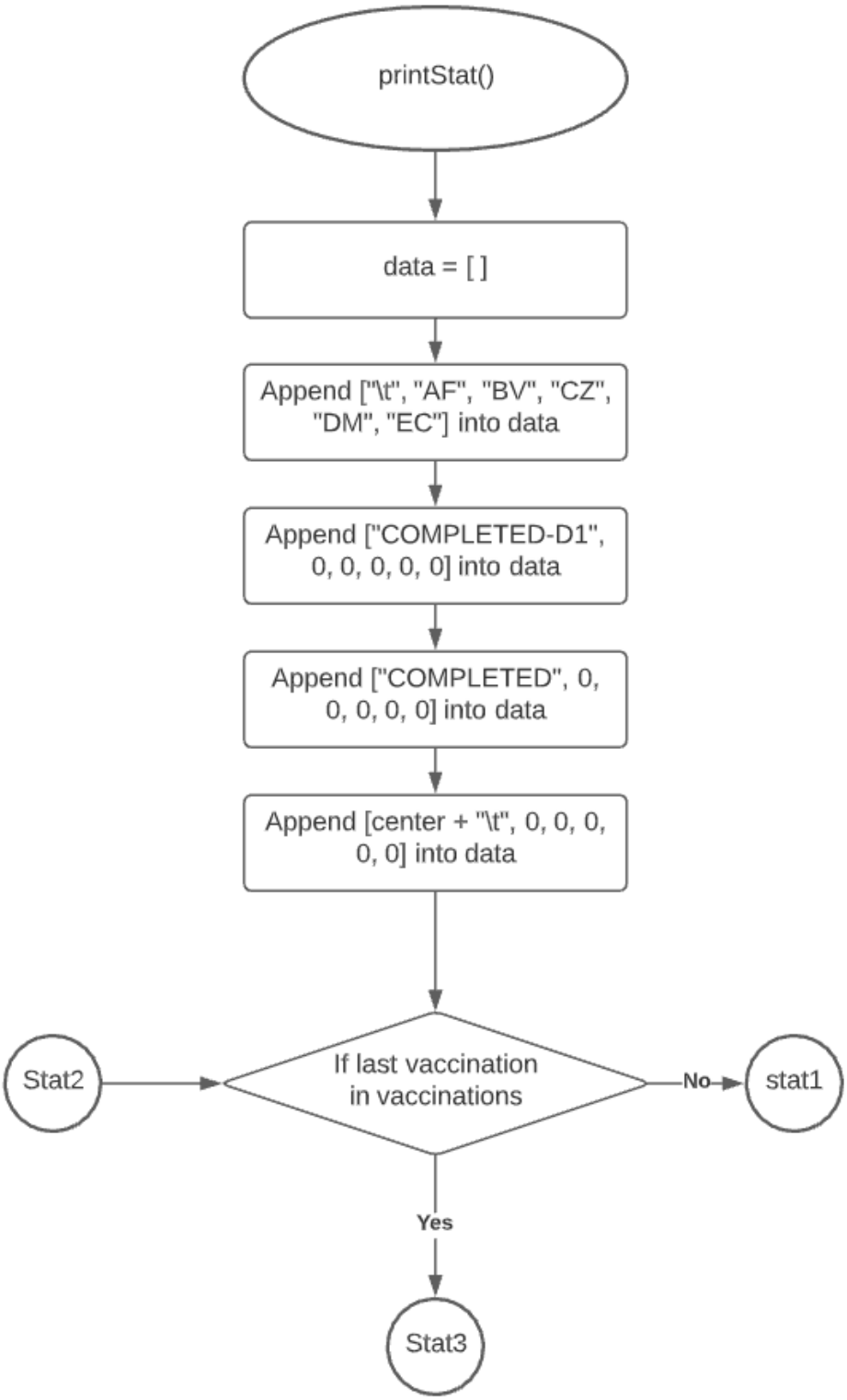

writeAllPatientData(patients)
filePatient = open "patients.txt" for writing
Yes
If last patient in patients
No
If last info in patient
Yes
No
Write on filePatient info + "\t"
Close filePatient
RETURN

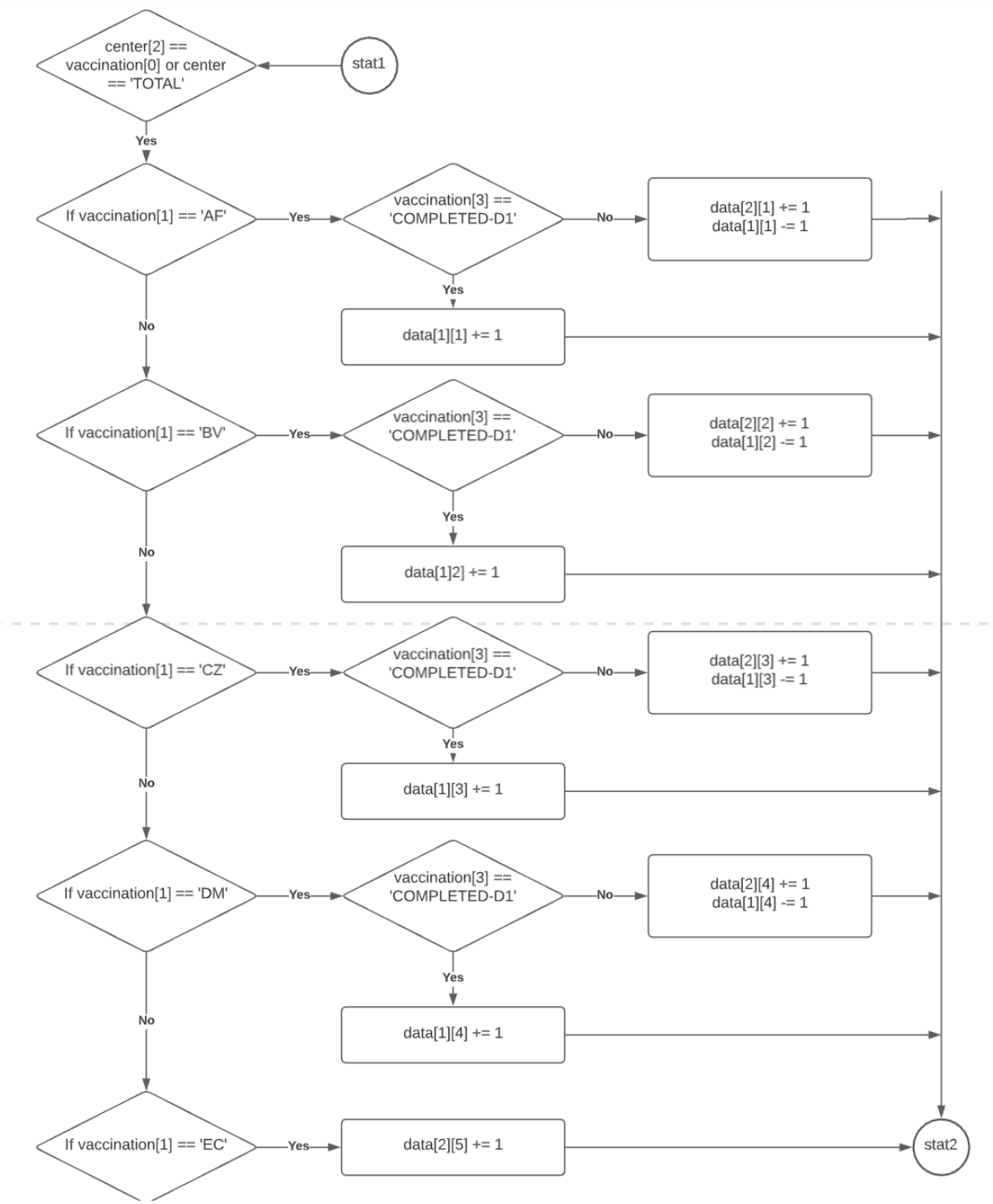

updateVaccineData(patients)
fileVaccine = open "vaccination.txt" for appending
Write on fileVaccine patient[1] + "\t"
Write on fileVaccine patient[3] + "\t"
Write on fileVaccine patient[0] + "\t"
Write on fileVaccine patient[4] + "\n"
Close fileVaccine
RETURN

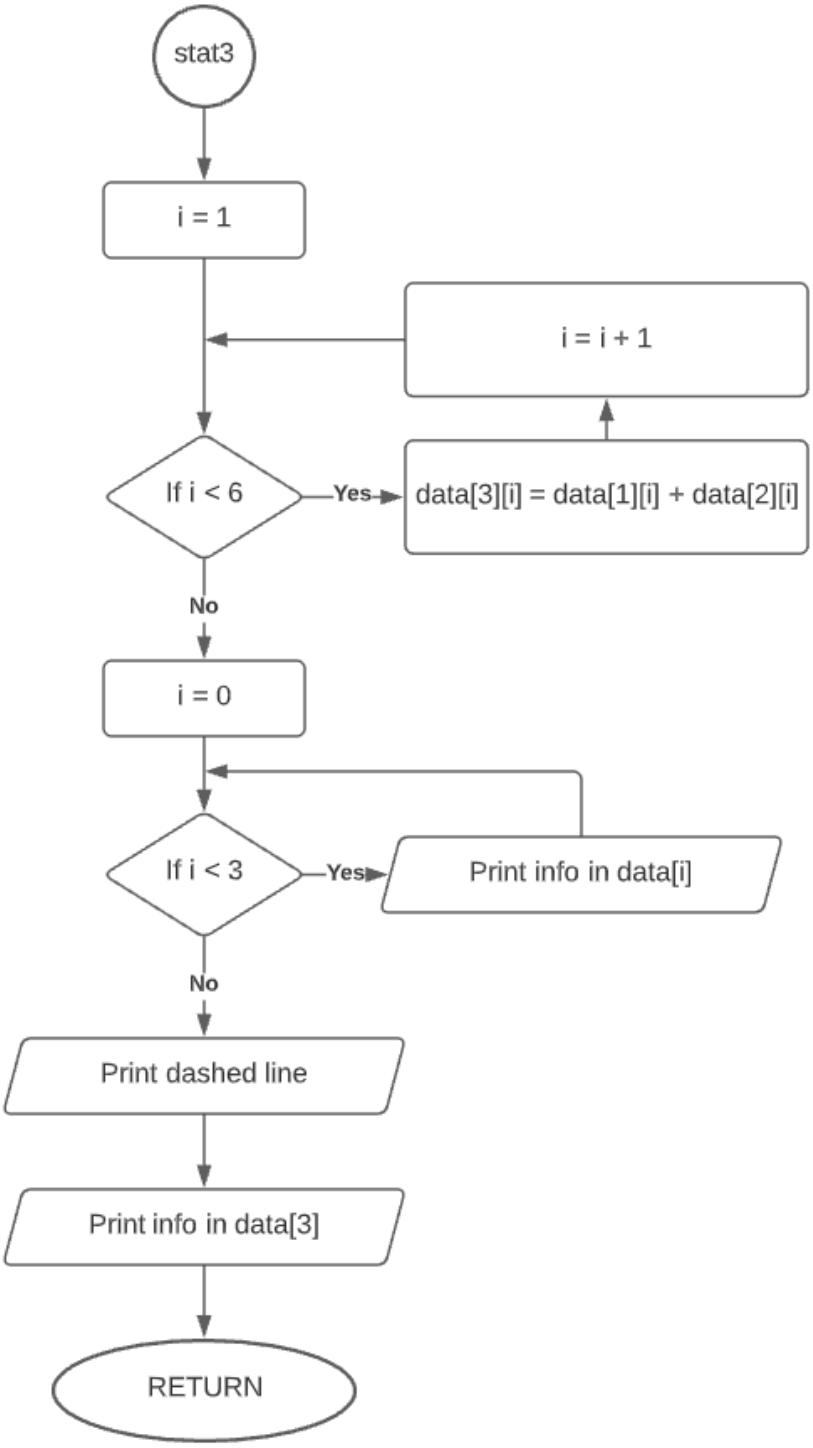

search()
Print "SEARCH MENU:"
filePatient = Open 'patients.txt' for reading
skip the first line
Print "SEARCH"
Print "Enter Search Keyword: "
RETURN
Read searchKey
Print "ID\t\tCenter\tAge\tVaccine\tStatus\t Name\tContact Number\tEmail"
matchFound = 0
search1


statistics()
Print "STATISTICAL INFORMATION:"
vaccinations = readAllvaccinationData()
If vaccinations = "None"
Yes
RETURN
No
printStat(vaccinations, 'VC1')
printStat(vaccinations, 'VC2')
printStat(vaccinations, 'TOTAL')
RETURN


readAllVaccinationData()
vaccinations = [ ]
fileVaccine = Open 'vaccinations.txt' for reading
Print "Zero patient vaccinated so far"
RETURN "None"
Append vaccination into vaccinations
If last line in fileVaccine?
No
vaccination = [ ]
Yes
Close fileVaccine
RETURN vaccinations
If last info in line
Yes
No
Append info into vaccination


printStat()
data = [ ]
Append ["\t", "AF", "BV", "CZ", "DM", "EC"] into data
Append ["COMPLETED-D1", 0, 0, 0, 0, 0] into data
Append ["COMPLETED", 0, 0, 0, 0, 0] into data
Append [center + "\t", 0, 0, 0, 0, 0] into data
Stat2
If last vaccination in vaccinations
No
stat1
Yes
Stat3


center[2] == vaccination[0] or center == 'TOTAL'
stat1
Yes
If vaccination[1] == 'AF'
Yes
vaccination[3] == 'COMPLETED-D1'
No
data[2][1] += 1
data[1][1] -= 1
Yes
data[1][1] += 1
No
If vaccination[1] == 'BV'
Yes
vaccination[3] == 'COMPLETED-D1'
No
data[2][2] += 1
data[1][2] -= 1
Yes
data[1]2] += 1
No
If vaccination[1] == 'CZ'
Yes
vaccination[3] == 'COMPLETED-D1'
No
data[2][3] += 1
data[1][3] -= 1
Yes
data[1][3] += 1
No
If vaccination[1] == 'DM'
Yes
vaccination[3] == 'COMPLETED-D1'
No
data[2][4] += 1
data[1][4] -= 1
Yes
data[1][4] += 1
No
If vaccination[1] == 'EC'
Yes
data[2][5] += 1
stat2


stat3
i = 1
i = i + 1
If i < 6
Yes
data[3][i] = data[1][i] + data[2][i]
No
i = 0
If i < 3
Yes
Print info in data[i]
No
Print dashed line
Print info in data[3]
RETURN

PROGRAM SOURCE CODE

Main Program

```
print("Welcome to COVID-19 Vaccination Record Management System\n")
menu()
```

Main program only contains two lines of codes. On the first line, it prints a welcome message. On the second line, it calls the menu function.

menu Function

```
def menu():
    while(True):

        #PRINT Home Menu
        print("HOME MENU:\n")

        print("  1. New Patient Registration")
        print("  2. Vaccine Administration")
        print("  3. Search Patient Record and Vaccination Status")
        print("  4. Statistical Information on Patients Vaccinated")
        print("  0. Exit")

        #INPUT - Home Menu Choice
        while(True):
            homeChoice = input("\n  Choose an option: ")
            if homeChoice in ['0', '1', '2', '3', '4']:
                break
            print("  Invalid input. Input must be a number between 0 and 4. Try again.")


        #Do task based on input
        if(homeChoice == '0'):
            print("\n  Exiting the program....")
            return

        if(homeChoice == '1'):
            registration()
            continue

        if(homeChoice == '2'):
            administration()
            continue

        if(homeChoice == '3'):
            search()
            continue

        if(homeChoice == '4'):
            statistics()
            continue
```

'menu' function has an infinite while loop in it. All the codes inside of the while loop are written in such a way that it keeps coming back to the home menu after completing a task.

The code inside of the while loop contains three sections. In the first section, from line 8-14, the function prints all the options of the home menu. In the second section, it takes an input from the user from line 17-21. The program keeps asking for the input until a valid input has been provided. If the input is invalid, the program shows an error message to help the user. In the last section, the program calls different functions based on the input provided. If the input is '0', the menu function returns, which results to exit the program. For other inputs from 1 to 4, it calls respective function.

registration Function

```
46 def registration():
47     print("\nNEW PATIENT REGISTRATION MENU:")
48 
49     #Take input for new patient and assign the info to 'patient' list
50     patient = regInput()
51 
52     #if the 'patient' holds no value, then return to Home Menu
53     if(patient == None):
54         return
55 
56     #generate next ID for new patient
57     patient[0] = generateId()
58 
59     #Update new patient info in file
60     writePatientData(patient)
```

Registration is one of the four major tasks of the program. In the 'registration' function, there are three significant sections just like the 'menu' function. In the first section, it calls 'regInput' function which takes all the inputs from the user and returns either a list containing all the information of the patient or "None". That return value is assigned to a variable named 'patient'. If it is "None" then the function returns right away and the program goes back to home menu.

If all the input has been taken properly and validated, the program then needs to generate a unique sequential ID for the patient. It is accomplished in the next section by calling 'generateId' function. This function returns a unique ID as a string and then the ID is replaced with the first element of 'patient' list, which was previously containing a place holder for ID.

In the last section, 'registration' function calls another function named 'writePatientData' and passes the 'patient' list as an argument. This function writes all the information from the 'patient'

list into the ‘patients.txt’ file. With that the registration process finishes and the program takes the user back to the home menu.

regInput Function

This function takes all necessary inputs from the user while a new patient is being registered. The inputs are Vaccination Center, Name, Age, Vaccine Type, Contact Number and Email. The function validates every input thoroughly and provides an error message to guide the user when the input is invalid. The program will be stuck in an infinite loop, repeatedly asking for input until it gets a valid one. For every information, the validation process might be different but the logic behind is similar.

```
64 def regInput():
65
66     #INPUT - Vaccination Center
67     while(True):
68         center = input("\n  Vaccination center [1/2]: ")
69         if(center == "1" or center == "2"):
70             break
71         print("  Invalid input. Input must be '1' or '2'. Try again.\n")
72
73     #INPUT - Name
74     while(True):
75         name = input("\n  Name: ")
76         if(name != ""):
77             break
78         print("  Please provide a name")
79
80     #INPUT - Age
81     while(True):
82         try:
83             age = float(input("\n  Age (in years): "))
84         except:
85             print("  Invalid input. Input must be a number. Try again.")
86             continue
87
88         if(age <= 0):
89             print("  Invalid input. Try again.")
90         else:
91             break
```

The input process starts by taking vaccination center input. The input is taken from inside of an infinite loop. If the input is valid, meaning it is either ‘1’ or ‘2’, it breaks out of the loop. Next input is the name, it is same as vaccination center, except the input is considered valid as long as it is not empty. As for age, it is validated in two steps. In the first one, the program checks whether

the input is a number or not using try-except. Age must be above 0 years to be considered valid, which is checked in the later step.

```
94      #PRINT - Available vaccines based on age
95      print("\n  Available vaccines: ")
96      if(age < 12):
97          print("    Sorry. No Vaccines available for this age.\n\n")
98          return None          #Return to home menu if no vaccine available
99      if(age >= 12):
100         print("    AF [2 Dosage with 2 weeks interval]")
101     if(age >= 18):
102         print("    BV [2 Dosage with 3 weeks interval]")
103     if(age >= 12 and age <= 45):
104         print("    CZ [2 Dosage with 3 weeks interval]")
105     if(age >= 12):
106         print("    DM [2 Dosage with 4 weeks interval]")
107     if(age >= 18):
108         print("    EC [1 Dosage]")
109
110
111     #INPUT - Vaccine type/code
112     while(True):
113         code = input("\n  Vaccine code: ")
114         if(code == "AF" and age >= 12):
115             break
116         if(code == "BV" and age >= 18):
117             break
118         if(code == "CZ" and age >= 12 and age <= 45):
119             break
120         if(code == "DM" and age >= 12):
121             break
122         if(code == "EC" and age >= 18):
123             break
124         print("  Invalid input. Try again.")
```

After getting age, the program prints the list of available vaccines based on age. For patients with age less than 12, there is no vaccines available. In such case, the program shows a message and returns ‘None’ as shown in line 98. Otherwise, it shows all the available vaccines and move on. Later, from line 112-124, the function takes input of vaccine type from inside of an infinite while loop. The input is checked with all the code of the vaccines along with age to make sure that the vaccine is appropriate for the patient’s age group. This way, the user will not be able to select a vaccine that is not in the list. If the input is valid, then the function breaks out of the loop. Otherwise, it shows an error message and repeatedly asks for input.

```
126     #INPUT - Contact Number
127     while(True):
128         contactNumber = input("\n  Contact Number: ")
129         if(contactNumber != ""):
130             break
131         print("  Please provide a contact number")
132 
133     #INPUT - Email
134     while(True):
135         email = input("\n  Email: ")
136         if(email != ""):
137             break
138         print("  Please provide an email address")
139 
140     #Return all the info in a list.
141     #First element of the list will be replaced later by ID
142     return ['ID', center, age, code, 'NEW', name, contactNumber, email]
```

Next two inputs, Contact Number and Email, are identical to Name input. The program keeps asking for an input if it gets an empty value.

To end the function, it puts all the value inside of a list, as we can see in line 142, where the first element is a placeholder for ID and then returns the list.

generateId Function

```
145 def generateId():
146 
147     #Try to get the last ID by reading the file in reverse
148     try:
149         filePatient = open("patients.txt", "r")      #Open file for reading
150 
151         for line in reversed(list(filePatient)):     #read the line of file in reverse
152             lastId = int(line.split()[0])            #read the first word of last line
153             break
154 
155         filePatient.close()                          #Close file
156     #If the file does not exist then set it to zero
157     except:
158         lastId = 0
159 
160 
161     Id = lastId + 1      #Next ID
162     Id = str(Id)         #Convert it to string
163     Id = Id.zfill(6)     #Add zeroes to make ID a six digit number
164 
165     return Id
```

This function creates a unique sequential ID. At first, it tries to open ‘patients.txt’ for reading. If the file exists then it reads the line of the file in reverse order as seen in line 151. In its first iteration, it reads the first word of the last line, which is the ID of the last patient, and breaks out of the loop.

If the program fails to open the file, which means it does not exist and it is the first patient ever, the program sets last ID as '0'. Then next ID is generated and converted into a six-digit number. Function returns the newly created 'Id'.

writePatientData Function

```
169 def writePatientData(patient):
170 
171     #Open file for appending
172     filePatient = open("patients.txt", "a")
173 
174     #Print Header Row for the first time
175     if(patient[0] == "1".zfill(6)):
176         filePatient.write("ID\tCenter\tAge\tVaccine\tStatus\tName\tContact Number\tEmail\n")
177 
178     #Write ever info about patient with TAB between them
179     for info in patient:
180         filePatient.write(str(info))
181         filePatient.write("\t")
182 
183     #Insert a new line
184     filePatient.write("\n")
185 
186     #Close file
187     filePatient.close()
188 
189     #Print patient information for confirmation
190     print("\n  New patient registered successfully\n")
191     print("  ID:", patient[0])
192     print("  Name:", patient[5])
193     print("  Age:", str(patient[2]), "Y")
194     print("  Center: VC" + patient[1])
195     print("  Vaccine:", patient[3])
196     print("  Status:", patient[4])
197     print("  Contact Number:", patient[6])
198     print("  Email:", patient[7])
199     print("")
```

This function writes all the information of new patient into the 'patients.txt' file. It opens the file for appending. If it is the first ID ever, then it writes the header row before doing anything. Then writes all the information separated by a tab. After writing into the file, it prints all the information, including ID, so that user can print the information and share it with the patient.

administration Function

```
203 def administration():
204     print("\nVACCINE ADMINISTRATION MENU:\n")
205 
206     #Read all patient data from patients.txt inside of patients 2D list
207     patients = readAllPatientData()
208 
209     #If its empty return to home menu
210     if (patients == None):
211         return
212 
213     #Get the ID of the patient who came to take their first/second dose
214     Id = getPatientId(patients)
215 
216     #Print patient info for confirmation
217     printPatientInfo(patients, Id)
218 
219     #If the patient completed vaccination already return to home menu
220     if (patients[int(Id)][4] == 'COMPLETED'):
221         print("\n  Vaccination Completed already\n")
222         return
223 
224     #Update the Status of the specific patient inside of patients 2D list
225     patients[int(Id)] = updatePatientStatus(patients[int(Id)])
226 
227     #Rewrite the patients.txt using patients 2D list
228     writeAllPatientData(patients)
229 
230     #Update vaccination.txt
231     updateVaccineData(patients[int(Id)])
```

This function handles the second major task of the program, which is to update the vaccination status of the patients. In the beginning of the function, it calls ‘readAllPatientData’ function which returns either all the patients’ information in a 2D list or ‘None’. If it is ‘None’ then the function returns right away to home menu. Then program calls ‘getPatientId’ function to get the ID of the patient and calls ‘printPatientInfo’ to print all the information about the patient with ID on the screen. If the patient has completed their vaccination already, then the program returns to home menu as we can see in line 220-222. Vaccination status of the patient is then updated by targeting the 2D ‘patients’ list using their ID and ‘updatePatientStatus’ function in line 225. The updated ‘patients’ list is passed into the ‘writeAllPatientData’ which overwrites the whole ‘patient.txt’ file. Information about the specific patient is then passed into the ‘updatedVaccineData’ to update the information in ‘vaccination.txt’ file.

readAllPatientData Function

```
252 def readAllPatientData():
253 
254     #Try opening patients.txt file
255     try:
256         filePatient = open("patients.txt", "r")
257     except:
258         print("  Zero patient registered so far.")
259         return None
260 
261     #Initialize patients list
262     patients = []
263 
264     #Update the patients 2D list with all data from patientx.txt
265     for line in filePatient:
266         patient = []
267 
268         for info in line.split("\t"):
269             patient.append(info.rstrip())
270         patients.append(patient)
271 
272     filePatient.close()
273 
274     return patients
```

This function reads the whole ‘patients.txt’ and creates a 2D list where every index of the primary list is a patient, and the secondary list is the information about them. It accomplishes this task by two for loops and python’s ‘split’, ‘rstrip’ and ‘append’ functions. In case the ‘patients.txt’ does not exist, it goes back to home menu. If everything turns out fine, the function returns the 2D ‘patients’ list.

getPatientId Function

```
234 def getPatientId(patients):
235     #INPUT - Patient ID
236     while(True):
237         Id = input("  Enter patient ID: ")
238         if(len(Id) == 6):
239             try:
240                 if(int(Id) > 0 and int(Id) < len(patients)):
241                     break
242                 else:
243                     print("  Invalid ID. ID does not exist. Try again.\n")
244             except:
245                 print("  Invalid ID. ID should be a six digit number. Try again.\n")
246         else:
247             print("  Invalid ID. ID should be a six digit number. Try again.\n")
248 
249     return Id
```

The task of this function is exactly as it sounds. It takes gets the ID of the patient from the user. The specialty is that it has a few layers of input validation. First it checks if the ID is six-digit or not, as all IDs in our system is six-digit. Then it checks the ID for numerical string and if the ID is within our total range of the IDs. If the ID turns out to be valid, then the function returns the ID, otherwise it repeatedly asks for input.

printPatientInfo Function

```
277 def printPatientInfo(patients, Id):
278     patient = patients[int(Id)]
279 
280     print("\n  Patient Information:")
281     print("    ID: " + patient[0])
282     print("    Name: " + patient[5])
283     print("    Age: " + patient[2] + " Y")
284     print("    Vaccine: " + patient[3])
285     print("    Current Status: " + patient[4])
```

This function is very simple. It just addresses the ‘patients’ list using the ‘Id’ as index and prints all the information about the patient who came for vaccination on the screen.

updatePatientStatus Function

```
287 def updatePatientStatus(patient):
288     if(patient[4] == 'COMPLETED-D1'):
289         patient[4] = 'COMPLETED'
290         print("\n  Status Updated to 'COMPLETED'\n")
291         return patient
292 
293     if(patient[4] == 'NEW' and patient[3] == 'AF'):
294         patient[4] = 'COMPLETED-D1'
295         print("\n  Status Updated to 'COMPLETED-D1'")
296         print("  Please come back after 2 weeks for second dose\n")
297         return patient
298 
299     if(patient[4] == 'NEW' and patient[3] == 'BV'):
300         patient[4] = 'COMPLETED-D1'
301         print("\n  Status Updated to 'COMPLETED-D1'")
302         print("  Please come back after 3 weeks for second dose\n")
303         return patient
304 
305     if(patient[4] == 'NEW' and patient[3] == 'CZ'):
306         patient[4] = 'COMPLETED-D1'
307         print("\n  Status Updated to 'COMPLETED-D1'")
308         print("  Please come back after 3 weeks for second dose\n")
309         return patient
310 
311     if(patient[4] == 'NEW' and patient[3] == 'DM'):
312         patient[4] = 'COMPLETED-D1'
313         print("\n  Status Updated to 'COMPLETED-D1'")
314         print("  Please come back after 4 weeks for second dose\n")
315         return patient
316 
317     if(patient[4] == 'NEW' and patient[3] == 'EC'):
318         patient[4] = 'COMPLETED'
319         print("\n  Status Updated to 'COMPLETED'\n")
320         return patient
```

This function contains 6 ‘if’ condition. First one is for the patients who completed dose 1, it changes their status to ‘COMEPLTED’. Next four ‘if’ conditions are for new patients, which changes their status to ‘COMPLETED-D1’. And the last one for new patients for ‘EC’ vaccine. It changes status to ‘COMPLETED’.

writeAllPatientData Function

```
324 def writeAllPatientData(patients):
325     filePatient = open("patients.txt", "w")
326 
327     for patient in patients:
328         for info in patient:
329             filePatient.write(str(info))
330             filePatient.write("\t")
331 
332         filePatient.write("\n")
333 
334     filePatient.close()
```

The function overwrites the whole ‘patients.txt’ file with the information from 2D ‘patients’ list. Every element of the primary list is a line and every element of secondary list is an information separated with tab.

updateVaccineData Function

```
336 def updateVaccineData(patient):
337     fileVaccine = open("vaccination.txt", "a")
338 
339     fileVaccine.write(patient[1] + "\t")    #Center
340     fileVaccine.write(patient[3] + "\t")    #Vaccine
341     fileVaccine.write(patient[0] + "\t")    #ID
342     fileVaccine.write(patient[4] + "\n")    #Status
343 
344     fileVaccine.close()
```

This function appends a line in the ‘vaccination.txt’ file every time it runs. In the line, there are four information about a patient: Center, Vaccine, ID, Status. It gets all these information from the argument.

search Function

```
346 def search():
347     print("\nSEARCH MENU:\n")
348 
349     #OPEN file
350     try:
351         filePatient = open("patients.txt", "r")
352         next(filePatient)
353     except:
354         print("  Zero patient registered so far\n")
355         return
356 
357     #INPUT - Search Keyword
358     searchKey = input("  Enter Search Keyword: ")
359 
360     #PRINT - header row
361     print("\n  ID\tCenter\tAge\tVaccine\tStatus\tName\tContact Number\tEmail")
362 
363     matchFound = 0
364 
365     for line in filePatient:
366         line = line.rstrip()
367 
368         if searchKey.lower() in line.lower():
369             print("  " + line)
370             matchFound += 1
371 
372     print("\n  Total Match Found = ", matchFound, "\n")
373 
374     filePatient.close()
```

This function accomplishes the third major task of the program. It tries to open the file for reading. If it can, then it skips the first row of the file, which is header row. If it cannot, then the function returns to home menu. Then it takes the ‘searchKey’ input, prints header row to make it more user friendly. The file handler runs through every line using a for loop and if the ‘searchKey’ is found in that line, it prints the whole line and increases ‘matchFound’ variable by 1.

statistics Function

```
376 def statistics():
377     print("\nSTATISTICAL INFORMATION:\n")
378 
379     #Read all data from vaccinations.txt into vaccinations list
380     vaccinations = readAllVaccinationData()
381 
382     #if the list is empty return to home menu
383     if(vaccinations == None):
384         return
385 
386     #Print 3 statatistics table: Center 1, Center 2, Total
387     printStat(vaccinations, 'VC1')
388     printStat(vaccinations, 'VC2')
389     printStat(vaccinations, 'TOTAL')
```

This function is the last of four major functions. It reads the whole ‘vaccination.txt’ file using ‘readAllVaccinationData’ which either returns a 2D list or ‘None’. Then it prints statistics in three tables using ‘printStat’ function three times. Statistics are for VC1, VC2 and total.

readAllVaccinationData Function

```
396 def readAllVaccinationData():
397     #Initializing an empty list
398     vaccinations = []
399 
400     #OPEN file for reading
401     try:
402         fileVaccine = open("vaccination.txt", "r")
403     except:
404         print("  Zero patient vaccinated so far")
405         return None
406 
407     #Iterate through every line in the file
408     #Convert line into lists
409     for line in fileVaccine:
410         vaccination = []
411 
412         for info in line.split("\t"):
413             vaccination.append(info.rstrip())
414 
415         vaccinations.append(vaccination)
416 
417     #CLOSE file
418     fileVaccine.close()
419 
420     return vaccinations
```

The function gets all the information from the 'vaccination.txt' and creates a 2D list named 'vaccinations' where every element of the primary list refers to a line and every element of the secondary list contains information from the line. The function either returns the 'vaccinations' list or 'None' if 'vaccination.txt' file does not exist.

printStat Function

```
422 def printStat(vaccinations, center):
423     #Initialize the data with header row and necessary info
424     data = []
425 
426     data.append(["\t", "AF", "BV", "CZ", "DM", "EC"])
427     data.append(["COMPLETED-D1", 0, 0, 0, 0, 0])
428     data.append(["COMPLETED", 0, 0, 0, 0, 0])
429     data.append([center + "\t", 0, 0, 0, 0, 0])
430 
431     #Update the number inside of data
432     for vaccination in vaccinations:
433         if(center[2] == vaccination[0] or center == 'TOTAL'):
434             if(vaccination[1] == 'AF'):
435                 if(vaccination[3] == 'COMPLETED-D1'):
436                     data[1][1] += 1
437                 else:
438                     data[2][1] += 1
439                     data[1][1] -= 1
440 
441             elif(vaccination[1] == 'BV'):
442                 if(vaccination[3] == 'COMPLETED-D1'):
443                     data[1][2] += 1
444                 else:
445                     data[2][2] += 1
446                     data[1][2] -= 1
447 
448             elif(vaccination[1] == 'CZ'):
449                 if(vaccination[3] == 'COMPLETED-D1'):
450                     data[1][3] += 1
451                 else:
452                     data[2][3] += 1
453                     data[1][3] -= 1
454 
455             elif(vaccination[1] == 'DM'):
456                 if(vaccination[3] == 'COMPLETED-D1'):
457                     data[1][4] += 1
458                 else:
459                     data[2][4] += 1
460                     data[1][4] -= 1
461 
462             elif(vaccination[1] == 'EC'):
463                 data[2][5] += 1
```

In the beginning of the function, a 2D list is initialized which later would be used to print all the information in a table. The table would have a header row with all the vaccine types, first column with status and last row with sum of the second and third row. Then the information is updated using a for loop which iterates through every index of 'vaccinations' 2D list. If a patient with 'COMPLETED-D1' is found, then the second row is updated and if a patient with 'COMPLETED' is found then second and third both rows get updated. This is because the patients with 'COMPLETED' status comes twice in the file.

```
465     #Add the numbers
466     for i in range(1, 6):
467         data[3][i] = data[1][i] + data[2][i]
468 
469     #Print 3 rows
470     for line in data[0:3]:
471         print("  ", end="")
472 
473         for info in line:
474             print(info, end="\t")
475 
476         print("")
477 
478     #Print a dashed line
479     print("  " + "-"*50)
480 
481     #Print the sum
482     print("  ", end="")
483     for info in data[3]:
484         print(info, end="\t")
485 
486     print("\n")
```

After updating the second and third rows, the program adds them to generate the last row. With that, the list contains all the information. Now is the time to print the list on the screen. First, it prints the first three rows. Then it prints a dashed line to make it easier for the user to understand that the last row is the sum. Then the last row gets printed on the screen.

SAMPLE INPUT/OUTPUT

Home Menu

```
Welcome to COVID-19 Vaccination Record Management System

HOME MENU:

  1. New Patient Registration
  2. Vaccine Administration
  3. Search Patient Record and Vaccination Status
  4. Statistical Information on Patients Vaccinated
  0. Exit

  Choose an option:
```

User will choose an option from the menu. After accomplishing the task, it come back to home again except if the input is '0'.

New Patient Registration

```
Welcome to COVID-19 Vaccination Record Management System

HOME MENU:

  1. New Patient Registration
  2. Vaccine Administration
  3. Search Patient Record and Vaccination Status
  4. Statistical Information on Patients Vaccinated
  0. Exit

  Choose an option: 1

NEW PATIENT REGISTRATION MENU:

  Vaccination center [1/2]: 2

  Name: Fawzan

  Age (in years): 25

  Available vaccines:
    AF [2 Dosage with 2 weeks interval]
    BV [2 Dosage with 3 weeks interval]
    CZ [2 Dosage with 3 weeks interval]
    DM [2 Dosage with 4 weeks interval]
    EC [1 Dosage]

  Vaccine code: DM

  Contact Number: 0123123123

  Email: fawzanalim@gmail.com

  New patient registered successfully

  ID: 000016
  Name: Fawzan
  Age: 25.0 Y
  Center: VC2
  Vaccine: DM
  Status: NEW
  Contact Number: 0123123123
  Email: fawzanalim@gmail.com
```

If the user chooses an option from the home menu, it will direct the user towards the new patient registration process. The program asks for all the inputs and lastly prints all the information for confirmation.

Vaccine Administration

```
HOME MENU:

  1. New Patient Registration
  2. Vaccine Administration
  3. Search Patient Record and Vaccination Status
  4. Statistical Information on Patients Vaccinated
  0. Exit

  Choose an option: 2

VACCINE ADMINISTRATION MENU:

  Enter patient ID: 000016

  Patient Information:
    ID: 000016
    Name: Fawzan
    Age: 25.0 Y
    Vaccine: DM
    Current Status: NEW

  Status Updated to 'COMPLETED-D1'
  Please come back after 4 weeks for second dose
```

```
HOME MENU:

  1. New Patient Registration
  2. Vaccine Administration
  3. Search Patient Record and Vaccination Status
  4. Statistical Information on Patients Vaccinated
  0. Exit

  Choose an option: 2

VACCINE ADMINISTRATION MENU:

  Enter patient ID: 000013

  Patient Information:
    ID: 000013
    Name: Jaber
    Age: 15.0 Y
    Vaccine: DM
    Current Status: COMPLETED-D1

  Status Updated to 'COMPLETED'
```

In the vaccine administration process, it only asks for nothing but ID. After updating status, it prints all the necessary information and the next appointment date.

Search Patient Record and Vaccination Status

```
HOME MENU:

  1. New Patient Registration
  2. Vaccine Administration
  3. Search Patient Record and Vaccination Status
  4. Statistical Information on Patients Vaccinated
  0. Exit

  Choose an option: 3

SEARCH MENU:

  Enter Search Keyword: NI

  ID            Center  Age     Vaccine Status  Name    Contact Number  Email
  000006        2       23.0    AF      NEW     Nick    018465454       nick@gmail.com
  000015        2       18.0    BV      NEW     Nishan  234324324       nishan@gmail.com

  Total Match Found =  2
```

If the user chooses option 3, it will direct the user to the search function of the program. It asks for search keyword, which can be anything. The program will print every match it finds on the screen.

Statistical Information on Patients Vaccinated

```
HOME MENU:

  1. New Patient Registration
  2. Vaccine Administration
  3. Search Patient Record and Vaccination Status
  4. Statistical Information on Patients Vaccinated
  0. Exit

  Choose an option: 4

STATISTICAL INFORMATION:

                AF      BV      CZ      DM      EC
  COMPLETED-D1  0       0       1       0       0
  COMPLETED     0       1       0       1       2
  ------------------------------------------------
  VC1           0       1       1       1       2

                AF      BV      CZ      DM      EC
  COMPLETED-D1  1       1       1       3       0
  COMPLETED     1       1       0       0       1
  ------------------------------------------------
  VC2           2       2       1       3       1

                AF      BV      CZ      DM      EC
  COMPLETED-D1  1       1       2       3       0
  COMPLETED     1       2       0       1       3
  ------------------------------------------------
  TOTAL         2       3       2       4       3
```

Fourth option on the home menu leads to statistical information. It does not take any inputs, it outputs three table: one for VC1, one for VC2 and lastly Total.

Exiting the Program

```
HOME MENU:

  1. New Patient Registration
  2. Vaccine Administration
  3. Search Patient Record and Vaccination Status
  4. Statistical Information on Patients Vaccinated
  0. Exit

  Choose an option: 0

  Exiting the program....
>>>
```

The program keeps repeating and coming back to home page after accomplishing each task. If the input is zero, the program exits right away.

Input Validation

Here's some of the input validation used in the program to prevent it from crashing

```
HOME MENU:

  1. New Patient Registration
  2. Vaccine Administration
  3. Search Patient Record and Vaccination Status
  4. Statistical Information on Patients Vaccinated
  0. Exit

  Choose an option: ABC
  Invalid input. Input must be a number between 0 and 4. Try again.

  Choose an option: 5
  Invalid input. Input must be a number between 0 and 4. Try again.

  Choose an option: 1
```

```
NEW PATIENT REGISTRATION MENU:

  Vaccination center [1/2]: 3
  Invalid input. Input must be '1' or '2'. Try again.


  Vaccination center [1/2]: 2

  Name:
  Please provide a name

  Name: Alvi

  Age (in years):
  Invalid input. Input must be a number. Try again.

  Age (in years): -5
  Invalid input. Try again.

  Age (in years): a
  Invalid input. Input must be a number. Try again.

  Age (in years): 10

  Available vaccines:
    Sorry. No Vaccines available for this age.
```

```
  Available vaccines:
    AF [2 Dosage with 2 weeks interval]
    CZ [2 Dosage with 3 weeks interval]
    DM [2 Dosage with 4 weeks interval]

  Vaccine code: EC
  Invalid input. Try again.

  Vaccine code: AB
  Invalid input. Try again.

  Vaccine code: AF

  Contact Number:
  Please provide a contact number

  Contact Number: 0176514651
```

```
VACCINE ADMINISTRATION MENU:

  Enter patient ID: abcdef
  Invalid ID. ID should be a six digit number. Try again.

  Enter patient ID: 123456
  Invalid ID. ID does not exist. Try again.

  Enter patient ID: 000012

  Patient Information:
    ID: 000012
    Name: Ismail
    Age: 13.0 Y
    Vaccine: CZ
    Current Status: COMPLETED

  Vaccination Completed already
```

CONCLUSION

The objective of the assignment was to implement the knowledge we have gained in the module and research based on it. We had to experiment and research based on the things we have learned to meet the requirements of the assignment. The struggle along the way taught us a lot. We had to keep track of the big picture while working on every small portion of the code. Overall, it was a great learning experience.