INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT018-3-1-ICP

INTRODUCTION TO C PROGRAMMING

APU1F2002CS/CS(CYB)/CS(DF)/CS(IS)/EEE/IT/ME/PE/TE/

CGD/CE/IT(IOT)/CS(DA) UC1F2002CGD/CS(DA)/IT/IT(IOT)/CS

NAME: **LOKE WENG KHAY**

TP NUMBER: **TP062166**

HAND OUT DATE: **05 OCTOBER 2020**

HAND IN DATE: **04 NOVEMBER 2020**

WEIGHTAGE: **50%**

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

1.0 Table of Contents

2.0 Introduction

C programming language was initially developed in the early 1970s, which was developed by Dennis Ritchie at Bell Labs. C was the successor of the B programming language, and it was constructed to reimplement the kernel of the Unix operating system. Slowly from there, C gradually gained its popularity (Manley, 2020).

Although the C language is almost five-decade-old, C and C++ currently are widely used in the 21st century. This is because well-known operating systems such as Windows, Linux, and MAC were written mostly in C programming language to build their kernels. Moreover, database software such as MySQL, MS SQL Server, and Oracle Database mainly was coded in C and C++. Although C has changed over the years, it is still the most commonly used language to use in low-level programs such as kernels (Munoz, 2020).

2.1 Assumption

In this report, we will be explaining the assumption of this program, which is the Health Insurance System.

ZeeMediLife has order us to write a C program with the following features to manage the health insurance scheme for ZeeMediLife insurance company.

ZeeMediLife has introduced three different plans, which are Plan120, Plan150, and Plan200. The monthly premium for each of them is RM120, RM150, and RM200, respectively. An insurance subscriber of this scheme needs to select one of the plans based on his/her age eligibility.

There are two types of claim limits, which are the Annual Claim Limit and Lifetime Claim Limit. A subscriber can only subscribe to either one. For the Annual Claim Limit, a subscriber can claim up to the amount stated but until the age of 60. On the other hand, Lifetime Claim Limit allows a subscriber to claim for life until all the stated amount has been exhausted.

A subscriber can claim the normal wardroom and ICU charges based on eligibility under his/her plan. All subscribers are eligible for Hospital Supplies and Services, Surgical Fees, and other fees of any amount as long the charges are able to be covered by their plan's amount balance. The system should be able to search up the database and output the number of Annual Claim Limit member who has exhausted all their amount balance and also has another option to output the total amount claim of all Lifetime Claim Limit members

Last but not least, the system should be able to search for a particular member by using their name or id or age, plan type, and type of claim limit and output the user detail if found.

3.0 Design of Program

3.1 Pseudocode

```
PROGRAM Insurance Health System
BEGIN
        CALL mainScreenOption()
END

Function mainScreenOptions()
        DECLARE selection
        DECLARE answer=false
        DISPLAY “Enter Selection (1-5)”
        WHILE (answer != true)
                READ selection
                IF (selection==1)
                        SET answer=true
                        CALL insurancePlanSubscription()
                ELSE
                        IF (selection==2)
                                SET answer=true
                                CALL claimInsurance()
                        ELSE
                                IF (selection==3)
                                        SET answer=true
                                        CALL accountInformation()
                                ELSE
                                        IF (selection==4)
                                                SET answer=true
                                                CALL search()
                                        ELSE
                                                IF (selection==5)
```

```
                    SET answer=true
                    exit(0)
                ELSE
                    DISPLAY “Wrong input, enter again”
                ENDIF
            ENDIF
        ENDIF
    ENDIF
ENDIF
ENDWHILE
ENDFUNCTION
```

```
FUNCTION insurancePlanSubscription()
        DECLARE age, id
        DECLARE planType[10]
        DECLARE CL[4]
        DECLARE amount
        DECALRE name[20], contactNumber[15], healthHistory[100]
        DECLARE houseNumber[5], street[30], city[30],state[20], zipCode [6]

        DECLARE age1
        DECLARE firstAnswer = false
        DISPLAY “If subscriber is below 1 years old, please enter '-1' ”
        DISPLAY “Please enter age in Years:”
        WHILE (firstAnswer != true)
                READ age1
                IF (age1 == -1)
                        SET firstAnswer=true
                        age = CALL getAgeInDays()
                        CALL displayInsuranceOptions(age, true)
                        CALL getTypeofClaimLimit(CL)
                        amount = CALL getPlanType(age, CL, true, planType)
                        CALL getUserInformation(name, houseNumber, street, city, state,
                                                zipCode, contactNumber, healthHistory)
                        id = CALL getUserID()
                        CALL displayOverall(id, name, age, CL, planType, houseNumber,
                                        street, city, state, zipCode, contactNumber,
                                        healthHistory, true, amount)
                        CALL saveToAFile(id, name, age, CL, planType, houseNumber,
                                        street, city, state, zipCode, contactNumber,
                                        healthHistory, true, amount)
                ELSE
                        IF (age1 > 0)
                                SET firstAnswer = true
```

```
                    age = CALL getAgeInYears(age1)
                    CALL displayInsuranceOptions(age, false)
                    CALL getTypeofClaimLimit(CL)
                    amount = CALL getPlanType(age, CL, false, planType)
                    CALL getUserInformation(name, houseNumber, street, city,
                                            state, zipCode, contactNumber,
                                            healthHistory)
                    id = CALL getUserID()
                    CALL displayOverall(id, age, CL, planType, houseNumber,
                                        street, city, state, zipCode,
                                        contactNumber, healthHistory, false,
                                        amount)
                    CALL saveToAFile(id, name, age, CL, planType,
                                     houseNumber, street, city, state, zipCode,
                                     contactNumber, healthHistory, false,
                                     amount)
                ELSE
                    DISPLAY "Please enter an appropriate response"
                ENDIF
            ENDIF
        ENDWHILE
        RETURN 0
ENDFUNCTION

FUNCTION getAgeInDays()
        DECLARE ageInDays, j
        DECLARE ageInDays2=false
        DISPLAY "Please enter, how old is the subscriber in days:"
        FOR (j=0; ageInDays2 != true; j++)
            READ ageInDays
            IF (ageInDays > 14 AND ageInDays < 365)
                SET ageInDays2=true
                RETURN ageInDays
```

```
            ELSE
                IF ( ageInDays > 0 AND ageInDays < 15)
                    DISPLAY “Sorry, subscriber is not applicable for any insurance
                            as the minimum age in days must be above 15 days
                            old”
                    exit(0)
                ELSE
                    DISPLAY “Please enter an appropriate age”
                ENDIF
            ENDIF
        ENDFOR
ENDFUNCTION

FUNCTION getAgeInYears (age2)
        DECALRE i, ageInDays = age2
        DECALRE ageInDays2 = false
        FOR (i=0; ageInYears2 != true; i++)
            IF (ageInYears > 0 AND ageInYears < 55)
                SET ageInYears2=true
                RETURN ageInYears
            ELSE
                IF ( ageInYears > 54 )
                    DISPLAY “Sorry, subscriber is not applicable for any insurance
                            as the maximum age is cap at 54 years old”
                    exit(0)
                ELSE
                    DISPLAY “Please enter an appropriate age”
                ENDIF
            ENDIF
        ENDFOR
ENDFUNCTION
```

```
FUNCTION displayIsuranceOptions(age3 , ageInDays)
    IF((age3>0 AND age3<21) AND (ageInDays == false) OR (age3>14 AND age3<365)
       AND (ageInDays==true))
        DISPLAY Insurance Plans and Benefits for Plan120, Plan150 and Plan200
    ELSE
        IF ((age3>20 AND age3<41) AND (ageInDays==false))
            DISPLAY Insurance Plans and Benefits for Plan150 and Plan200
        ELSE
            DISPLAY Insurance Plans and Benefits for Plan200
        ENDIF
    ENDIF
ENDFUNCTION

FUNCTION getTypeofClaimLimit (CL2)
    DECLARE claimLimit[3]
    DECLARE k, l
    DECLARE typeOfClaimLimit = false
    DISPLAY "Please enter claim limit, ACL or LCL"
    FOR (k=0; typeOfClaimLimit != true; k++)
        READ claimLimit
        IF (claimLimit == "ACL")
            CL2 = claimLimit
            DISPLAY "You have selected: Annual Claim Limit"
            SET typeOfClaimLimit=true
        ELSE
            IF (claimLimit == "LCL")
                CL2 = claimLimit
                DISPLAY "You have selected: Lifetime Claim Limit"
                SET typeOfClaimLimit=true

            ELSE
```

```
                    DISPLAY “Enter an appropriate type of claim limit”
                ENDIF
            ENDIF
        ENDFOR
ENDFUNCTION

FUNCTION getPlanType (age3, CL3, ageInDays2, PT)
        DECLARE userPlanType=false
        DECALRE planType[9]
        DECALRE plan120[8]="Plan120", plan150[8]="Plan150", plan200[8]="Plan200"
        DECLARE m, n
        DISPLAY “Please enter a plan to go with”
        IF (CL3 == “ACL”)
            DISPLAY “Annual Claim Limit”
        ELSE
            DISPLAY “Lifetime Claim Limit”
        ENDIF
        FOR (m=0; userPlanType != true; m++)
            READ planType
             IF (((age3>0 AND age3<21) AND (ageInDays2==false)) OR ((age3>14
                AND age3<365) AND (ageInDays2==true)))
                IF ((planType == "Plan 120") OR (planType == "plan 120") OR
                   (planType == "Plan120") OR (planType == "plan120"))
                        SET userPlanType = true
                        PT = plan120
                        DISPLAY “You have selected: Plan 120”
                        IF (CL3 == “ACL”)
                            RETURN 120000
                        ELSE
                            RETURN 600000
                        ENDIF
```

```
        ELSE
            IF ((planType == "Plan 150") OR (planType == "plan 150") OR
               (planType == "Plan150") OR (planType == "plan150"))
                SET userPlanType = true
                PT = plan150
                DISPLAY “You have selected: Plan 150”
                IF (CL3 == “ACL”)
                    RETURN 150000
                ELSE
                    RETURN 750000
                ENDIF
        ELSE
            IF ((planType == "Plan 200") OR (planType == "plan 200") OR
               (planType == "Plan200") OR (planType == "plan200"))
                SET userPlanType = true
                PT = plan200
                DISPLAY “You have selected: Plan 200”
                IF (CL3 == “ACL”)
                    RETURN 200000
                ELSE
                    RETURN 1000000
                ENDIF
        ELSE
            DISPLAY “Please enter an appropriate plan type”
        ENDIF
    ELSE
        IF ((age3>20 AND age3<41) AND (ageInDays2==false))
            IF ((planType == "Plan 150") OR (planType == "plan 150") OR
               (planType == "Plan150") OR (planType == "plan150"))
                SET userPlanType = true
                PT = plan150
```

```
                    DISPLAY “You have selected: Plan 150”
                    IF (CL3 == “ACL”)
                        RETURN 150000
                    ELSE
                        RETURN 750000
                    ENDIF
                ELSE
                    IF ((planType == "Plan 200") OR
                       (planType == "plan 200") OR
                       (planType == "Plan200") OR
                       (planType == "plan200"))
                        SET userPlanType = true
                        PT = plan200
                        DISPLAY “You have selected: Plan 200”
                        IF (CL3 == “ACL”)
                            RETURN 200000
                        ELSE
                            RETURN 1000000
                        ENDIF
                ELSE
                    DISPLAY “Enter an appropriate plan type”
                ENDIF
ELSE
        IF ((planType == "Plan 200") OR (planType == "plan 200") OR
           (planType == "Plan200") OR (planType == "plan200"))
            SET userPlanType = true
            PT = plan200
            DISPLAY “You have selected: Plan 200”
            IF (CL3 == “ACL”)
                RETURN 200000
            ELSE
                RETURN 1000000
```

```
                ENDIF
            ELSE
                DISPLAY “Please enter an appropriate plan type”
            ENDIF
        ENDIF
    ENDFOR
ENDFUNCTION

FUNCTION getUserInformation(name2,houseNumber2,street2,city2,state2,zipCode2,CN,HH)
    DECLARE s
    DECLARE healthHistory[100]
    DECLARE none[5]={"None"}
    DECLARE condition=false
    DECLARE haveHealthIssue[3]

    DISPLAY “User Details”
    DISPLAY “Enter Name:”
    READ name2
    DISPLAY “Enter House Number:”
    READ houseNumber2
    DISPLAY “Enter Street Address:”
    READ street2
    DISPLAY “Enter City:”
    READ city2
    DISPLAY “Enter State:”
    READ state2
    DISPLAY “Enter Zip Code:”
    READ zipCode2
    DISPLAY “Enter Contact Number:”
    READ CN
```

```
        DISPLAY “Do you have health issue: ('Yes','No')”
        FOR (s=0; condition != true; s++)
                READ haveHealthIssue
                IF (haveHealthIssue == “Yes”)
                        DISPLAY “State all your health issue that you are facing in one line”
                        READ HH
                        DISPLAY “Thank you for entering all the information needed”
                        SET condition = true
                ELSE
                        IF (haveHealthIssue == “No”)
                                HH = none
                                DISPLAY “Thank you for entering all the information needed”
                                SET condition = true
                        ELSE
                                DISPLAY “Wrong answer entered”
                                DISPLAY “Either ('Yes','No')”
                        ENDIF
                ENDIF
        ENDFOR
ENDFUNCTION

FUNCTION displayOverall(id1,name3, age4, CL4, PT2, houseNumber3, street3, city3, state3,
                        zipCode3, CN2, HH2, ageInDays3, amount1)
        DISPLAY “User Information:”
        DISPLAY “User ID: ”, id1
        DISPLAY “Name: ”, name3
        DISPLAY “Age: ”, age4
        IF (ageInDays3==true)
                DISPLAY “  Days Old”
        ELSE
                DISPLAY “ Years Old”
```

```
        DISPLAY “Claim Limit Type: ”, CL4
        DISPLAY “Type of plan: ”, PT2
        DISPLAY “Amount Subscribe: RM ”, amount1
        DISPLAY “Date Subscription: ”, GET SYSTEM DATE
        DISPLAY “Address: No. ”, houseNumber3, street3, city3, state3, zipCode3
        DISPLAY “Contact: ”, CN2
        DISPLAY “Health Issues: ”, HH2
ENDFUNCTION

FUNCTION saveToAFile(id2, name4, age5, CL5, PT3, houseNumber4, street4,  city4, state4,
                                zipCode4, CN3, HH3, ageInDays4, amount2)
        OPENFILE Insurance_Plans.txt FOR READ mode
        IF ( “Insurance_Plans.txt”  == NULL  )
                CLOSEFILE Insurance_Plans.txt
                OPENFILE Insurance_Plans.txt FOR WRITE mode
                WRITEFILE “ID: ”, id2
                WRITEFILE “Name: ”, name4
                WRITEFILE “Age: ”, age5
                IF (ageInDays4==true)
                        WRITEFILE “ Days Old”
                ELSE
                        WRITEFILE “ Years Old”
                ENDIF
                WRITEFILE “Type of Claim Limit: ”, CL5
                WRITEFILE “Type of Plan: ”, PT3
                WRITEFILE “Amount Balance: RM ”,  amount2
                WRITEFILE “Amount Claim: RM 0.00”
                WRITEFILE “Date Subscription: ”, GET SYSTEM DATE
                WRITEFILE “Address: No. ”, houseNumber4, street4, city4, state4, zipCode4
                WRITEFILE “Contact Number: ”, CN3
                WRITEFILE “Health History: ”,  HH3
```

```
            CLOSEFILE “Insurance_Plans.txt”
        ELSE
            CLOSEFILE Insurance_Plans.txt
            OPENFILE Insurance_Plans.txt FOR APPEND mode
            WRITEFILE “ID: ”, id2
            WRITEFILE “Name: ”, name4
            WRITEFILE “Age: ”, age5
            IF (ageInDays4==true)
                WRITEFILE “ Days Old”
            ELSE
                WRITEFILE “ Years Old”
            ENDIF
            WRITEFILE “Type of Claim Limit: ”, CL5
            WRITEFILE “Type of Plan: ”, PT3
            WRITEFILE “Amount Balance: RM ”,  amount2
            WRITEFILE “Amount Claim: RM 0.00”
            WRITEFILE “Date Subscription: ”, GET SYSTEM DATE
            WRITEFILE “Address: No. ”, houseNumber4, street4, city4, state4, zipCode4
            WRITEFILE “Contact Number: ”, CN3
            WRITEFILE “Health History: ”, HH3
            CLOSEFILE “Insurance_Plans.txt”
        ENDIF
ENDFUNCTION

FUNCTION getUserID()
        DECLARE j, i=0
        OPENFILE “Unique_ID.txt” FOR READ mode
        IF (“Uniques_ID.txt” == NULL)
            CLOSEFILE “Uniques_ID.txt”
            OPENFILE “Unique_ID.txt” FOR WRITE mode
```

```
        INCREAMENT the value of i by 1
        WRITEFILE i
        CLOSEFILE “Uniques_ID.txt”
        RETURN i
    ELSE
        CLOSEFILE “Uniques_ID.txt”
        OPENFILE “Unique_ID.txt” FOR READ mode
        READFILE j
        CLOSEFILE “Uniques_ID.txt”
        OPENFILE “Unique_ID.txt” FOR WRITE mode
        INCREAMENT the value of j by 1
        WRITEFILE j
        CLOSEFILE “Uniques_ID.txt”
        RETURN j
    ENDIF
ENDFUNCTION
```

```
FUNCTION claimInsurance()
        DECLARE userAnswer[30], method[10], TypeOfPlan[8], id[8], name[25]
        DECLARE ageType[12], age[3]
        DECLARE location, getRoomCharge ,getIcuCharge
        DECLARE getHospitalSuppliesAndService, getSurgicalAndOtherFee
        DECLARE amountBalance, amountClaim, total

        CALL GetNameOrId(userAnswer, method)
        location = CALL compareAndGiveLocation(userAnswer, method)
        CALL checkUserAge(location)
        CALL storeUserNameAndId(id, name, location)
        amountBalance = CALL getAmountBalance(location)
        amountClaim = CALL getAmountClaim(location)
        CALL getPlanType2(TypeOfPlan, location)
        CALL display(TypeOfPlan)
        getRoomCharge = CALL getRoomCharges()
        getIcuCharge = CALL getIcuCharges()
        getHospitalSuppliesAndService = CALL getHospitalSuppliesAndServices()
        getSurgicalAndOtherFee = CALL getSurgicalAndOtherFees()
        total = CALL totalRequestClaim(TypeOfPlan, amountBalance, getRoomCharge,
                                       getIcuCharge, getHospitalSuppliesAndService,
                                       getSurgicalAndOtherFee)
        amountBalance = CALL changeAmountBalance(amountBalance, total)
        amountClaim = CALL changeAmountClaim(amountClaim, total)
        CALL displayOverallClaim(id, name, total, amountBalance)
        CALL recordClaim(id, name, total, amountBalance)
        CALL updateInsurancePlansFile(location, amountClaim, amountBalance)
ENDFUNCTION

FUNCTION getNameOrId(userAnswer1, method1)
        DECLARE k, n, num
```

```
    DECLARE c[]="usingName", d[]="usingId", userName[25], userID[8]
    DECLARE answer=false
    DISPLAY “Enter 1 to search using name”
    DISPLAY “Enter 2 to search using user ID”
    DISPLAY “Your Selection:”
    FOR (k=0; answer != true; k++)
        READ num
        IF (num==1)
            DISPLAY “Enter Name”
            READ userName
            method1 = c
            SET answer=true
            STRING COPY (userAnswer1,"Name: ")
            STRING CONCATENATE(userAnswer1, userName)
        ELSE
            IF (num==2)
                DISPLAY “Enter ID”
                READ userID
                method1 = d
                SET answer=true
                STRING COPY(userAnswer1,"ID: ")
                STRING CONCATENATE (userAnswer1, userID)
            ELSE
                DISPLAY “Wrong number entered, enter again”
            ENDIF
        ENDIF
    ENDFOR
ENDFUNCTION
```

```
FUNCTION compareAndGiveLocation(userAnswer2, method2)
        DECLARE i=0, count=0
        DECLARE x[12001][101]
        DECLARE nameFound=false, idFound=false
        OPENFILE "Insurance_Plans.txt" FOR READ mode
        IF ("Insurance_Plans.txt" == NULL)
                DISPLAY “Cannot open file”
        ELSE
                WHILE (i<12001)
                        READFILE x[i]
                        INCREAMENT the value of i by 1
                ENDWHILE
                IF (method2 == “usingName”)
                        WHILE (nameFound!=true)
                                IF (userAnswer2 == x[count])
                                        DISPLAY “Name Found”
                                        SET nameFound=true
                                        RETURN count-1
                                ELSE
                                        IF (STRING LENGTH(x[count])==0)
                                                DISPLAY “Name not found”
                                                SET nameFound=true
                                                exit(0)
                                ENDIF
                                INCREAMENT the value of count by 1
                        ENDWHILE
                ELSE
                        IF (method2 == “usingId”)
                                WHILE (idFound!=true)
                                        IF (userAnswer2 == x[count])
```

```
                                    DISPLAY “Id Found”
                                    SET idFound=true
                                    RETURN count
                                ELSE
                                    IF (STRING LENGTH(x[count])==0)
                                        DISPLAY “Id not found”
                                        SET idFound=true
                                        exit(0)
                                ENDIF
                                INCREAMENT the value of count by 1
                            ENDWHILE
                        ELSE
                            DISPLAY “Error occurred”
                        ENDIF
                ENDIF
        ENDIF
        CLOSEFILE "Insurance_Plans.txt"
ENDFUNCTION

FUNCTION checkUserAge (location0)
        DECLARE getAgeType1[15], age1[3], w[12001][101], subscribeDate[5]
        DECLARE count, h=0, j, n, k=0, g, d, f, p=0, a=0
        DECALRE userFound=false
        DECLARE year
        DECLARE difference, num1, num2, num3
        OPENFILE “Insurance_Plans.txt” FOR READ mode
        GET SYSTEM current year and store the data in the variable year
        IF ( “Insurance_Plans.txt” == NULL)
                DISPLAY “Cannot open file”
        ELSE
```

```
WHILE (h<12001)
        READFILE w[h]
        INCREAMENT the value of h by 1
ENDWHILE
n = STRING LENGTH(w[location0+2])
IF ( w[location0+3] == "Type of Claim Limit: ACL")
        FOR (j=n-10; j<n+1; j++)
                getAgeType1[a]=w[location0+2][j]
                INCREAMENT the value of a by 1
        ENDFOR
        IF (getAgeType1 == "Years Old")
                g=STRING LENGTH(w[location0+7])
                FOR (j=5;j<n-10;j++)
                        age1[p]=w[location0+2][j]
                        INCREAMENT the value of p by 1
                ENDFOR
                FOR (f=24; f<g+1; f++)
                        subscribeDate[d]=w[location0+7][f]
                        INCREAMENT the value of d by 1
                ENDFOR
                COMPUTE num1=ASCII to integer(subscribeDate)
                COMPUTE num2=ASCII to integer(year)
                COMPUTE difference=num2-num1
                COMPUTE num3=ASCII to integer(age1)
                IF (num3+difference>60)
                        DISPLAY “Sorry, you can claim until at age 60 years old for
                                ACL”
                        exit(0)
                ENDIF
        ENDIF
ENDIF
```

```
        ENDIF
        CLOSEFILE “Insurance_Plans.txt”
ENDFUNCTION

FUNCTION storeUserNameAndId(id1, name1, location1)
        DECLARE i=0, s, w, d, e, q=0
        DECLARE y[12001][101]
        OPENFILE "Insurance_Plans.txt" FOR READ mode
        IF ( "Insurance_Plans.txt" == NULL )
                DISPLAY “Cannot open file”
        ELSE
                WHILE (i<12001)
                        READFILE y[i]
                        INCREAMENT the value of i by 1
                ENDWHILE
                d=STRING LENGTH(y[location1])
                e=STRING LENGTH(y[location1+1])
                FOR (s=4;s<d;s++)
                        id1[q]=y[location1][s]
                        INCREAMENT the value of q by 1
                ENDFOR
                SET value of q to 0
                FOR (w=6;w<e;w++)
                        name1[q]=y[location1+1][w]
                        INCREAMENT the value of q by 1
                ENDFOR
        ENDIF
        CLOSEFILE "Insurance_Plans.txt"
ENDFUNCTION
```

```
FUNCTION getAmountBalance(location2)
        DECLARE m=0, l, n, a=0
        DECLARE str1[1][35], z[12001][101], amountBalance1[8]
        OPENFILE "Insurance_Plans.txt" FOR READ mode
        IF ("Insurance_Plans.txt" == NULL)
                DISPLAY “Cannot open file”
        ELSE
                WHILE (m<12001)
                        READFILE z[m]
                        INCREAMENT the value of m by 1
                ENDWHILE
                n=STRING LENGTH(z[location2+5])
                FOR (l=19;l<n;l++)
                        amountBalance1[a]=z[location2+5][l]
                        INCREAMENT the value of a by 1
                ENDFOR
        ENDIF
        CLOSEFILE"Insurance_Plans.txt"
ENDFUCNTION

FUNCTION getAmountClaim(location3)
        DECLARE i=0, g, h, a
        DECLARE z1[12001][101], amountClaim1[8]
        OPENFILE "Insurance_Plans.txt" FOR READ mode
        IF ("Insurance_Plans.txt" == NULL)
                DISPLAY “Cannot open file”
        ELSE
                WHILE (i<12001)
                        READFILE z1[i]
                        INCREAMENT the value of i by 1
```

```
            ENDWHILE
            h=STRING LENGTH(z1[location3+6])
            FOR (g=17;g<h;g++)
                amountClaim1[a]=z1[location3+6][g]
                INCREAMENT the value of a by 1
            ENDFOR
        ENDIF
        CLOSEFILE “Insurance_Plans.txt”
        RETURN ASCII to float (amountClaim1)
ENDFUNCTION

FUNCTION getPlanType2(PT, location4)
        DECLARE i=0, g, h, a=0
         DECLARE z2[12001][101]
        OPENFILE “Insurance_Plans.txt” FOR READ mode
        IF (“Insurance_Plans.txt” == NULL)
            DISPLAY “Cannot open file”
        ELSE
            WHILE (i<12001)
                READFILE z2[i]
                INCREAMENT the value of i by 1
            ENDWHILE
            h=STRING LENGTH(z2[location4+4])
            FOR (g=14;g<h-1;g++)
                *(PT+a)=z2[location4+4][g]
                INCREAMENT the value of a by 1
            ENDFOR
        ENDIF
        CLOSEFILE “Insurance_Plans.txt”
ENDFUNCTION
```

```
FUNCTION display(PT3)
        IF (PT3 == “Plan120”)
                DISPLAY Benefits for Plan120
        ELSE
                IF (PT3 == “Plan150”)
                        DISPLAY Benefits for Plan150
                ELSE
                        IF (PT3 == “Plan200”)
                                DISPLAY Benefits for Plan200
                        ELSE
                                DISPLAY “Error Occurred”
                        ENDIF
                ENDIF
        ENDIF
ENDFUNCTION

FUNCTION getRoomCharges()
        DECLARE num1
        DISPLAY “Please enter the number of days that you stay in the normal ward”
        READ num1
        RETURN num1
ENDFUNCTION

FUNCTION getIcuCharges()
        DECLARE num2
        DISPLAY “Please enter the number of days that you stay in the Intensive Care Unit”
        READ num2
        RETURN num2
ENDFUNCTION
```

```
FUNCTION getHospitalSuppliesAndServices()
        DECLARE num3
        DISPLAY “Please enter the total hospital supplies and services fees”
        READ num3
        RETURN num3
ENDFUNCTION

FUNCTION getSurgicalAndOtherFees()
        DECLARE num4
        DISPLAY “Please enter the total surgical and other fees”
        READ num4
        RETURN num4
ENDFUNCTION

FUNCTION totalRequestClaim(PT4, amountBalance1, getRoomCharges1, getIcuCharges1,
                           getHospitalSuppliesAndServices1, getSurgicalAndOtherFee1)
        DECLARE total
        DECLARE answer2=false
        DECLARE answer[2]
        DECLARE d
        IF (PT4 == “Plan120”)
                COMPUTE total = ((getRoomCharges1*120) + (getIcuCharges1*250) +
                                 getHospitalSuppliesAndServices1 +
                                 getSurgicalAndOtherFee1)
        ELSE
                IF (PT4 == “Plan150”)
                        COMPUTE total=((getRoomCharges1*150) + (getIcuCharges1*400) +
                                       getHospitalSuppliesAndServices1 +
                                       getSurgicalAndOtherFee1)
        ELSE
                IF (PT4 == “Plan200”)
```

```
                COMPUTE total=((getRoomCharges1*200) + (getIcuCharges1*700) +
                                getHospitalSuppliesAndServices1 +
                                getSurgicalAndOtherFee1)
ELSE
        DISPLAY “An error occurred”
ENDIF
IF (total > amountBalance1)
        COMPUTE total=total-amountBalance1
        DISPLAY “Your limit have been reach, you can claim RM”, amountBalance1
        DISPLAY “The rest of the amount, RM ”, total “have to be borne by you”
        DISPLAY “Would you which to continue? ('Y','N')”
        FOR (d=0;answer2!=true;d++)
                READ answer
                IF (answer == “Y”)
                        DISPLAY “Your claim is RM ” amountBalannce1
                        SET answer2 = true
                        COMPUTE total=amountBalance1
                        RETURN total
                ELSE
                        IF (answer == “N”)
                                DISPLAY “Please contact our customer service to
                                        subscribe to the premium insurance to be
                                        insured more”
                                DISPLAY “Thank You for our page”
                                SET answer2 = true
                                exit(0)
                ELSE
                        DISPLAY “Please enter the appropriate response”
                ENDIF
        ENDFOR
ELSE
        DISPLAY “Your claim have been received”
```

```
            RETURN total
        ENDIF
ENDFUNCTION

FUNCTION changeAmountBalance(amountBalance2, total1)
        COMPUTE amountBalance2=amountBalance2-total1
        RETURN amountBalance2
ENDFUNCTION

FUNCTION changeAmountClaim(amountClaim1, total2)
        COMPUTE amountClaim1=amountClaim1+total2
        RETURN amountClaim1
ENDFUNCTION

FUNCTION displayOverallClaim(id2, name2, total3, amountBalance2)
        DISPLAY “Insurance Claim: ”
        DISPLAY “ID: ”, id2
        DISPLAY “Name: ”, name2
        DISPLAY “Claim Date: ”, GET SYSTEM DATE
        DISPLAY “Total Amount Claim: RM ”, total3
        DISPLAY “Balance Claimable Amount: RM ”, amountBalance2
ENDFUNCTION

FUNCTION recordClaim(id3, name3, total4, amountBalance3)
        OPENFILE “Insurance_Claims.txt” FOR READ mode
        IF (“Insurance_Claims.txt” == NULL)
            CLOSEFILE “Insurance_Claims.txt”
            OPENFILE “Insurance_Claims.txt” FOR WRITE mode
            WRITEFILE “Insurance Claim:”
            WRITELINE “ID: ”, id3
```

```
            WRITEFILE “Name: ”, name3
            WRITEFILE “Claim Date: ”, GET SYSTEM DATE
            WRITEFILE “Total Amount Claim: RM ” total4
            WRITEFILE “Balance Claimable Amount: RM ” amountBalance3
            CLOSEFILE “Insurance_Claims.txt”
        ELSE
            CLOSEFILE “Insurance_Claims.txt”
            OPENFILE “Insurance_Claims.txt” FOR APPEND mode
            WRITEFILE “Insurance Claim:”
            WRITELINE “ID: ”, id3
            WRITEFILE “Name: ”, name3
            WRITEFILE “Claim Date: ”, GET SYSTEM DATE
            WRITEFILE “Total Amount Claim: RM ” total4
            WRITEFILE “Balance Claimable Amount: RM ” amountBalance3
            CLOSEFILE “Insurance_Claims.txt”
        ENDIF
ENDFUNCTION

FUNCTION updateInsurancePlansFile(location5, amountClaim2, amountBalance4)
        DECLARE f=0, l=0, b, v
        DECLARE z3[12001][101], amount1[40], amount2[10], amount3[40], amount4[10]
        OPENFILE "Insurance_Plans.txt" FOR READ mode
        IF ("Insurance_Plans.txt" == NULL)
            DISPLAY “File is not available”
        ELSE
            WHILE (f<12001)
                READFILE z3[f]
                INCREAMENT the value of f by 1
            ENDWHILE
            STRING PRINT FORMAT(amount2, "%.2f", amountBalance4)
```

```
            STRING COPY(amount1,"Amount Balance: RM ")
            STRING CONCATENATE (amount1, amount2)

            STRING PRINT FORMAT(amount4, "%.2f", amountClaim2)
            STRING COPY(amount3,"Amount Claim: RM ")
            STRING CONCATENATE (amount3, amount4)
            FOR (b=0;b<STRING LENGTH(amount1)+1;b++)
                  z3[location5+5][b]=amount1[b]
            ENDFOR
            FOR (v=0;v<STRING LENGTH(amount3)+1;v++)
                  z3[location5+6][v]=amount3[v]
            ENDFOR
            CLOSEFILE "Insurance_Plans.txt”
            OPENFILE "Insurance_Plans.txt” FOR WRITE mode
            WHILE (STRING LENGTH(z3[l])!=0)
                  fprintf(fp10,"%s",z3[l])
                  INCREAMENT the value of l by 1
            ENDWHILE
            CLOSEFILE "Insurance_Plans.txt”
      ENDIF
ENDFUCNTION
```

```
FUNCTION accountInformation()
        DECLARE num
        num= CALL displayOptions()
        IF (num==1)
                CALL displayTotalLCLAmountClaim()
        ELSE
                CALL totalOfExhaustedACL()
        ENDIF
ENDFUNCTION

FUNCTION displayOptions()
        DECLARE selection2
        DECLARE correctSelection=false
        DECALRE k
        DISPLAY “Enter 1 for total amount claimed by Lifetime Claim Limit”
        DISPLAY “Enter 2 for total Annual Claim Limit who exhausted all their balance”
        FOR (k=0; correctSelection != true; k++)
                READ selection2
                IF (selection2 == 1)
                        SET correctSelection = true
                ELSE
                        IF (selection2 == 2)
                                SET correctSelection = true
                        ELSE
                                DISPLAY “Enter appropriate response”
                        ENDIF
                ENDIF
        ENDFOR
        RETURN selection2
ENDFUNCTION
```

```
FUNCTION displayTotalLCLAmountClaim
        DECLARE f [12001][101]
        DECLARE h, count=0, j, l=0
        DECLARE CL=false
        DECLARE amountClaim[11]
        DECLARE totalClaim, amountClaim2
        OPEN “Insurance_Plans.txt” FOR READ
        WHILE (h<12001)
                READFILE f[h]
                INCREAMENT the value of h by 1
        ENDWHILE
        WHILE (CL != true)
                IF (f[count] == “Type of Claim Limit: LCL”)
                        FOR (j=17;j<STRING LENGTH(f[count+3]);j++)
                                amountClaim[l] = f[count+3][j]
                                INCREAMENT the value of l by 1
                        COMPUTE amountClaim2=ASCII to float(amountClaim)
                        COMPUTE totalClaim=totalClaim+amountClaim2
                ELSE
                        IF (STRING LENGTH(f[count]) == 0)
                                SET CL = true
                        ENDIF
                ENDIF
                INCREAMENT the value of count by 1
        ENDWHILE
        DISPLAY “Total Claim by LCL subscriber is ”, totalClaim
        CLOSE “Insurance_Plans.txt”
ENDFUNCTION
```

```
FUNCTION totalOfExhaustedACL()
        DECLARE g[12001][101]
        DECLARE j, count=0, EC=0
        DECLARE EC1 = false
        OPEN “Insurance_Plans.txt” FOR READ mode
        WHILE (h<12001)
                READFILE g[h]
                INCREAMENT the value of h by 1
        ENDWHILE
        WHILE (EC1 != true)
                IF (g[count] == “Type of Claim Limit: ACL”)
                        IF (g[count+2] == “Amount Balance: RM 0.00”)
                                INCREAMENT the value of EC by 1
                        ENDIF
                ELSE
                        IF (STRING LENGTH(g[count]) == 0)
                                SET EC1 = true
                        ENDIF
                ENDIF
                INCREAMENT the value of count by 1
        ENDWHILE
        DISPLAY “Total ACL subscriber that have exhausted their amount is”  EC
        CLOSE “Insurance_Plans.txt”
ENDFUNCTION
```

```
FUNCTION search()
        DECLARE userAnswer1[30], userAnswer2[30], userAnswer3[30], method[30]
        DECLARE num, age
        DECLARE ageInDays[20], ageInYears[20]

        getNum(num, age, ageInDays, ageInYears, userAnswer1, userAnswer2, userAnswer3,
                method)
        openFileCompareDisplay(userAnswer1, userAnswer2, userAnswer3, method)
        RETURN 0
ENDFUNCTION

FUNCTION getNum(num, age, ageInDays, ageInYears, userAnswer1, userAnswer2,
                        userAnswer3, method1)
        DECLARE k,m,b,n
        DECLARE c[]="usingName", d[]="usingId", userName[25], userID[8]
        DECLARE answer=false, name=true
        DISPLAY “Enter 1 to search using name”
        DISPLAY “Enter 2 to search using user ID”
        DISPLAY “Enter 3 to search using Plan, Claim Limit Type and Age”
        FOR (k=0;answer!=true;k++)
                READ num
                IF (num==1)
                        DISPLAY “Enter Name”
                        READ userName
                        STRING COPY(method1,c)
                        SET answer=true
                        STRING COPY (userAnswer1,"Name: ")
                        STRING CONCATENATE(userAnswer1, userName)
                ELSE
                        IF (num==2)
                                DISPLAY” Enter ID”
```

```
                    READ userID
                    STRING COPY(method1,d)
                    SET answer=true
                    STRING COPY (userAnswer1,"ID: ")
                    STRING CONCATENATE(userAnswer1, userID)
                ELSE
                    IF (num==3)
                        CALL get3Attribute(age, ageInDays, ageInYears,
                                           userAnswer1, userAnswer2,
                                           userAnswer3, method1)
                        SET answer=true
                    ELSE
                        DISPLAY “Wrong number entered”
                    ENDIF
                ENDIF
            ENDIF
        ENDFOR
ENFUNCTION

FUNCTION get3Attribute(age2, ageInDays2, ageInYears2, userAnswer4, userAnswer5,
                       userAnswer6, method2)
    DECLARE n, ageInDays3
    DECLARE e[]="using3AttributeWithDaysOld", f[]="using3AttributeWithYearsOld"
    DECLARE plan120[8]="Plan120", plan150[8]="Plan150", plan200[8]="Plan200"
    DECLARE CLAnswer=false, PTAnswer=false, ageAnswer=false,
    DECLARE ageInDaysAnswer=false, plan[10], CL[5]

    DISPLAY “If the subscriber is less than 1 years old, please enter '-1'”
    DISPLAY “Enter age:”
    WHILE ( ageAnswer != true )
        READ age2
```

```
IF (age2==-1)
        SET ageAnswer=true
        STRING COPY(method2,e)
        DISPLAY “Enter the age in days”
        WHILE (ageInDaysAnswer!=true)
                READ ageInDays2
                COMPUTE ageInDays3=ASCII to integer(ageInDays2)
                IF (ageInDays3>0 AND ageInDays3<365)
                        SET ageInDaysAnswer=true
                        STRING PRINT FORMAT(ageInDays2,"%d",ageInDays3)
                ELSE
                        DISPLAY “Please enter an appropriate age in days”
                ENDIF
        ENDWHILE
        STRING COPY(userAnswer4,"Age: ")
        STRING CONCATENATE(ageInDays2," Days Old\n")
        STRING CONCATENATE(userAnswer4,ageInDays2)
        DISPLAY “ACL = Annual Claim Limit”
        DISPLAY “LCL = Lifetime Claim Limit”
        DISPLAY “Enter Claim Limit Type: ”
        WHILE (CLAnswer!=true)
                READ CL
                IF (CL == “ACL”)
                        SET CLAnswer=true
                ELSE
                        IF (CL == “LCL”)
                                SET CLAnswer=true
                        ELSE
                                DISPLAY “Enter an appropriate claim limit”
                        ENDIF
```

```
        ENDIF
ENDWHILE
STRING COPY (userAnswer5,"Type of Claim Limit: ")
STRING CONCATENATE(userAnswer5,CL)
DISPLAY “Enter Plan Type: ”
WHILE (PTAnswer!=true)
        READ plan
        IF (plan == “Plan120”) OR (plan == “Plan 120”) OR (plan ==
           “plan120”) OR (plan == “plan 120”)
                STRING COPY(plan,plan120)
                SET PTAnswer=true
        ELSE
                IF (plan == “Plan150”) OR (plan == “Plan 150”) OR
                   (plan == “plan150”) OR (plan == “plan 150”)
                        STRING COPY(plan,plan150)
                        SET PTAnswer=true
                ELSE
                        IF(plan == “Plan200”) OR
                           (plan == “Plan 200”) OR
                           (plan == “plan200”) OR
                           (plan == “plan 200”)
                                STRING COPY(plan,plan200)
                                SET PTAnswer=true
                        ELSE
                                DISPLAY “Enter an appropriate plan”
                                DISPLAY “Either Plan 120, Plan 150 or
                                         Plan 200”
                        ENDIF
                ENDIF
        ENDIF
ENDWHILE
STRING COPY(userAnswer6,"Type of Plan: ")
```

```
        STRING CONCATENATE(plan,"\n")
        STRING CONCATENATE(userAnswer6,plan)
ELSE
        IF (age2 >0)
                SET ageAnswer=true
                STRING COPY(method2,f)
                STRING PRINT FORMAT(ageInYears2,"%d",age2)
                STRING COPY(userAnswer4,"Age: ")
                STRING CONCATENATE(ageInYears2," Years Old\n")
                STRING CONCATENATE(userAnswer4, ageInYears2)
                DISPLAY “ACL = Annual Claim Limit”
                DISPLAY “LCL = Lifetime Claim Limit”
                DISPLAY “Enter Claim Limit Type: ”
                WHILE (CLAnswer!=true)
                        READ CL
                        IF (CL == “ACL”)
                                SET CLAnswer=true
                        ELSE
                                IF (CL == “LCL”)
                                        SET CLAnswer=true
                                ELSE
                                        DISPLAY “Enter an appropriate claim
                                                limit”
                                ENDIF
                        ENDIF
                ENDWHILE
                STRING COPY(userAnswer5,"Type of Claim Limit: ")
                STRING CONCATENATE(userAnswer5, CL)
                DISPLAY “Enter Plan Type: ”
                WHILE (PTAnswer!=true)
                        READ plan
```

```
                    IF (plan == “Plan120”) OR (plan == “Plan 120”) OR
                      (plan == “plan120”) OR (plan == “plan 120”)
                        STRING COPY(plan,plan120)
                        SET PTAnswer=true
                    ELSE
                        IF (plan == “Plan150”) OR
                          (plan == “Plan 150”) OR
                          (plan == “plan150”) OR
                          (plan == “plan 150”)
                            STRING COPY(plan,plan150)
                            SET PTAnswer=true
                        ELSE
                            IF (plan == “Plan200”) OR
                              (plan == “Plan 200”) OR
                              (plan == “plan200”) OR
                              (plan == “plan 200”)
                                STRING COPY(plan,plan200)
                                SET PTAnswer=true
                            ELSE
                                DISPLAY “Enter an appropriate
                                        plan”
                                DISPLAY “Either Plan 120, Plan
                                        150 or Plan 200”
                            ENDIF
                        ENDIF
                    ENDIF
                ENDWHILE
                STRING COPY (userAnswer6,"Type of Plan: ")
                STRING CONCATENATE(plan,"\n")
                STRING CONCATENATE(userAnswer6, plan)
            ELSE
                DISPLAY “Please enter an appropriate age”
            ENDIF
        ENDIF
```

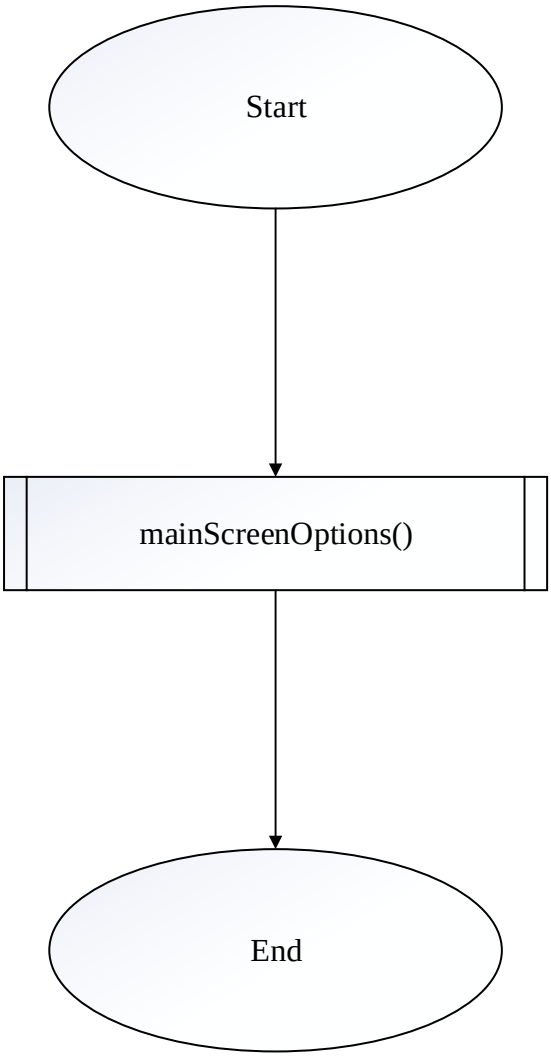
```
        ENDWHILE
ENDFUNTION

FUNCTION openFileCompareDisplay(userAnswer7, userAnswer8, userAnswer9, method3)

        DECLARE  i=0, l, count=0
        DECLARE x[12001][101]
        DECLARE nameFound=false, idFound=false, dataFound=false, display1=false
        OPENFILE "Insurance_Plans.txt" FOR READ mode
        IF ("Insurance_Plans.txt" == NULL)
                DISPLAY “Cannot open file”
        ELSE
                WHILE (i<12001)
                        READFILE x[i]
                        INCREAMENT the value of i by 1
                ENDWHILE
                IF (method3 == “usingName”)
                        WHILE (nameFound!=true)
                                IF (userAnswer7 == x[count])
                                        DISPLAY “Name Found”
                                        SET nameFound=true
                                        SET display1=true
                                ELSE
                                        IF (STRING LENGTH(x[count])==0)
                                                DISPLAY “Name not found”
                                                SET nameFound=true
                                        ENDIF
                                ENDIF
                                INCREAMENT the value of count by 1
                        ENDWHILE
                        IF (display1==true)
```

```
            FOR (l=count-2;l<count+9;l++)
                DISPLAY x[l]
            ENDFOR
        ENDIF
ELSE
        IF (method3 == “usingId”)
            WHILE (idFound!=true)
                IF (userAnswer7 == x[count])
                    DISPLAY “ID Found”
                    SET idFound=true
                    SET display1=true
                ELSE
                    IF (STRING LENGTH(x[count])==0)
                        DISPLAY “ID not found”
                        SET idFound=true
                    ENDIF
                ENDIF
                INCREAMENT the value of count by 1
            ENDWHILE
            IF (display1==true)
                FOR (l=count-1;l<count+10;l++)
                    DISPLAY x[l]
                ENDFOR
            ENDIF
        ELSE
             IF (method3 == “using3AttributeWithDaysOld”) OR
               (method3 ==  “using3AttributeWithYearsOld”)
                WHILE (dataFound!=true)
                    IF (userAnswer7 == x[count]) AND
                        (userAnswer8 == x[count+1]) AND
                        (userAnswer9 == x[count+2])
```

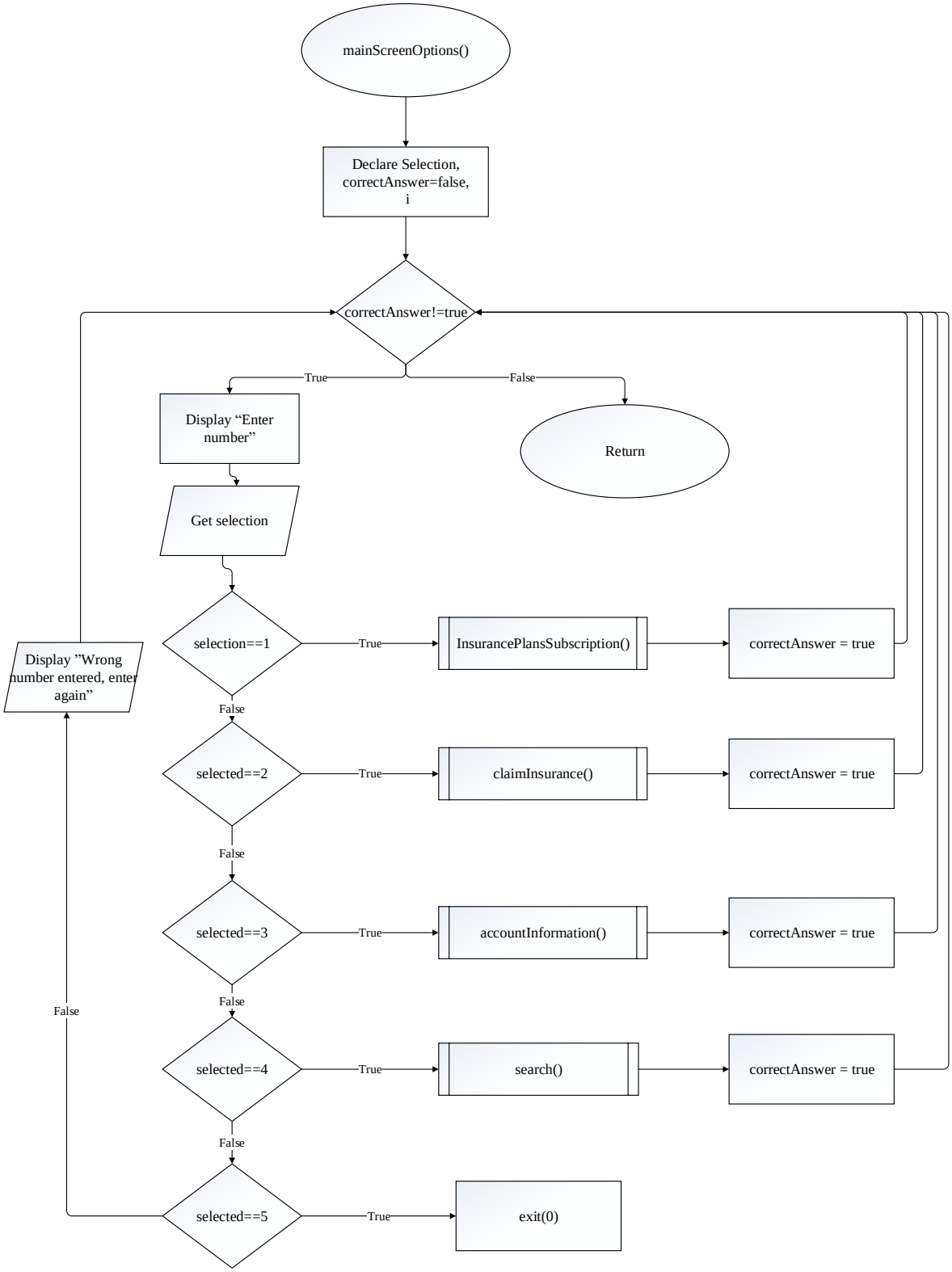
```
                                        DISPLAY “Data Found”
                                        SET display1=true
                                        IF (display1==true)
                                            FOR (l=count-3;l<count+8;l++)
                                                DISPLAY x[l]
                                            ENDFOR
                                        ENDIF
                                    ELSE
                                        IF (STRING LENGTH(x[count])==0)
                                            IF(display1 == false)
                                                DISPLAY “Data not
                                                        found”
                                            ENDIF
                                            SET dataFound=true
                                        ENDIF
                                    INCREAMENT the value of count by 1
                                ENDWHILE
                            ELSE
                                DISPLAY “Error Occurred”
                            ENDIF
                        ENDIF
                    ENDIF
                ENDIF
                CLOSEFILE "Insurance_Plans.txt"
ENDFUNCTION
```

3.2 Flowchart

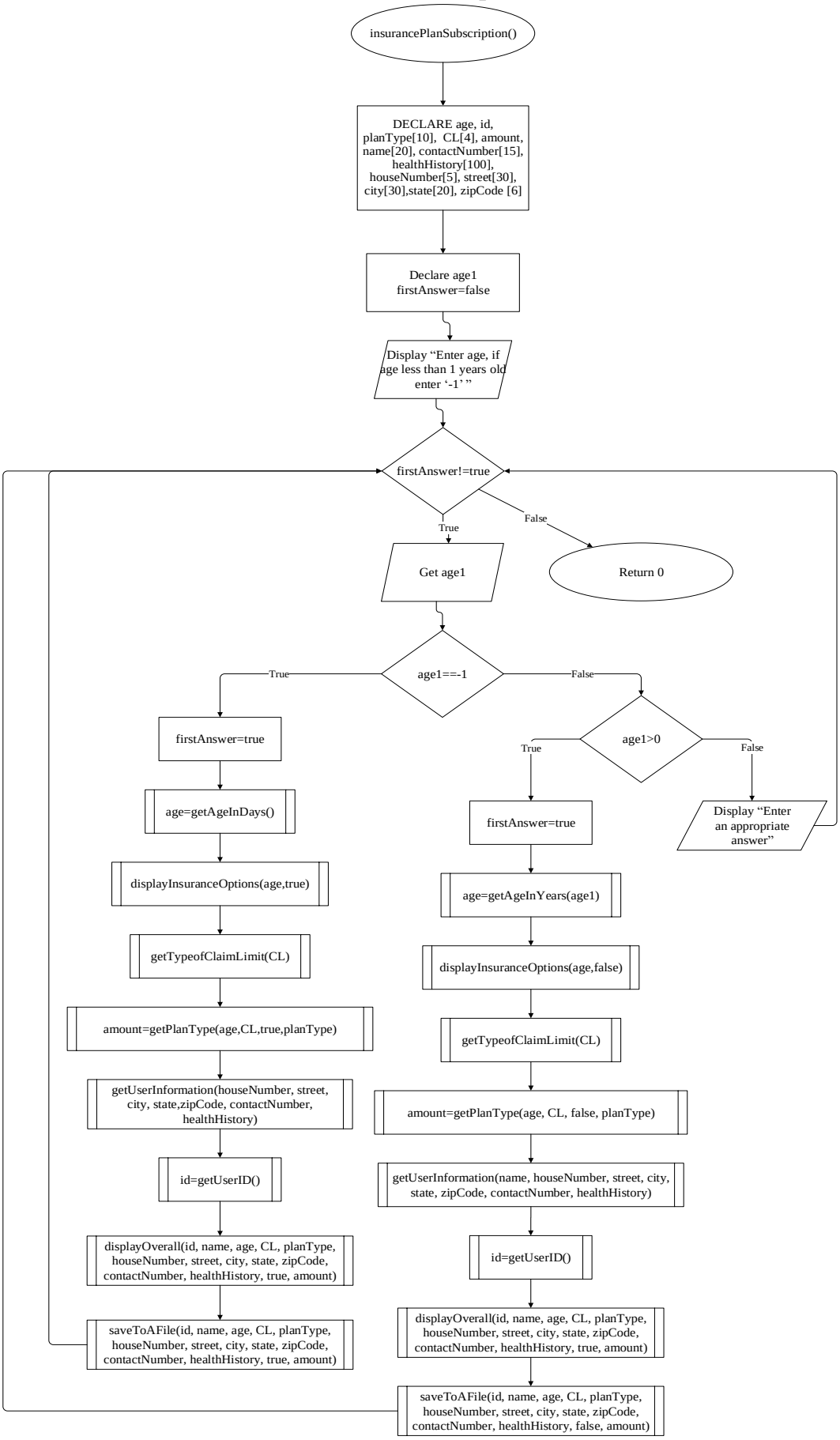
Flowchart for main program



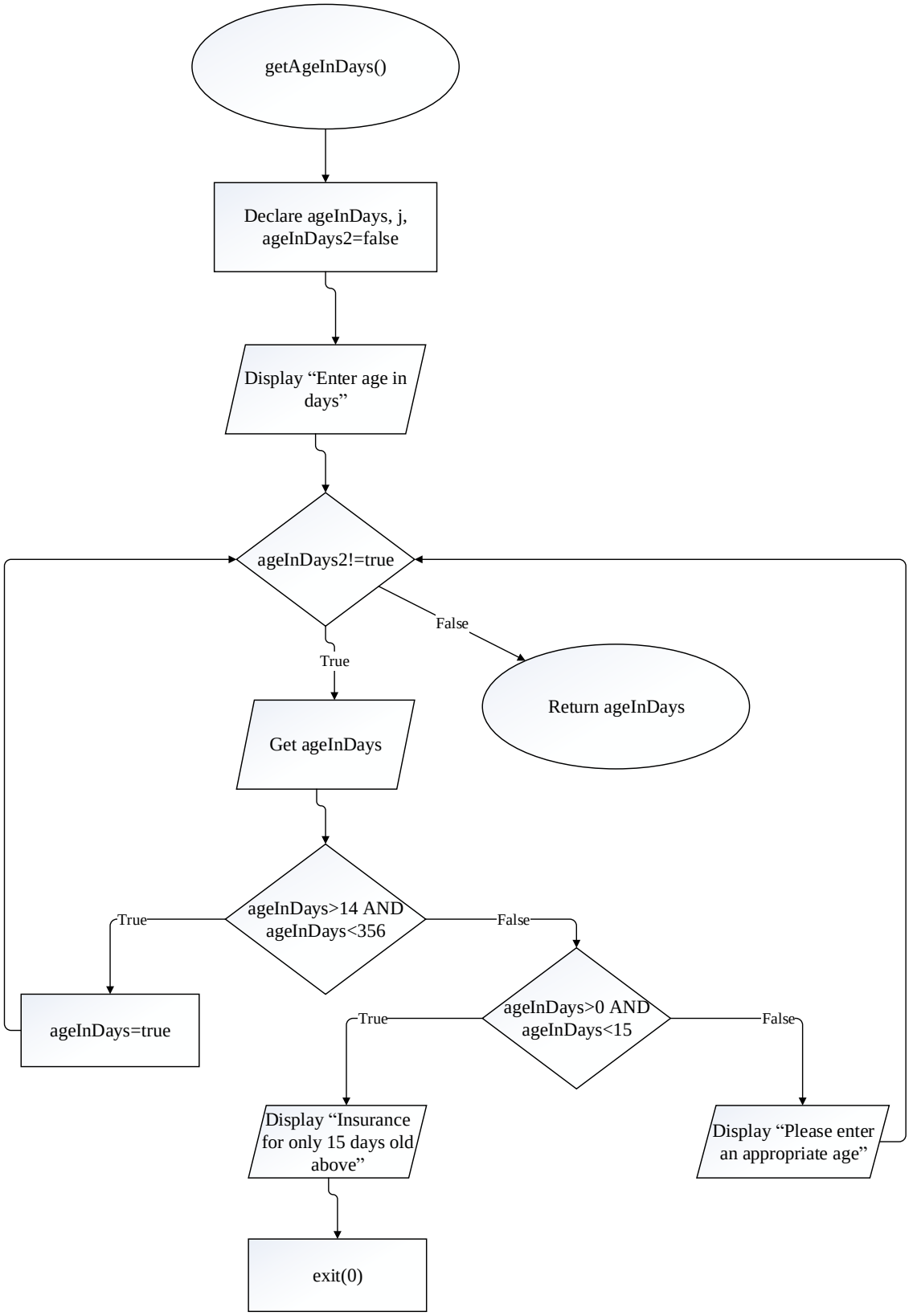
Flowchart for Function mainScreenOptions



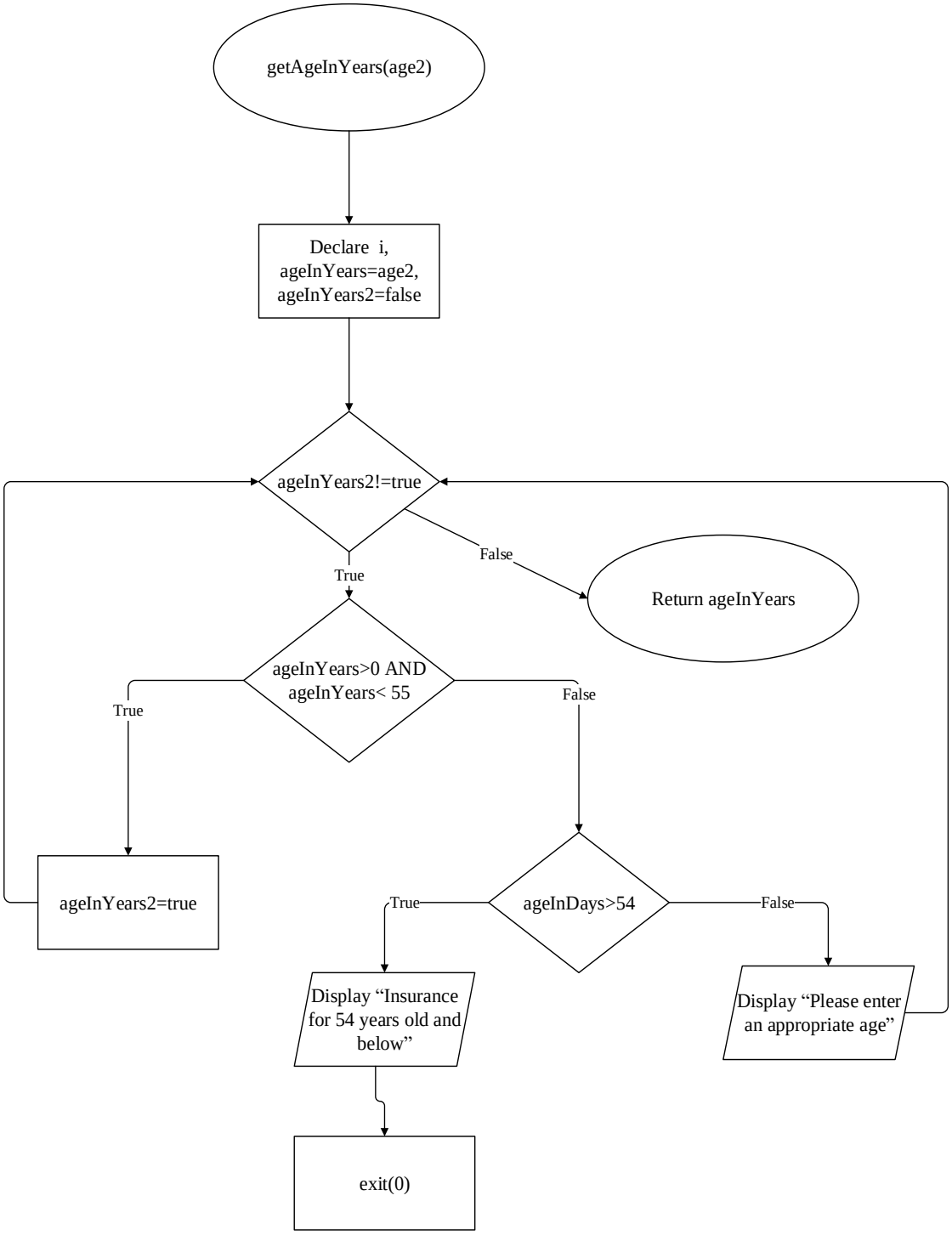
Flowchart for Function insurancePlanSubscription



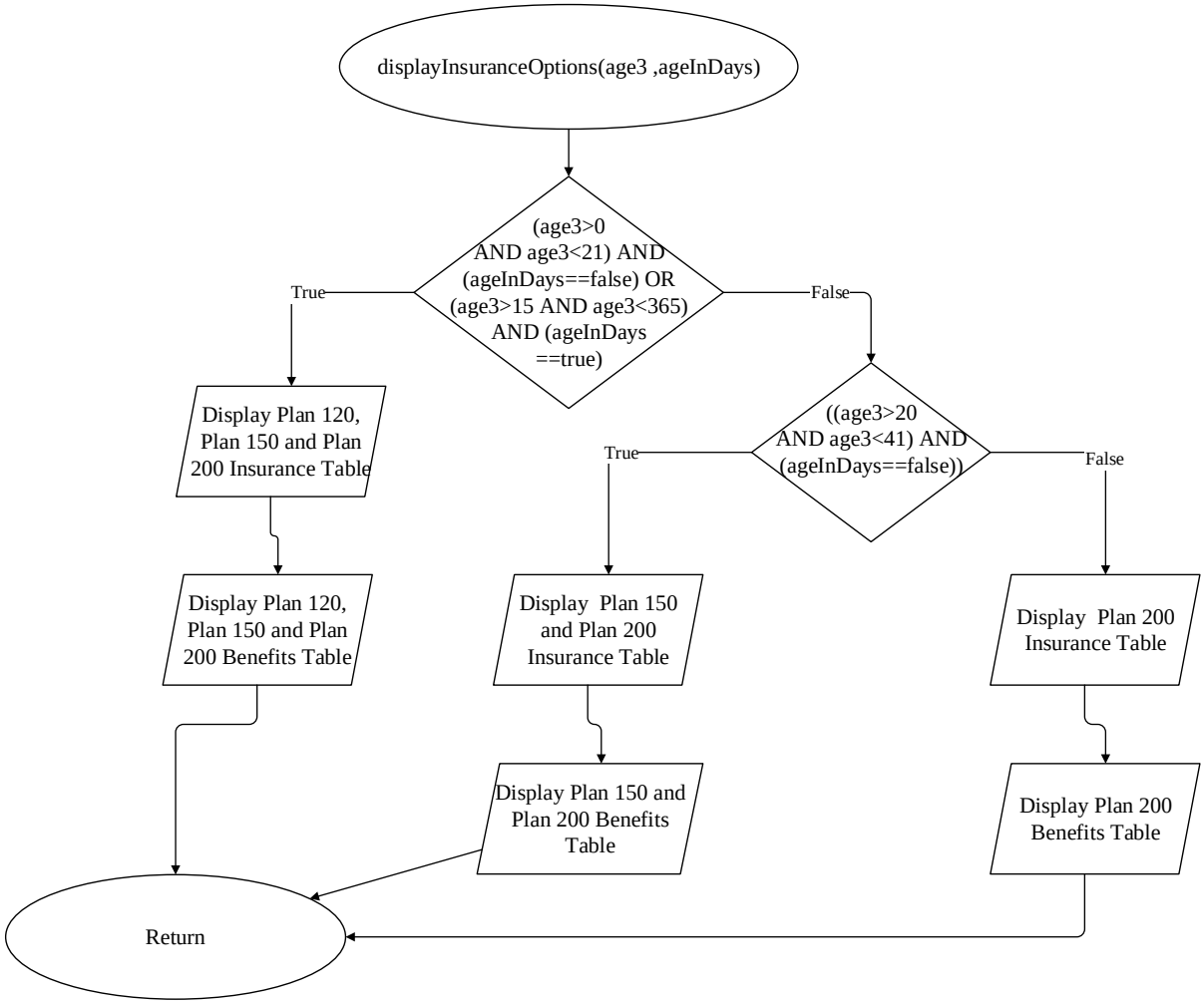
Flowchart for function getAgeInDays



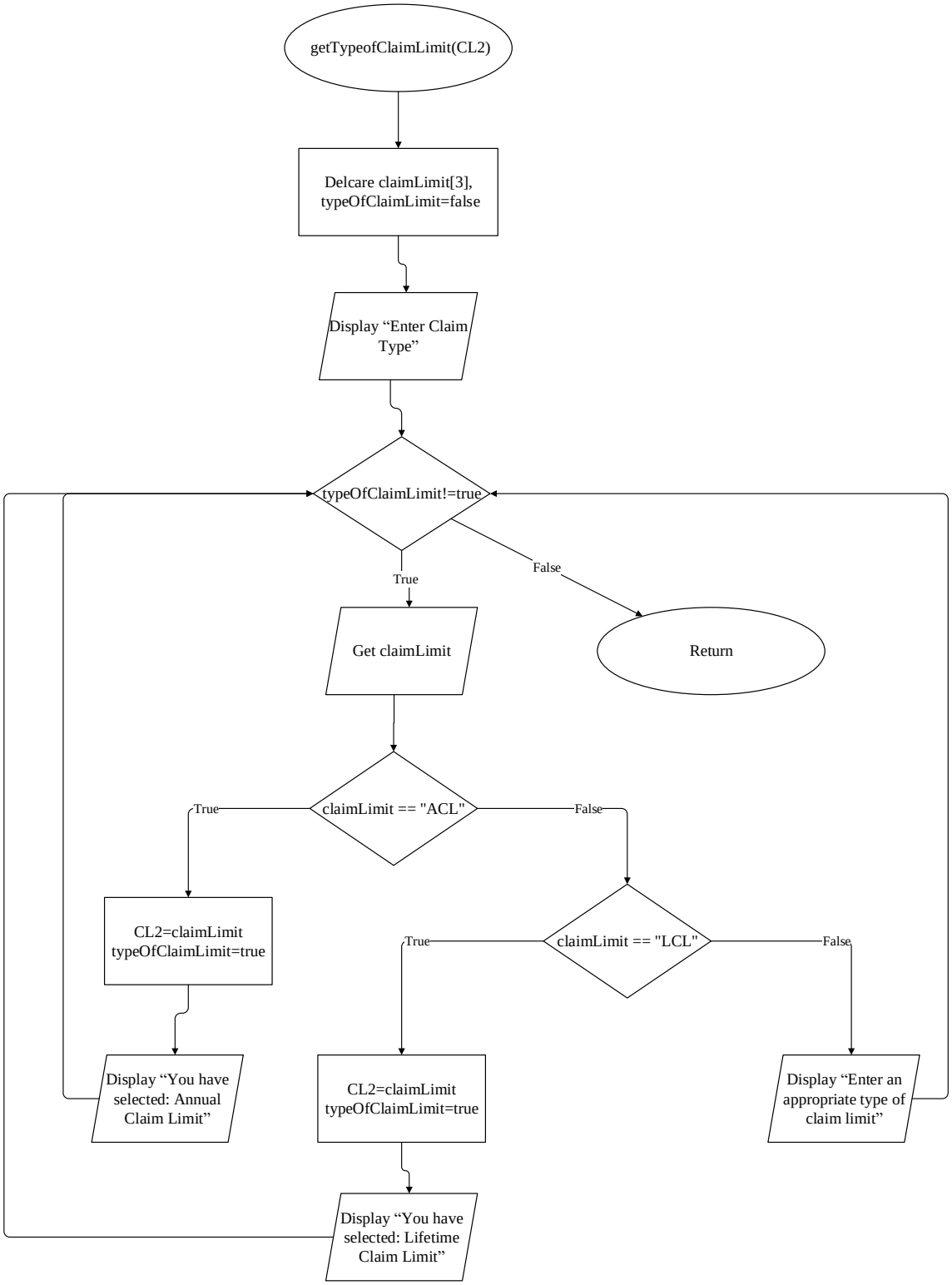
Flowchart for function getAgeInYears



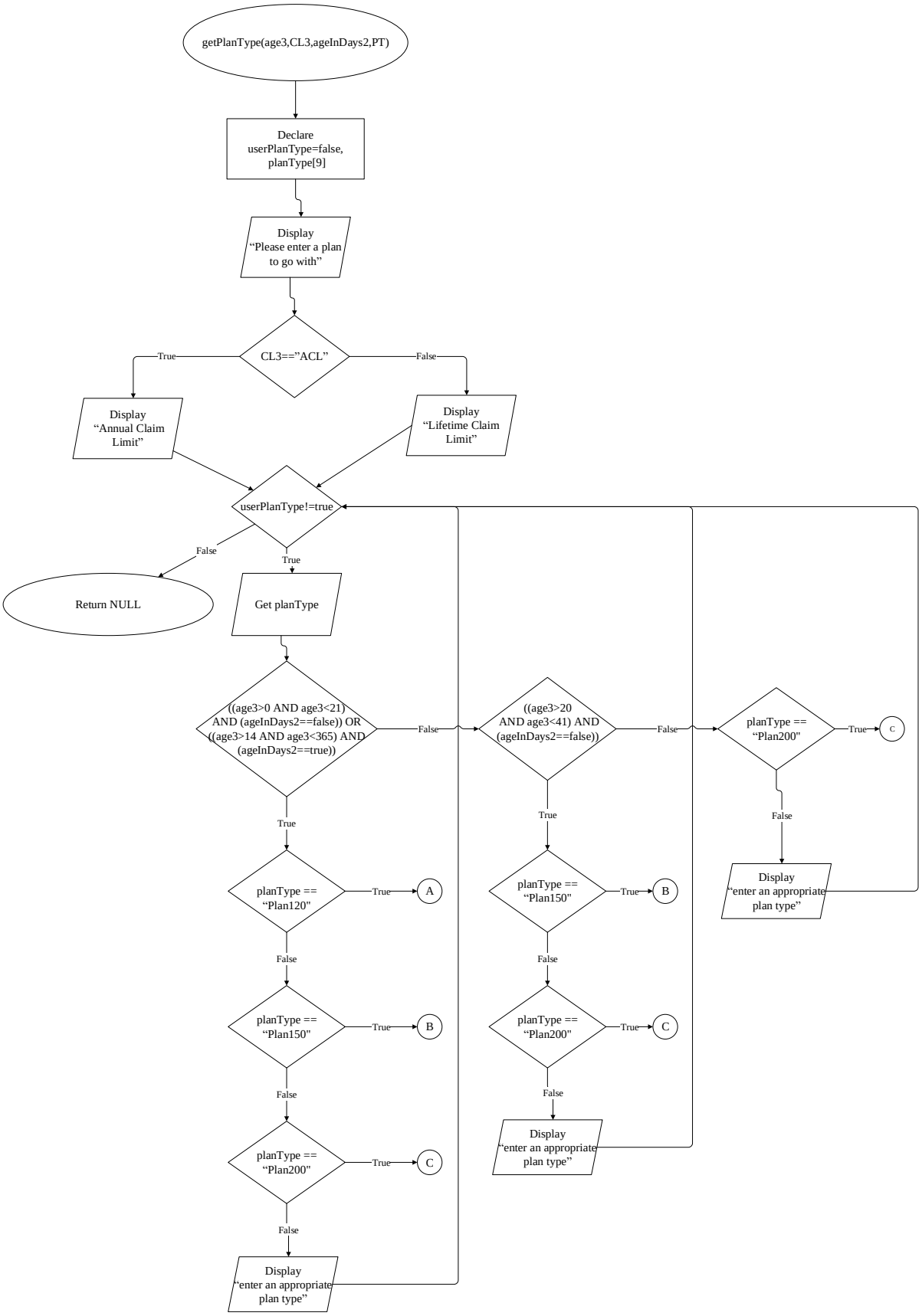
Flowchart for function displayIsuranceOptions



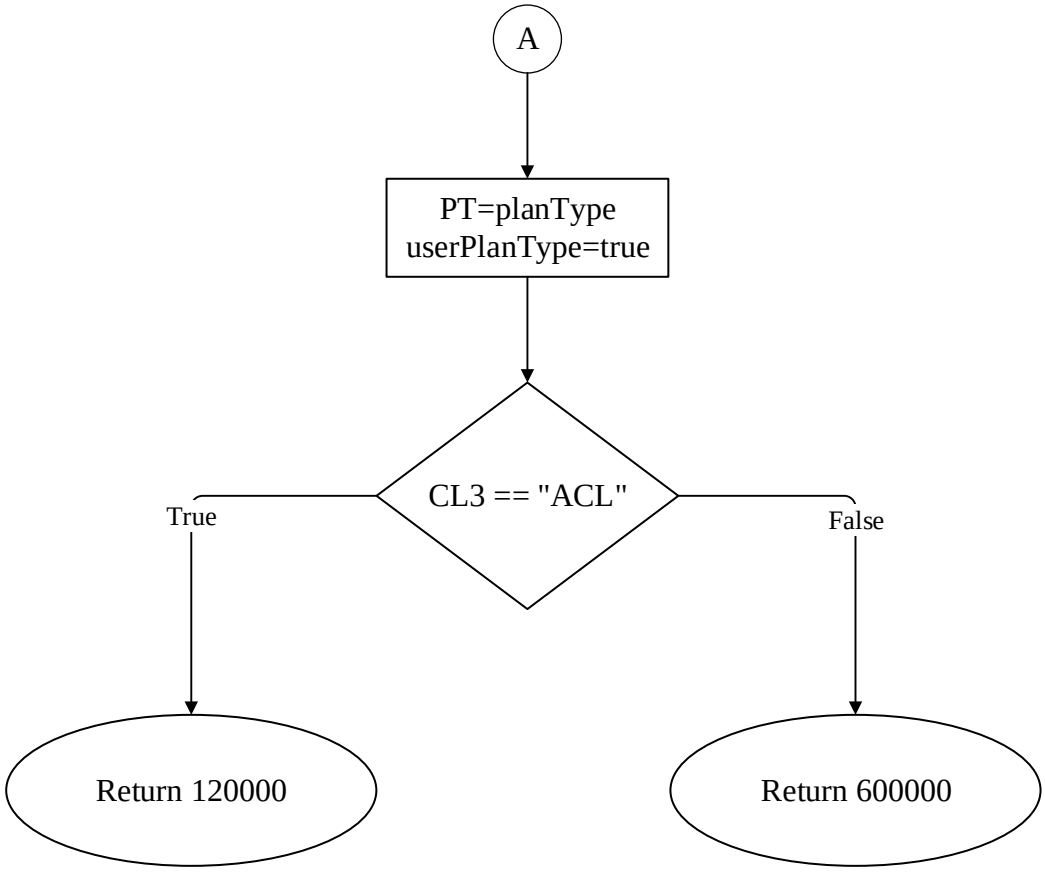
Flowchart for function getTypeofClaimLimit



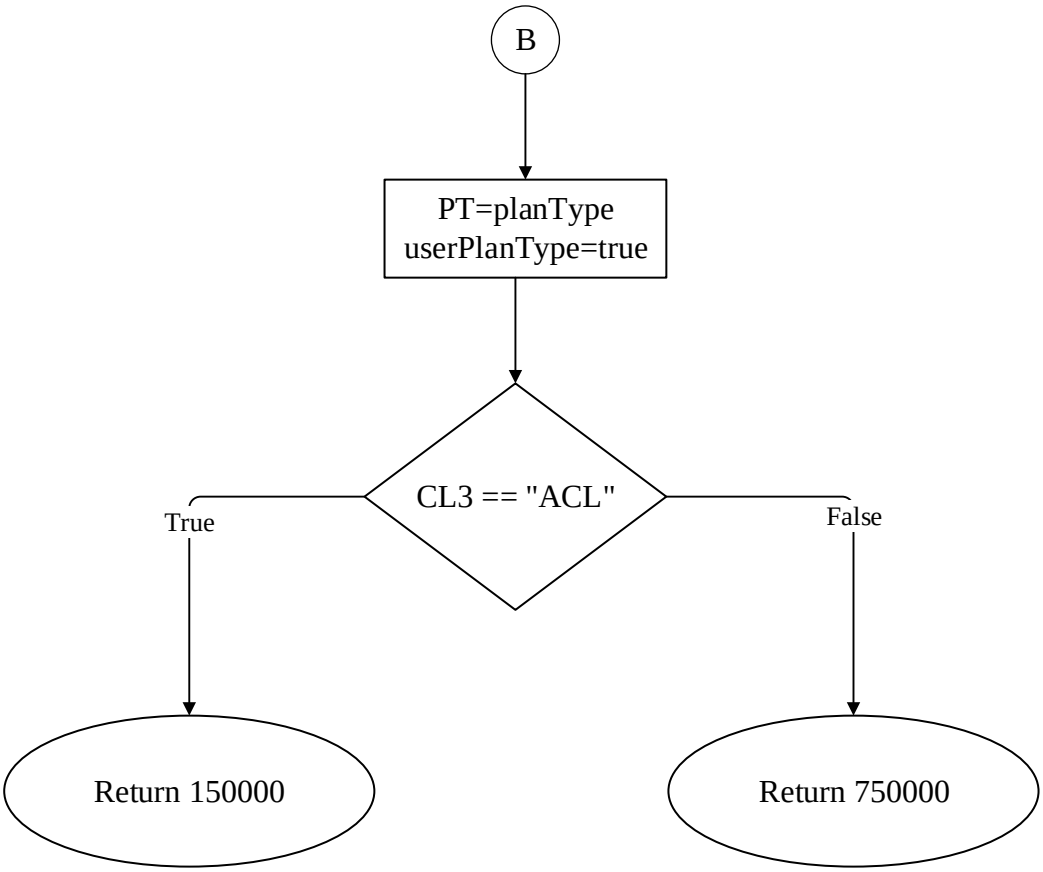
Flowchart for function getPlanType



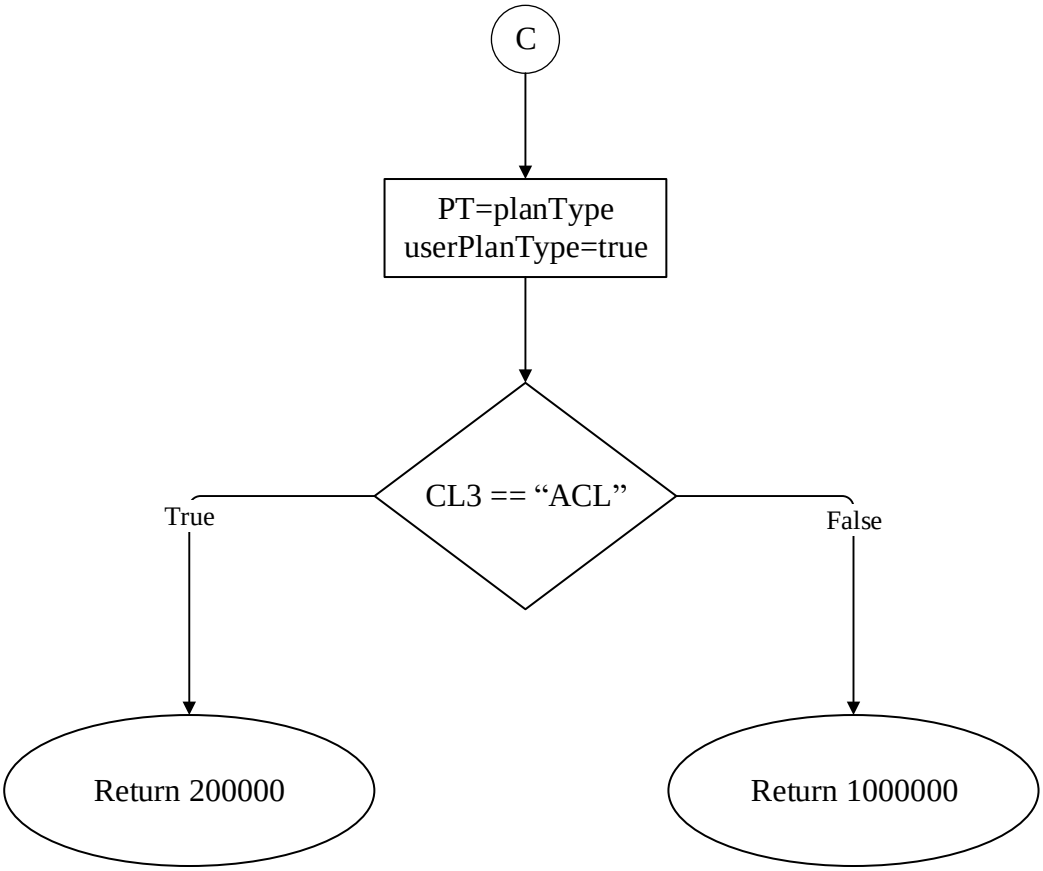
Same page connector A for function getPlanType



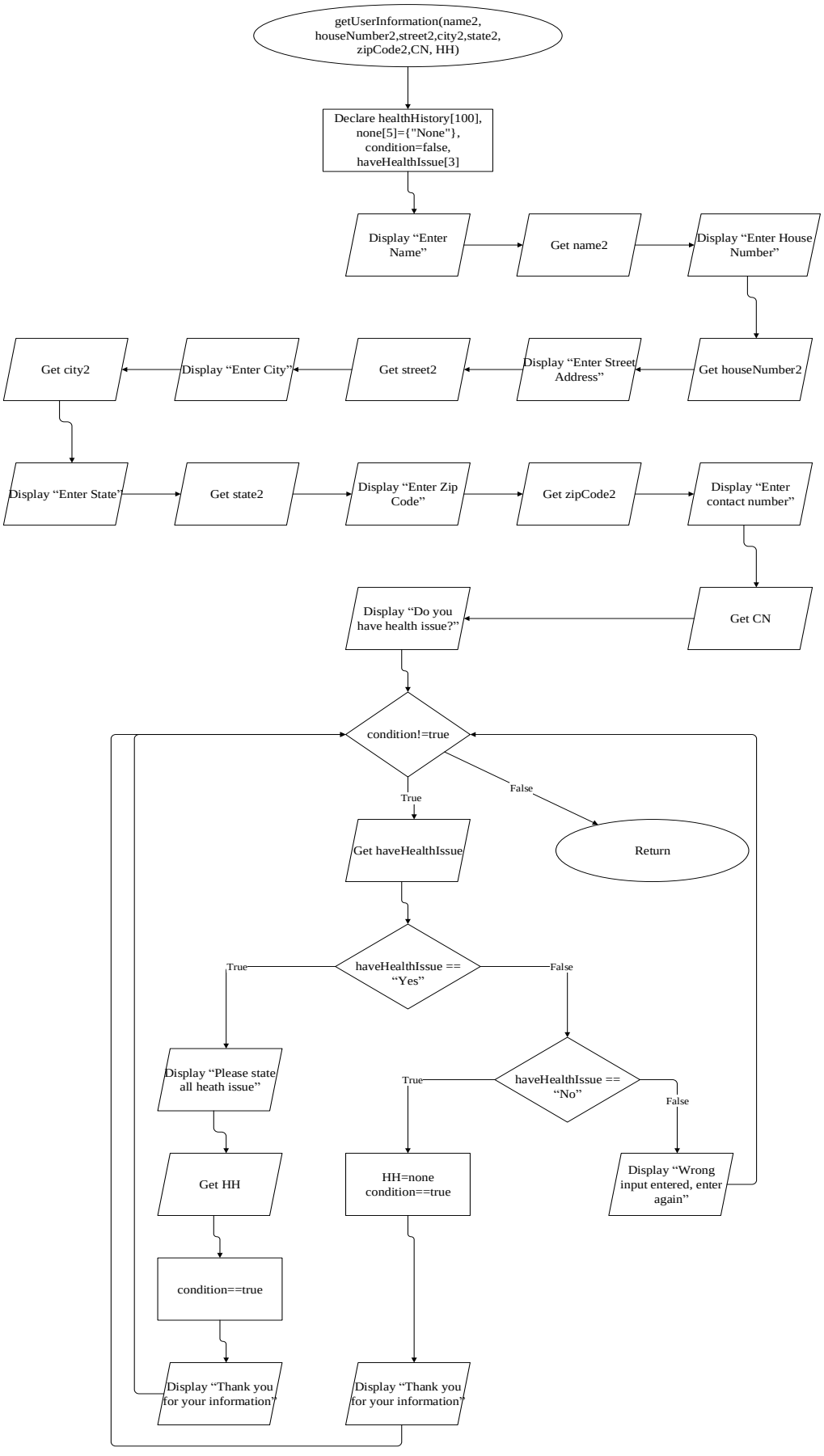
Same page connector B for function getPlanType



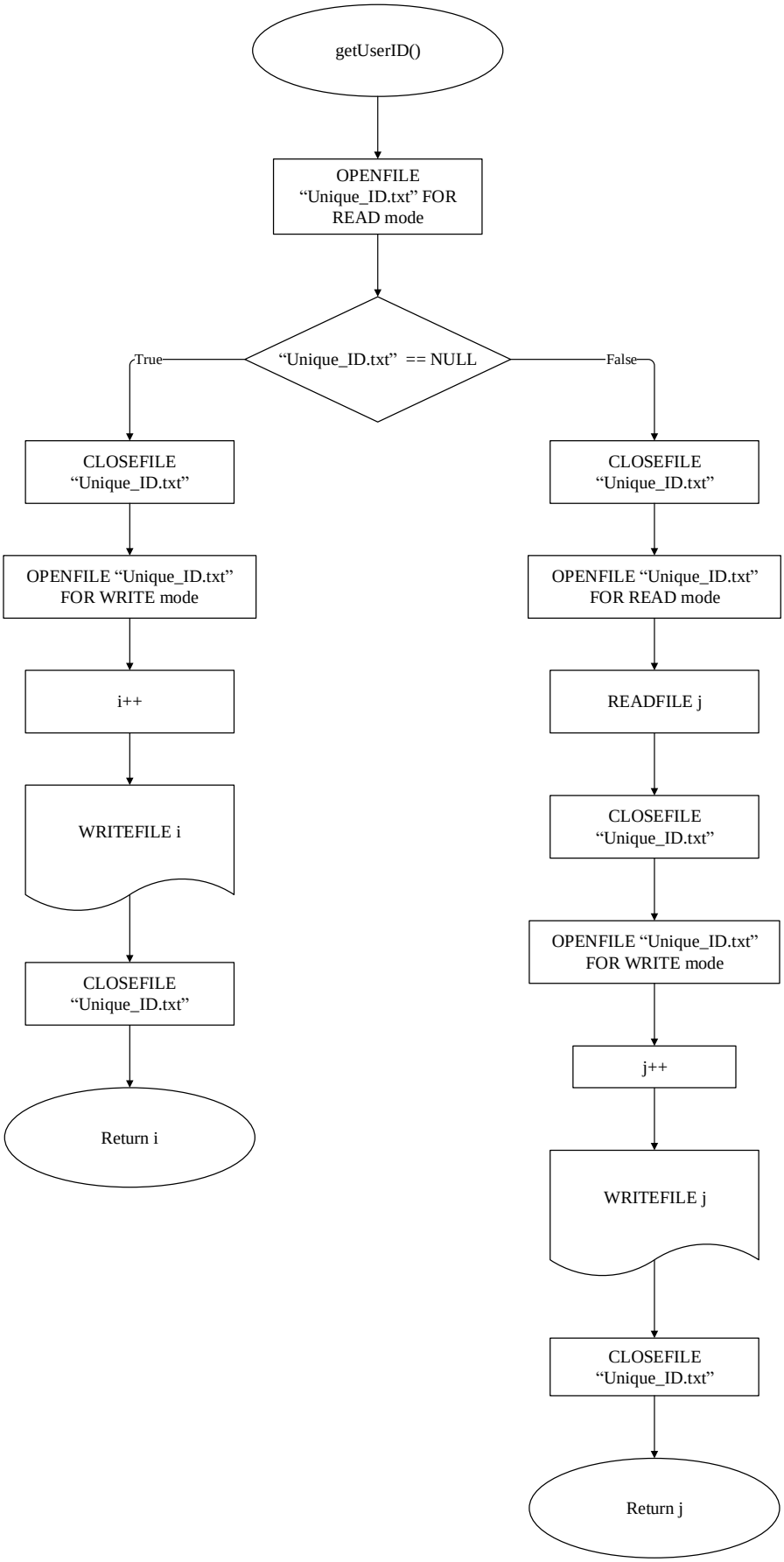
Same page connector C for function getPlanType



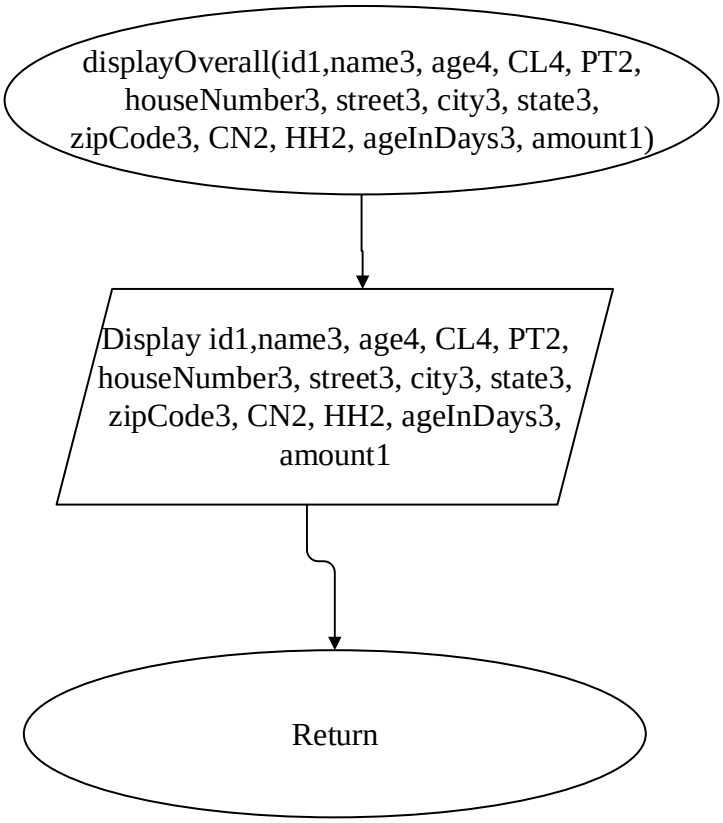
Flowchart for function getUserInformation



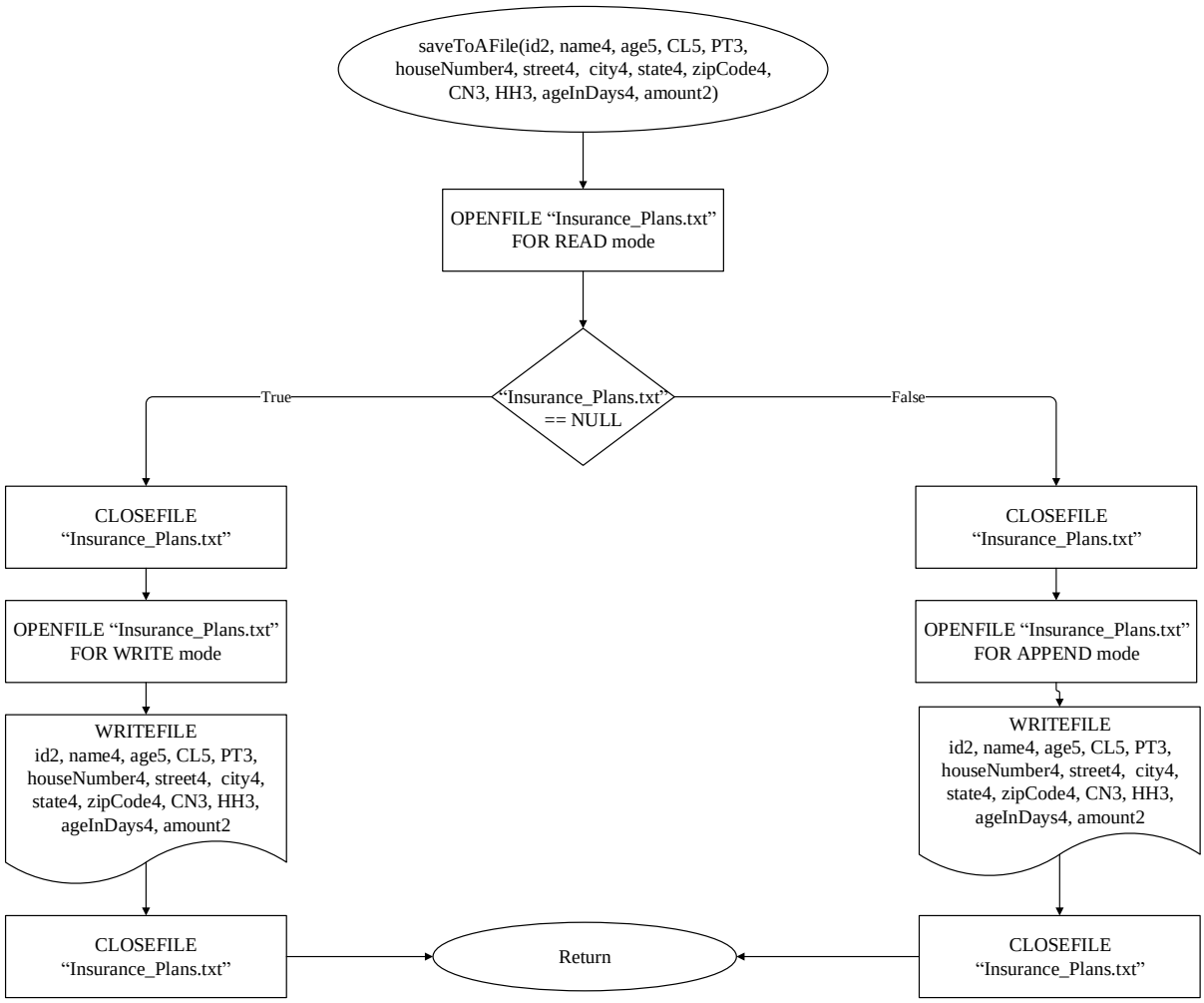
Flowchart for function getUserID



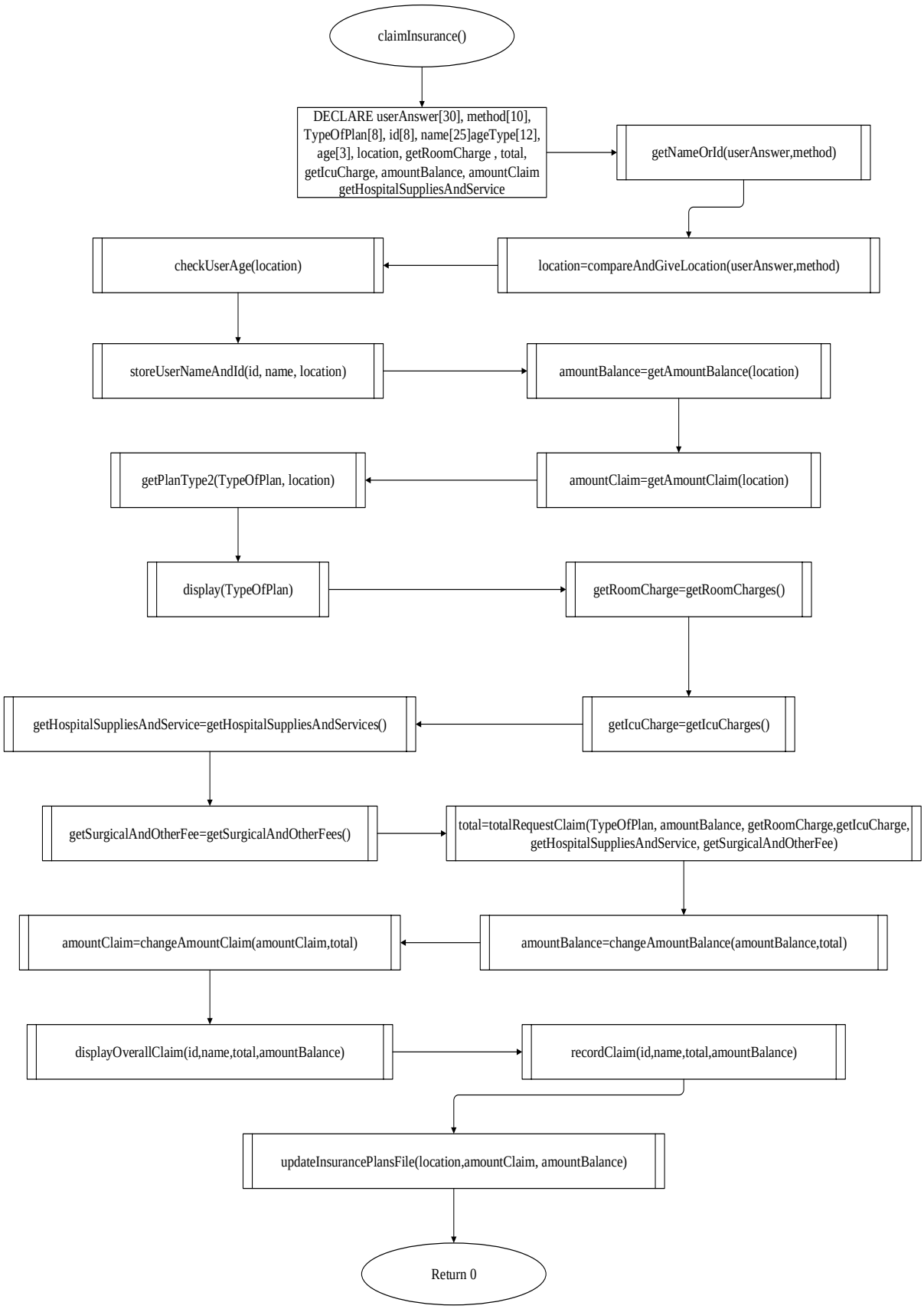
Flowchart for function displayOverall



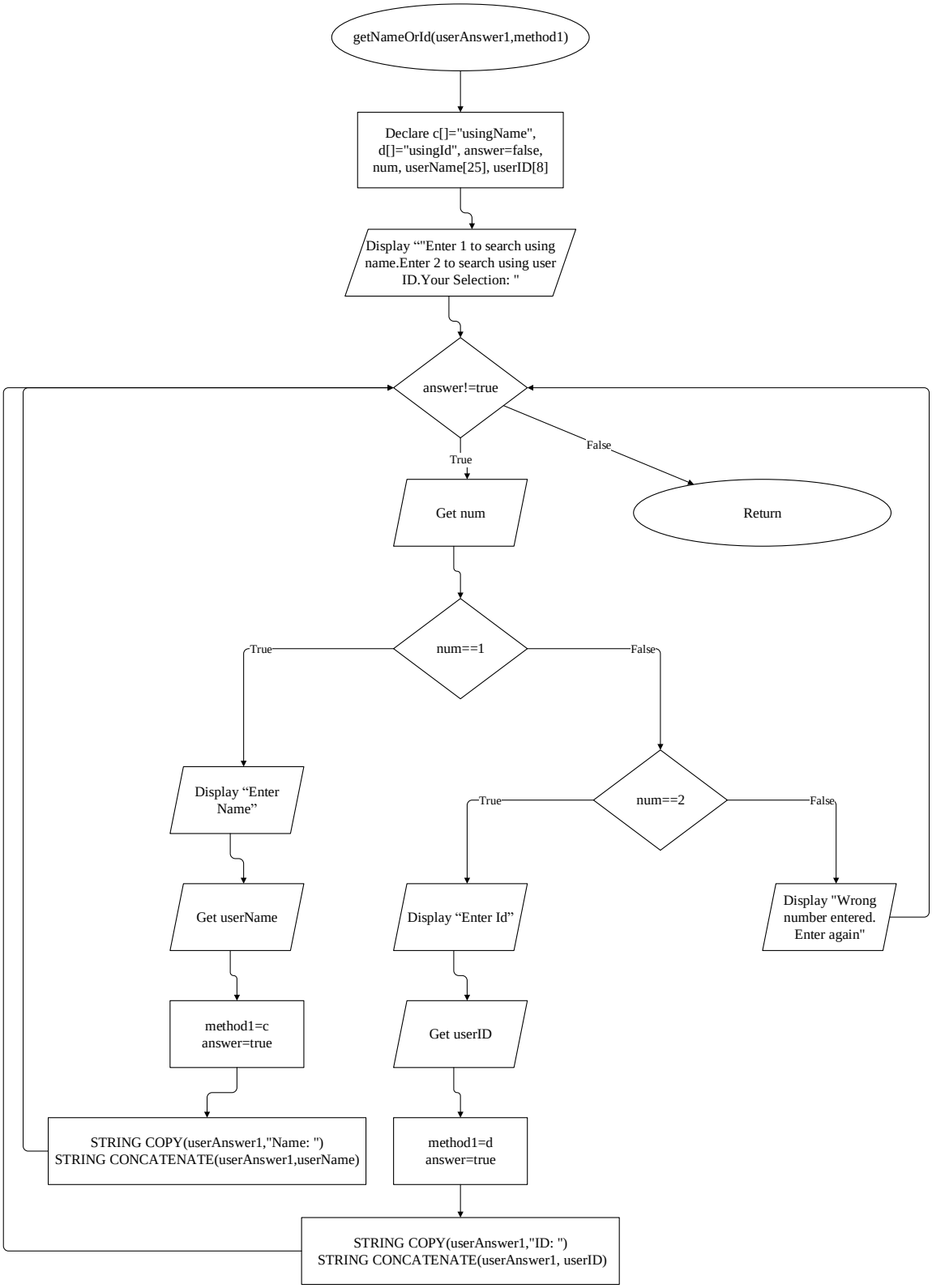
Flowchart for function saveToAFile



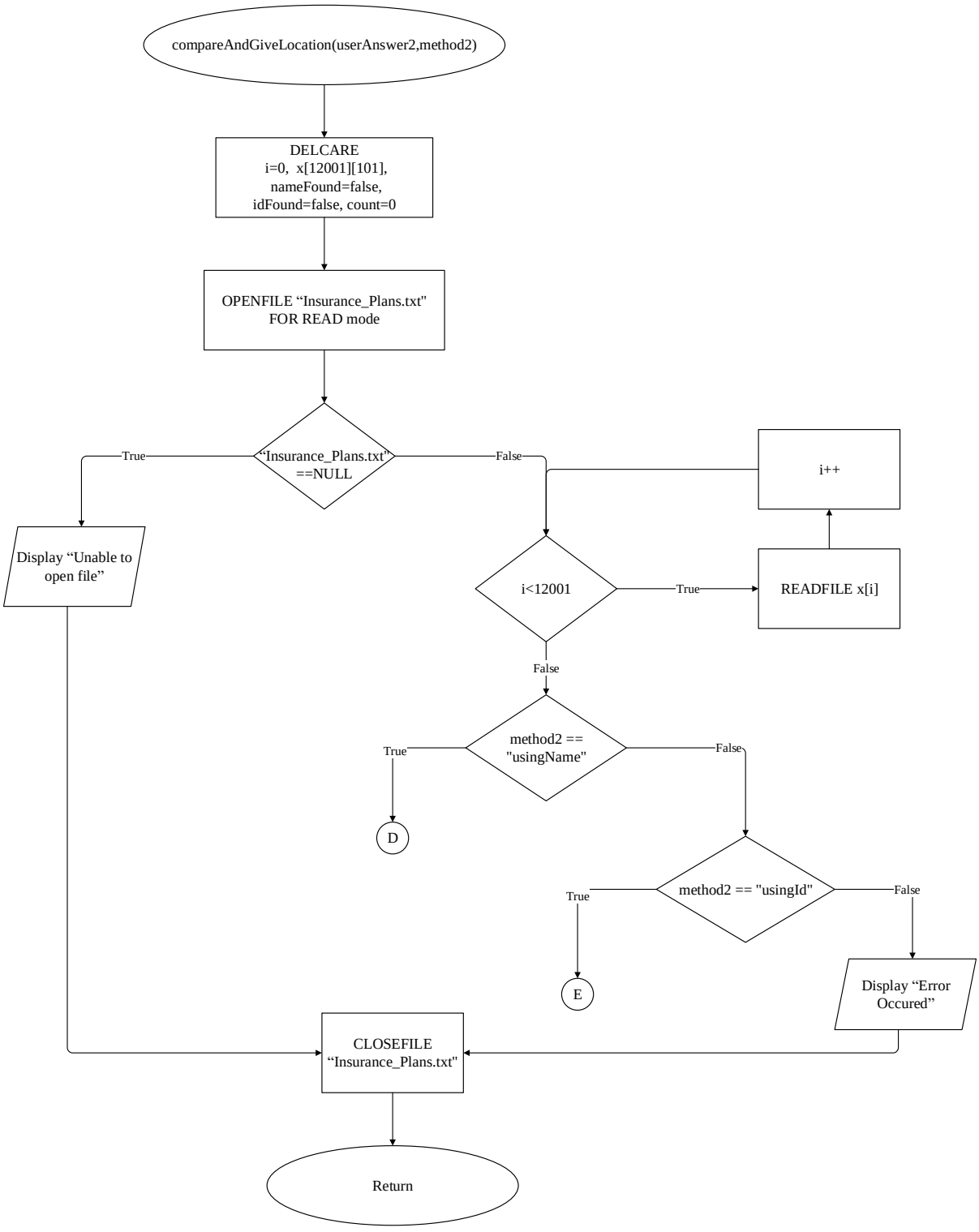
Flowchart for function claimInsurance



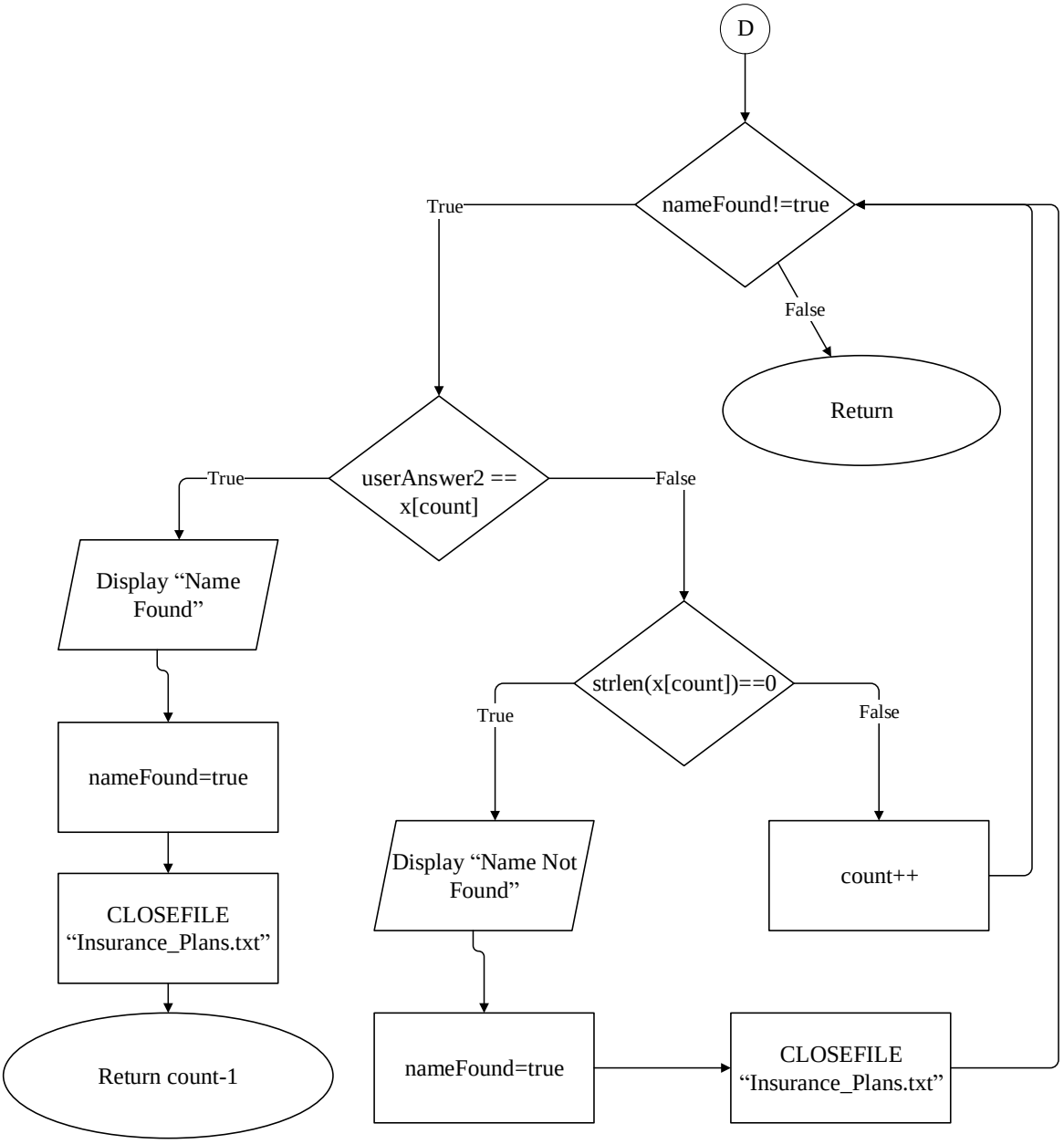
Flowchart for function getNameOrId



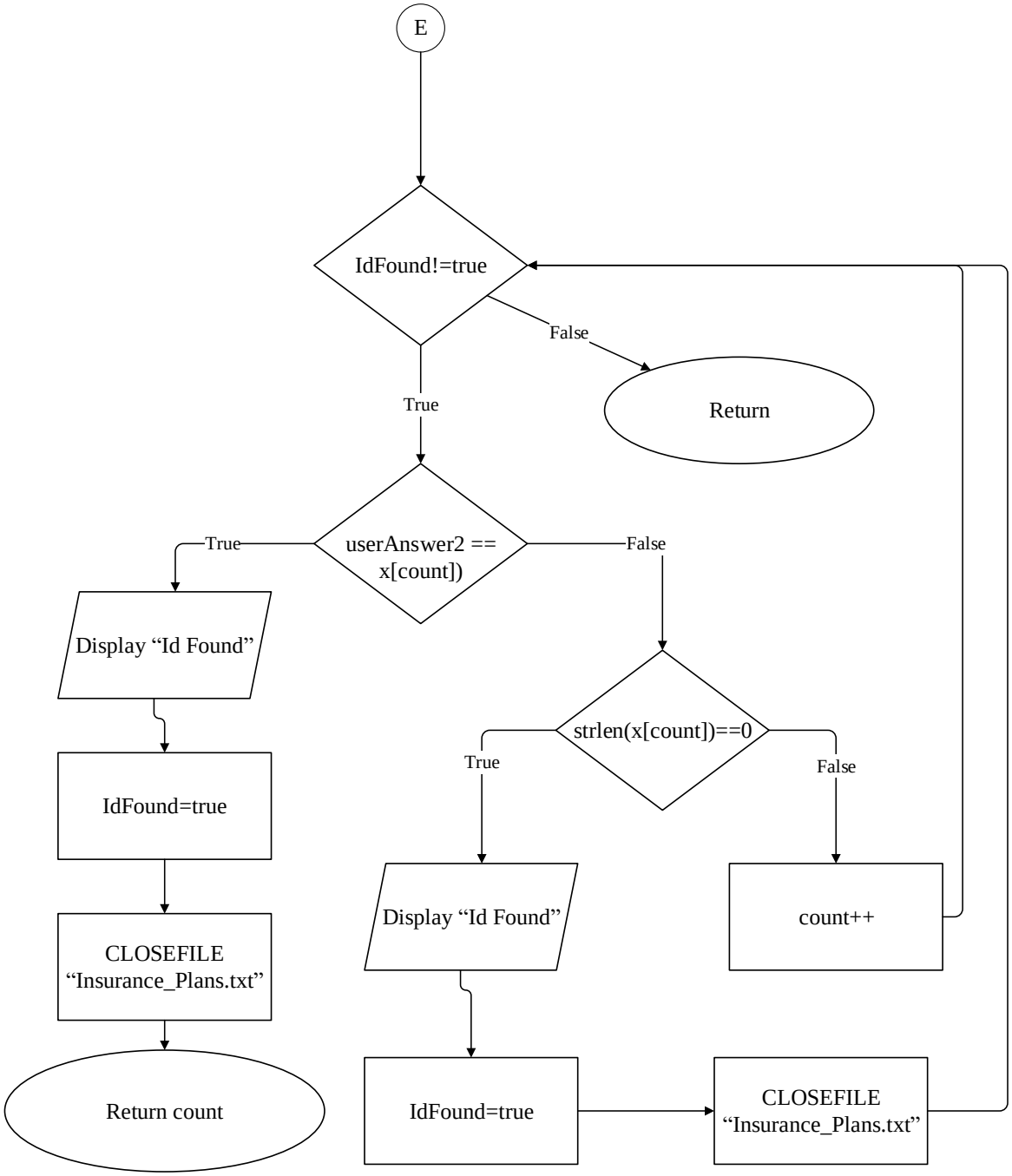
Flowchart for function compareAndGiveLocation



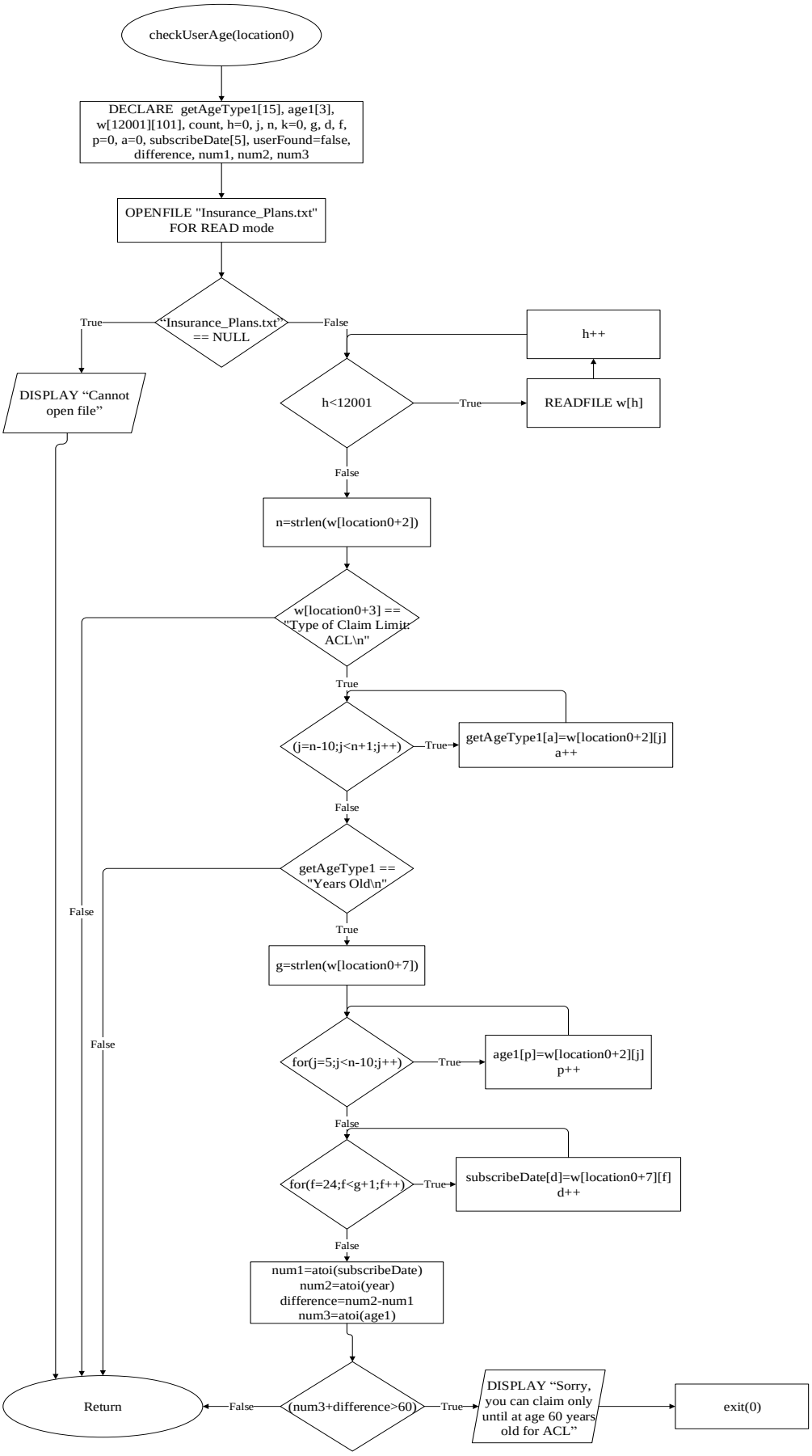
Same page connector D for function compareAndGiveLocation



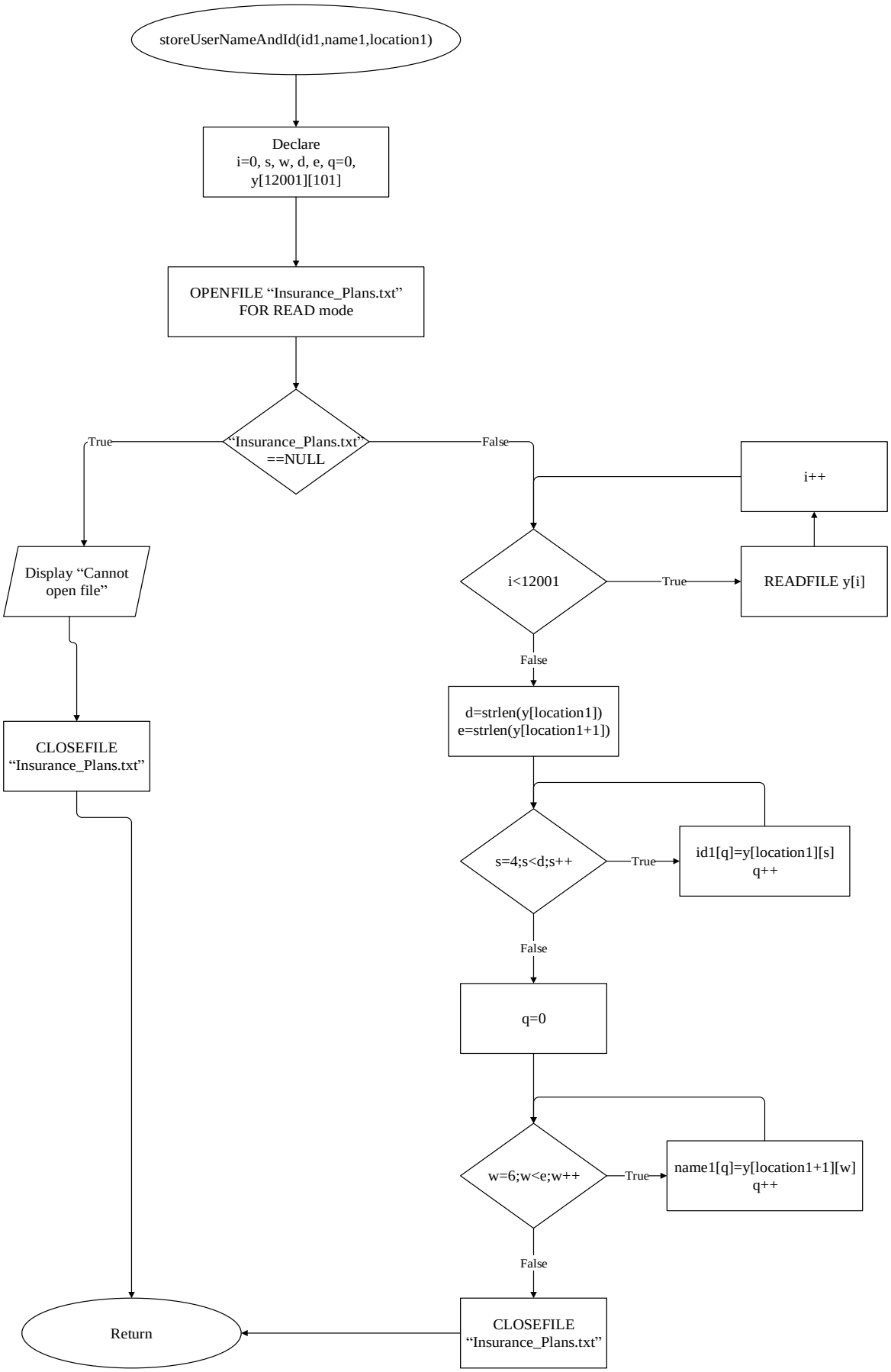
Same page connector E for function compareAndGiveLocation



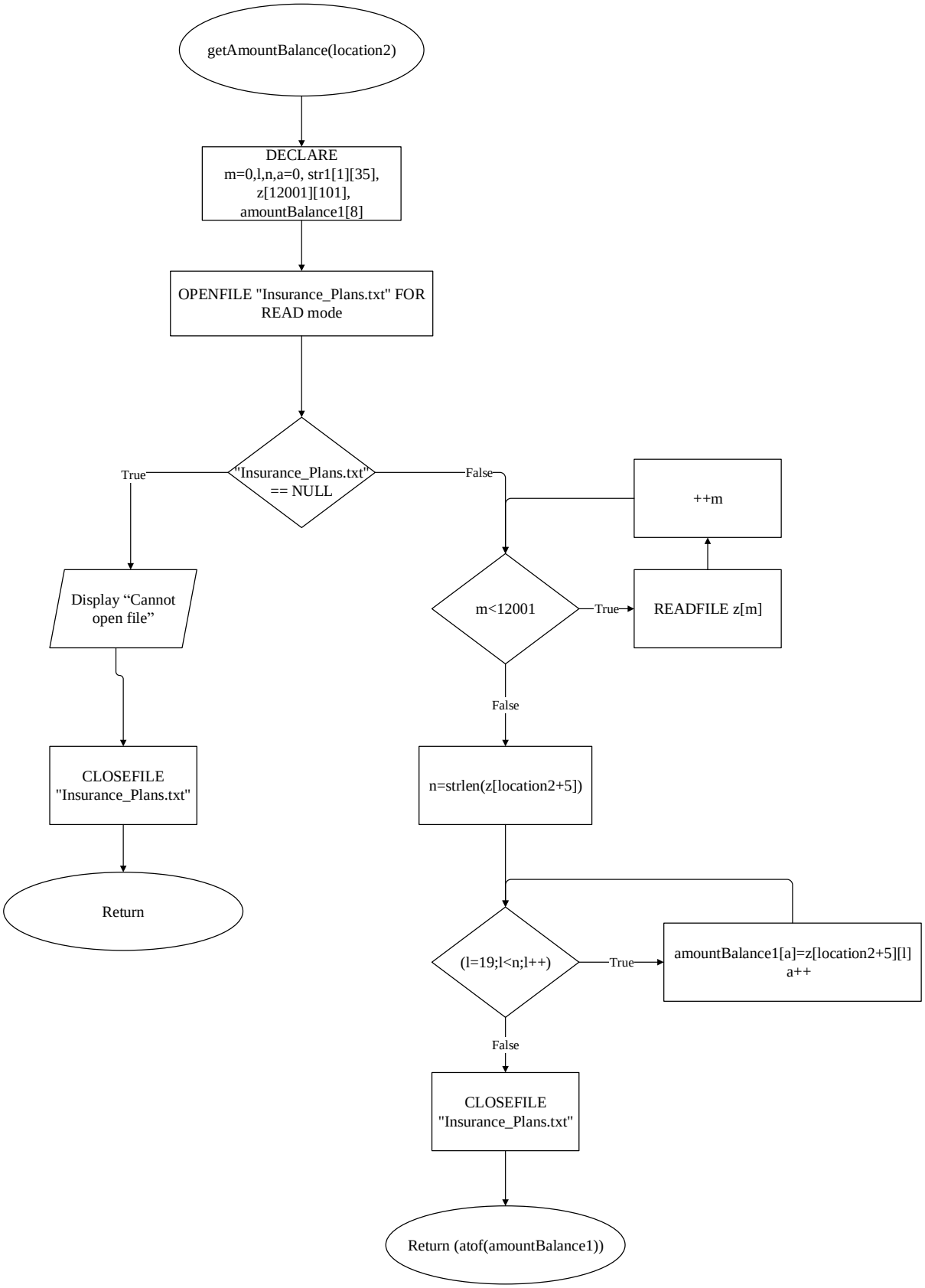
Flowchart for function checkUserAge



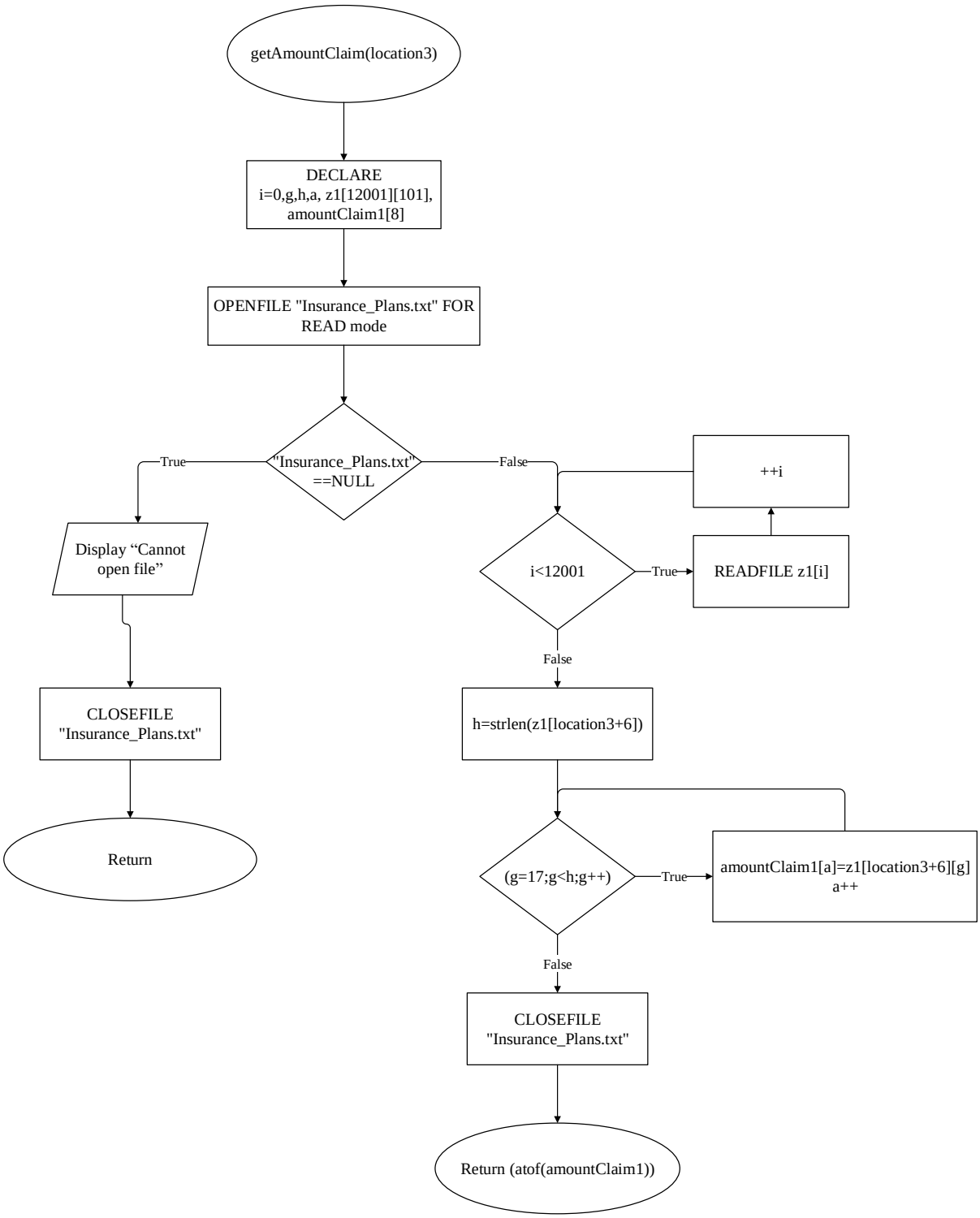
Flowchart for function storeUserNameAndId



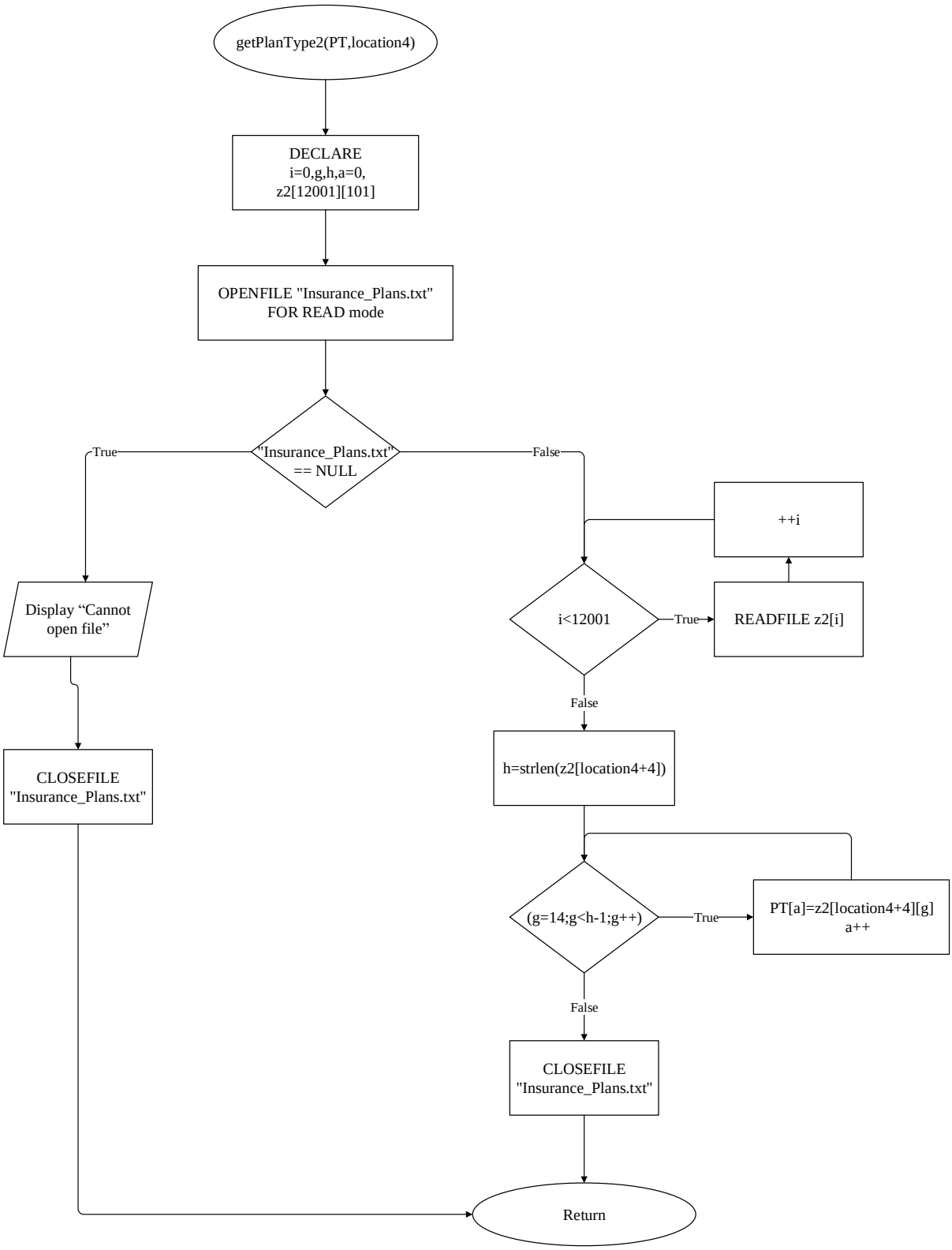
Flowchart for function getAmountBalance



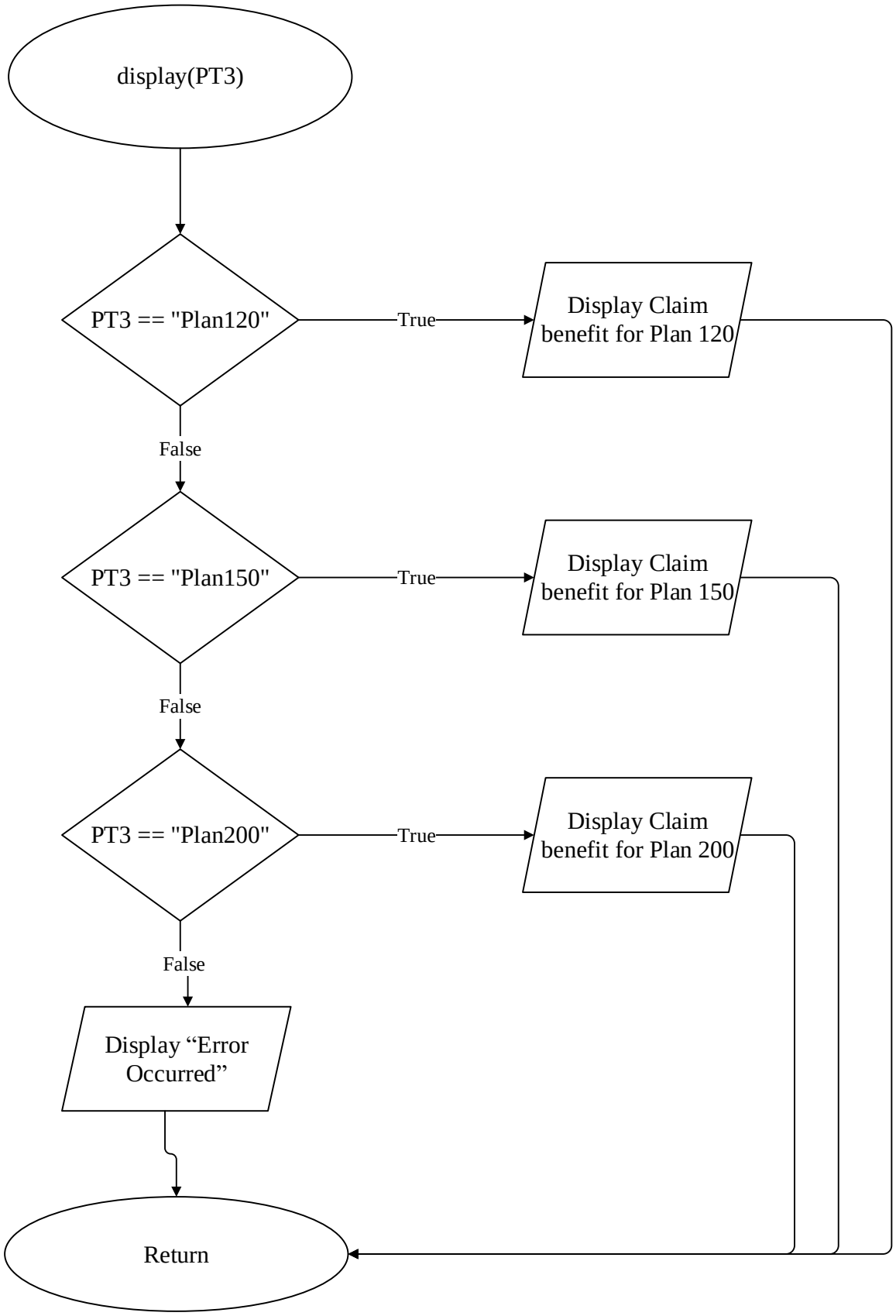
Flowchart for function getAmountClaim



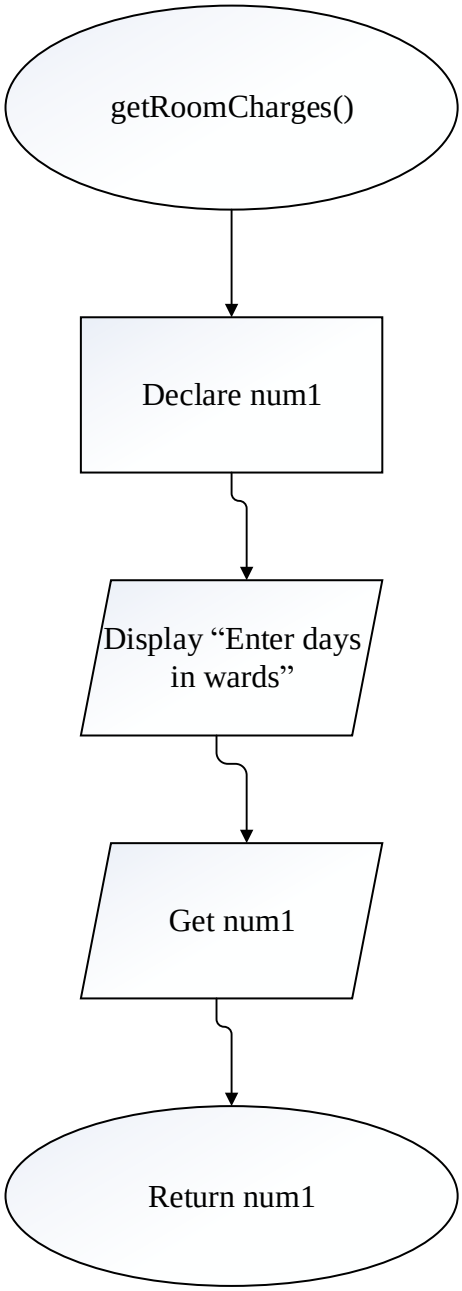
Flowchart for function getPlanType2



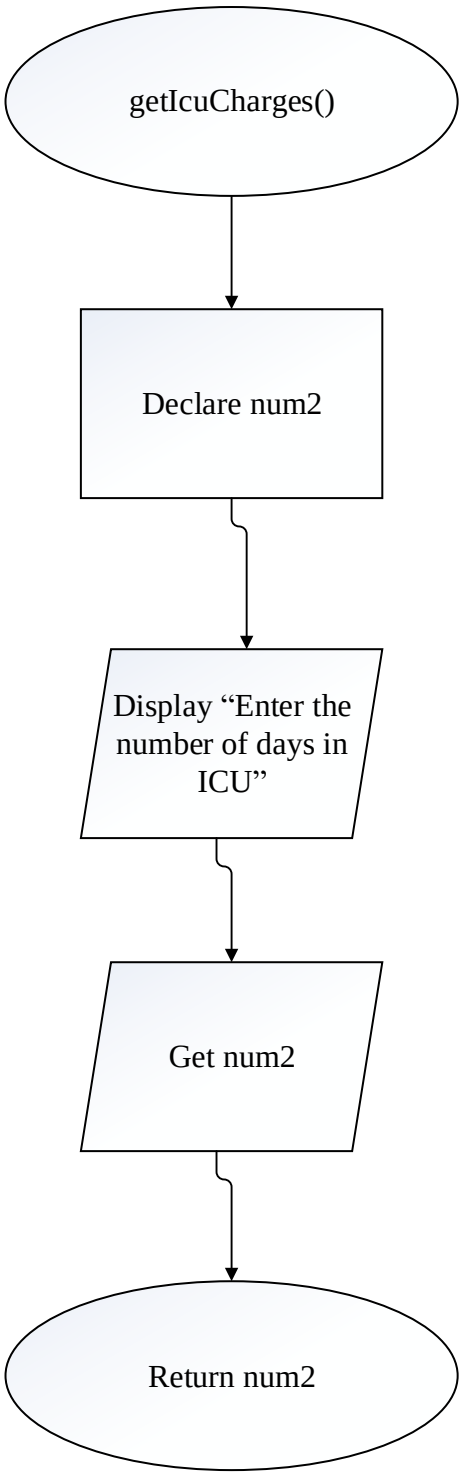
Flowchart for function display



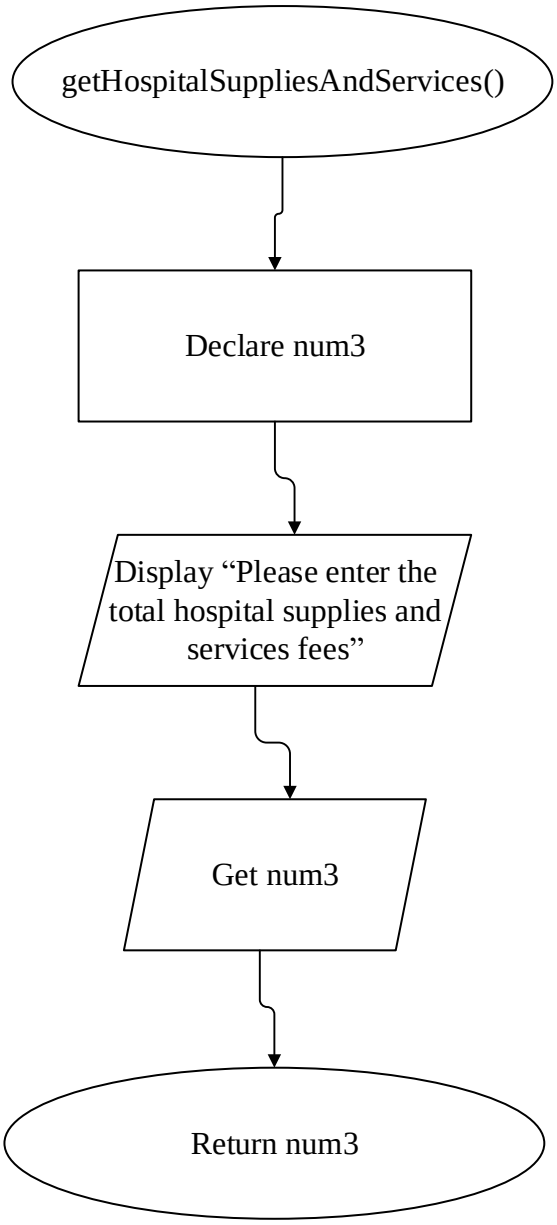
Flowchart for function getRoomCharges



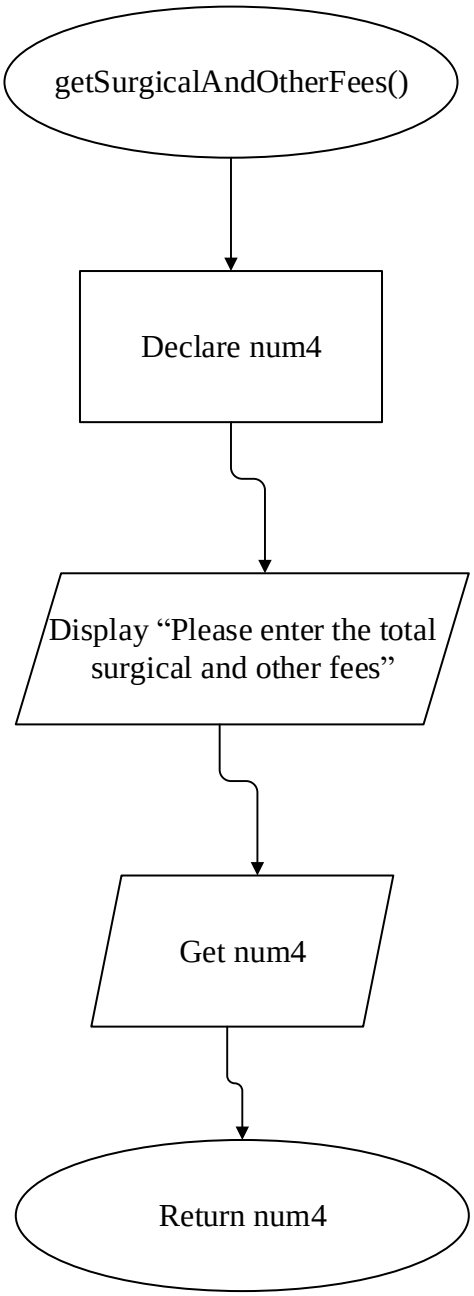
Flowchart for function getIcuCharges



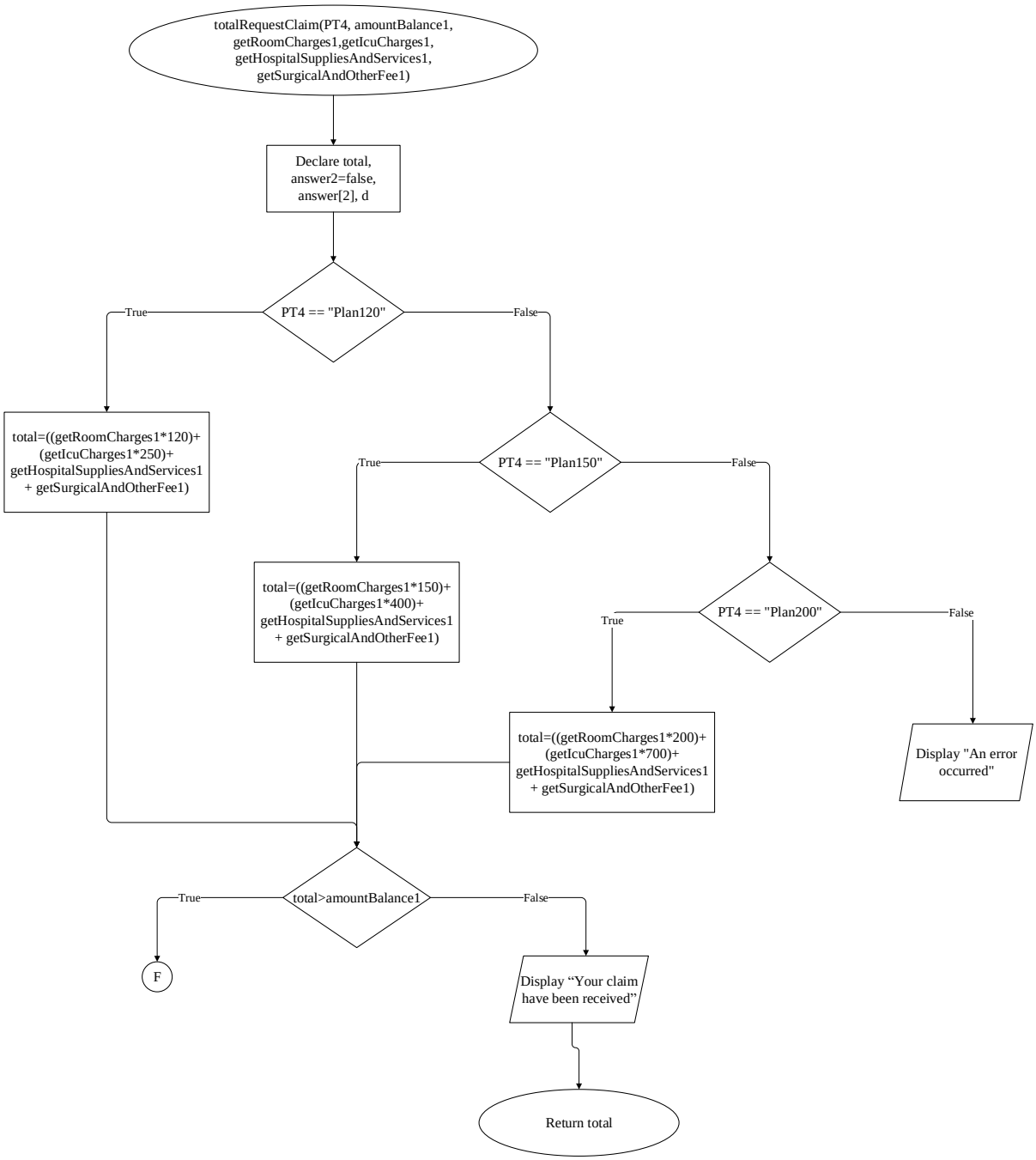
Flowchart for function getHospitalSuppliesAndServices



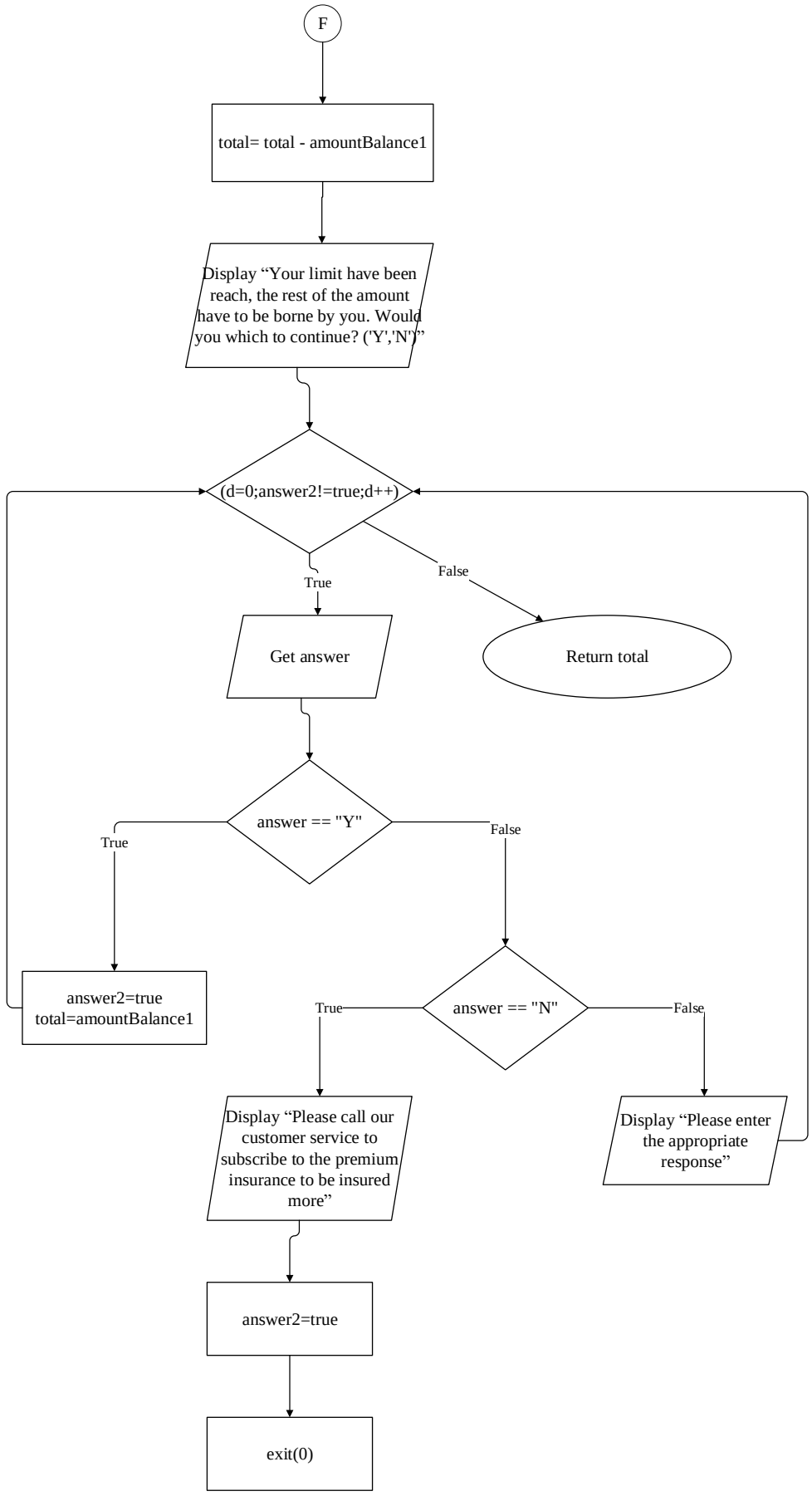
Flowchart for function getSurgicalAndOtherFees



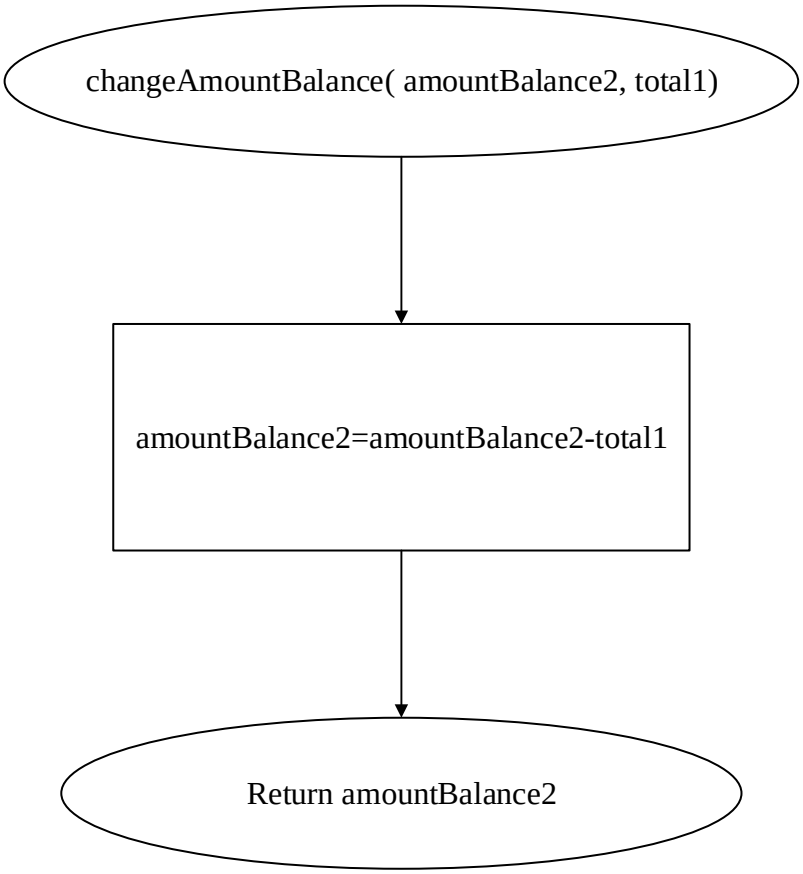
Flowchart for function totalRequestClaim



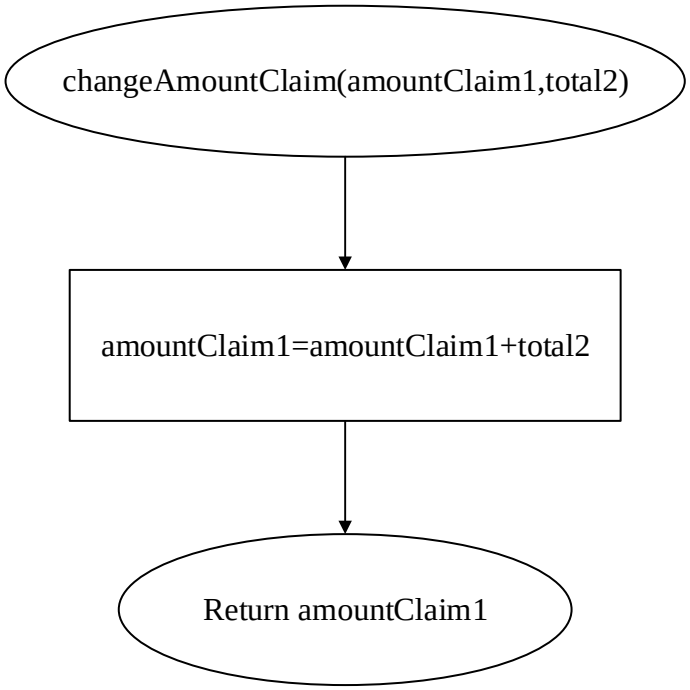
Same page connector F for function totalRequestClaim



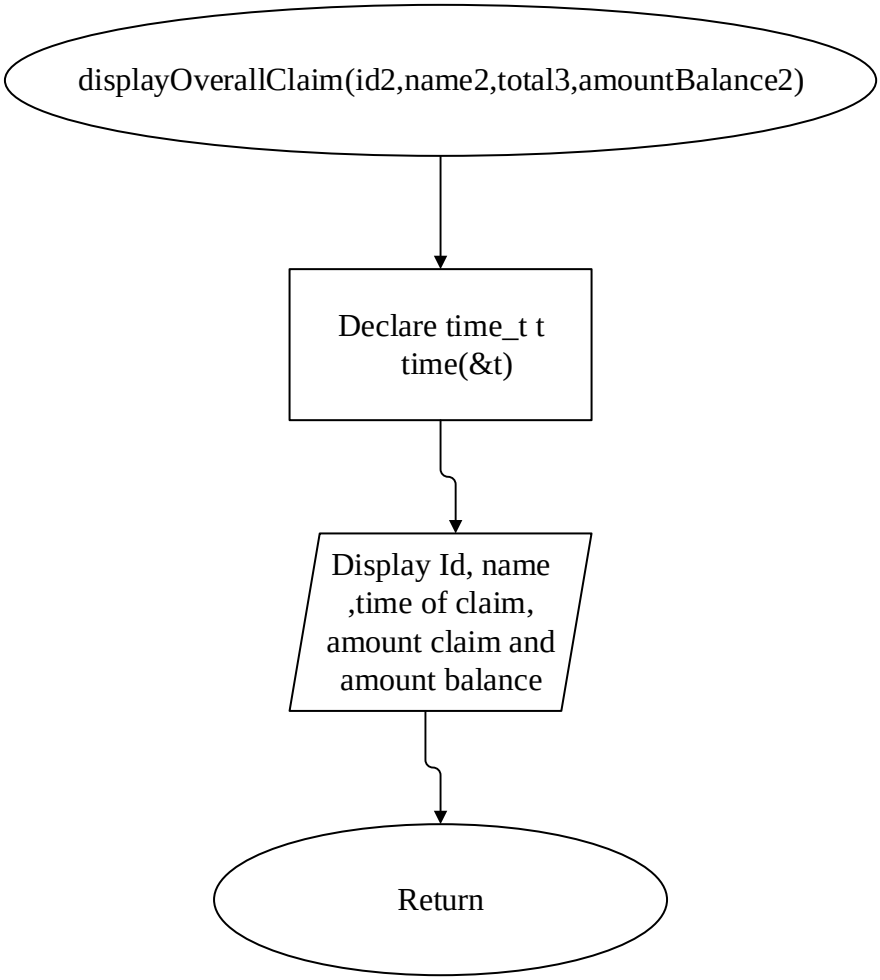
Flowchart for function changeAmountBalance



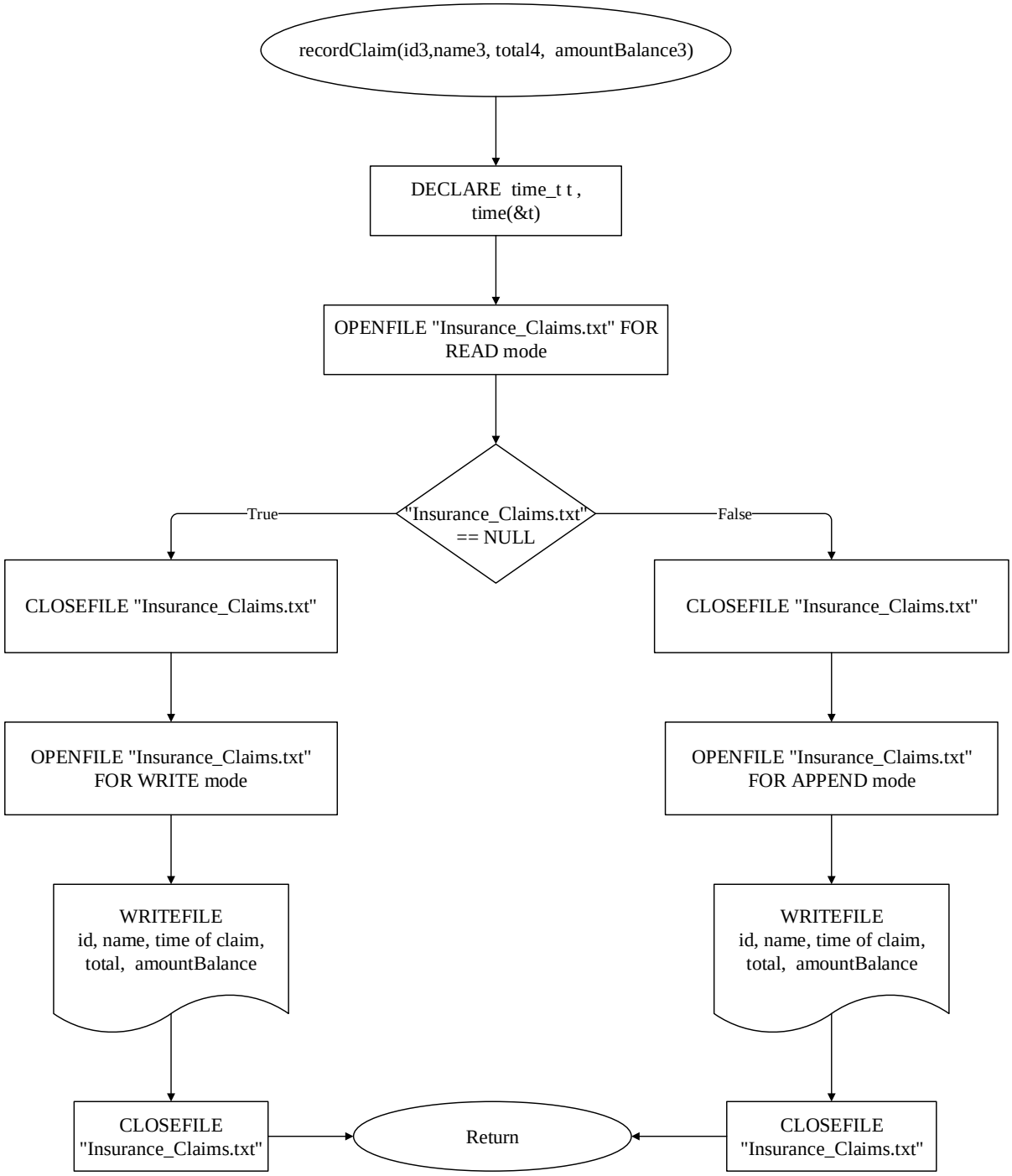
Flowchart for function changeAmountClaim



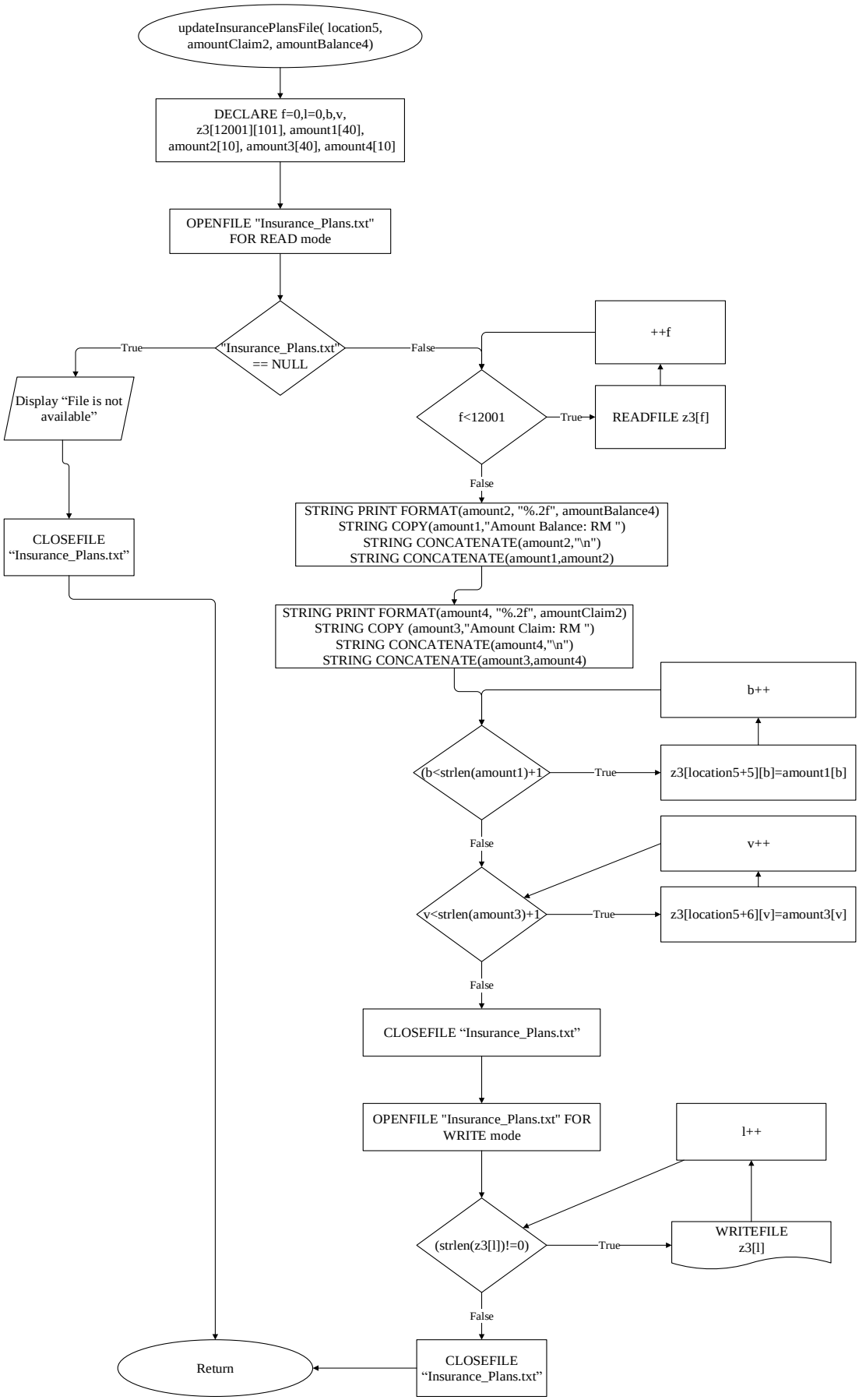
Flowchart for function displayOverallClaim



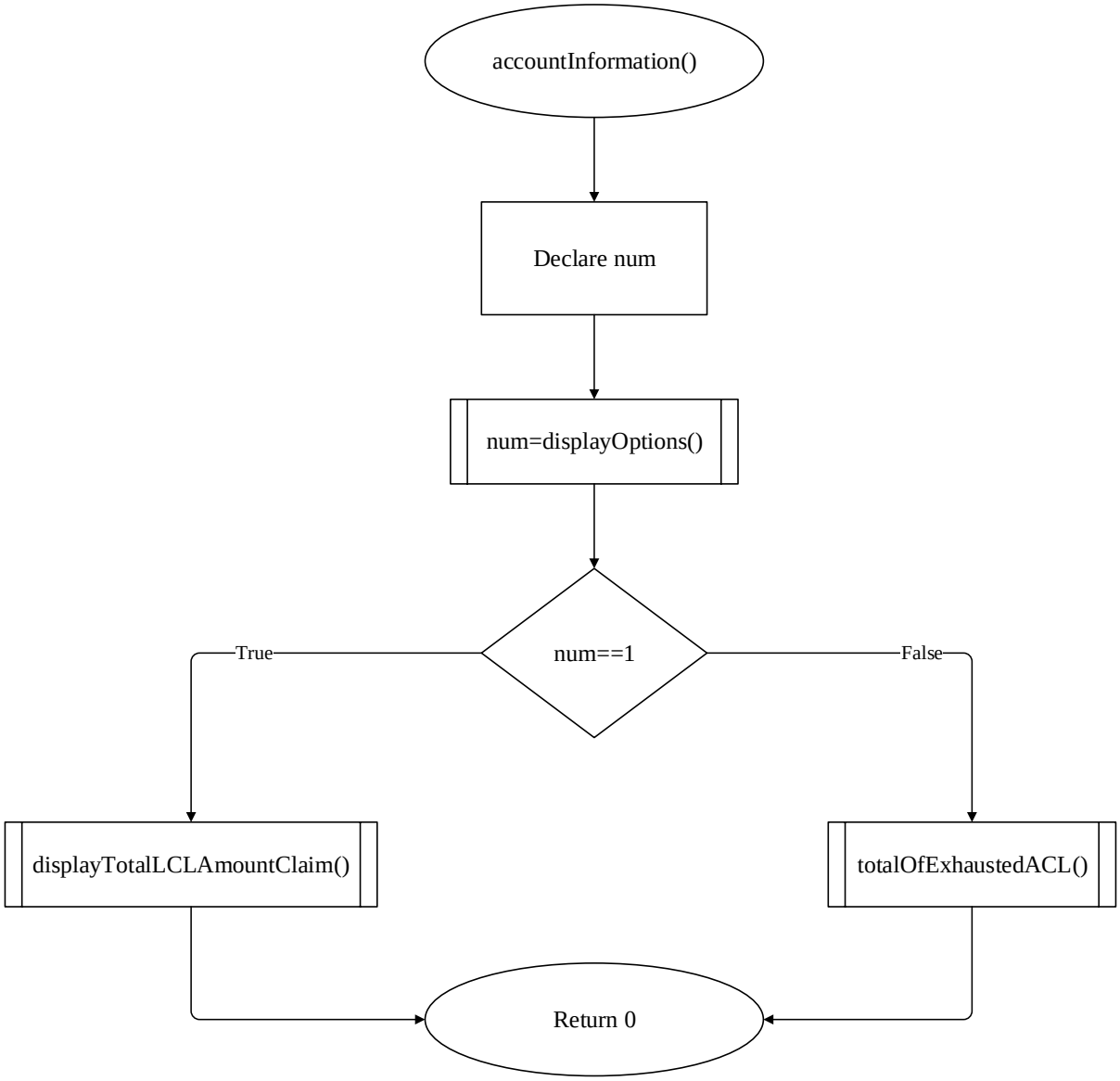
Flowchart for function recordClaim



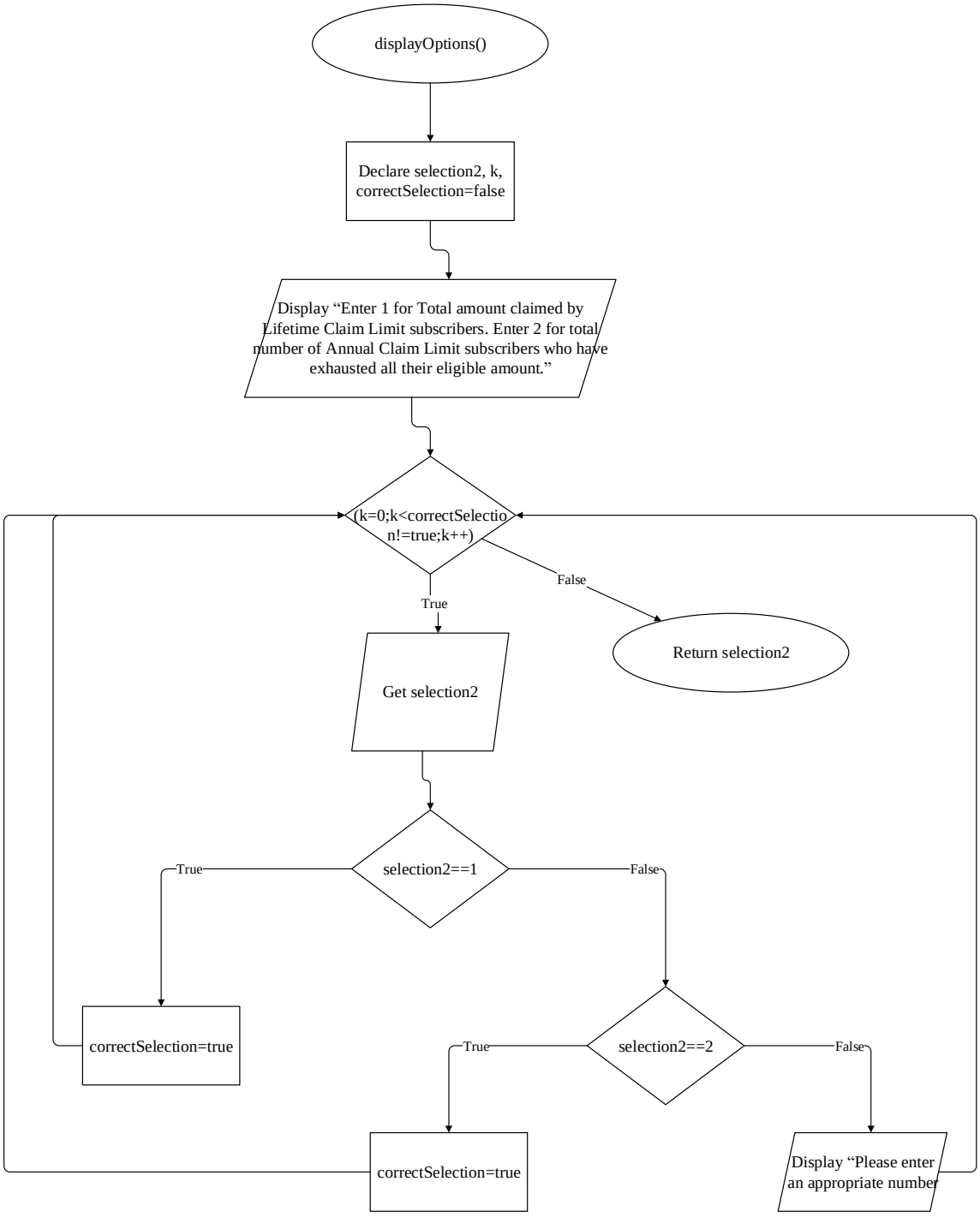
Flowchart for function updateInsurancePlansFile



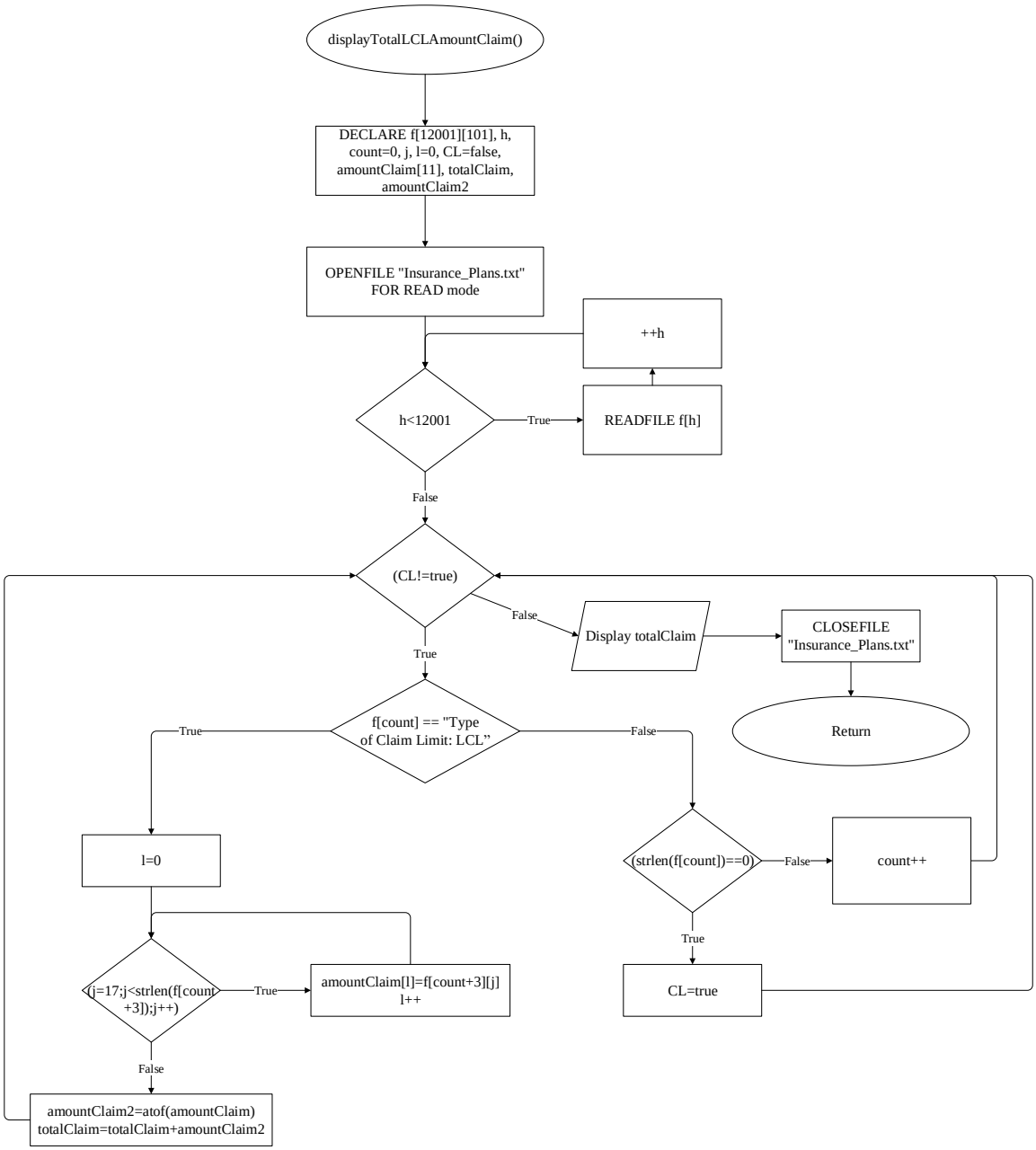
Flowchart for function accountInformation



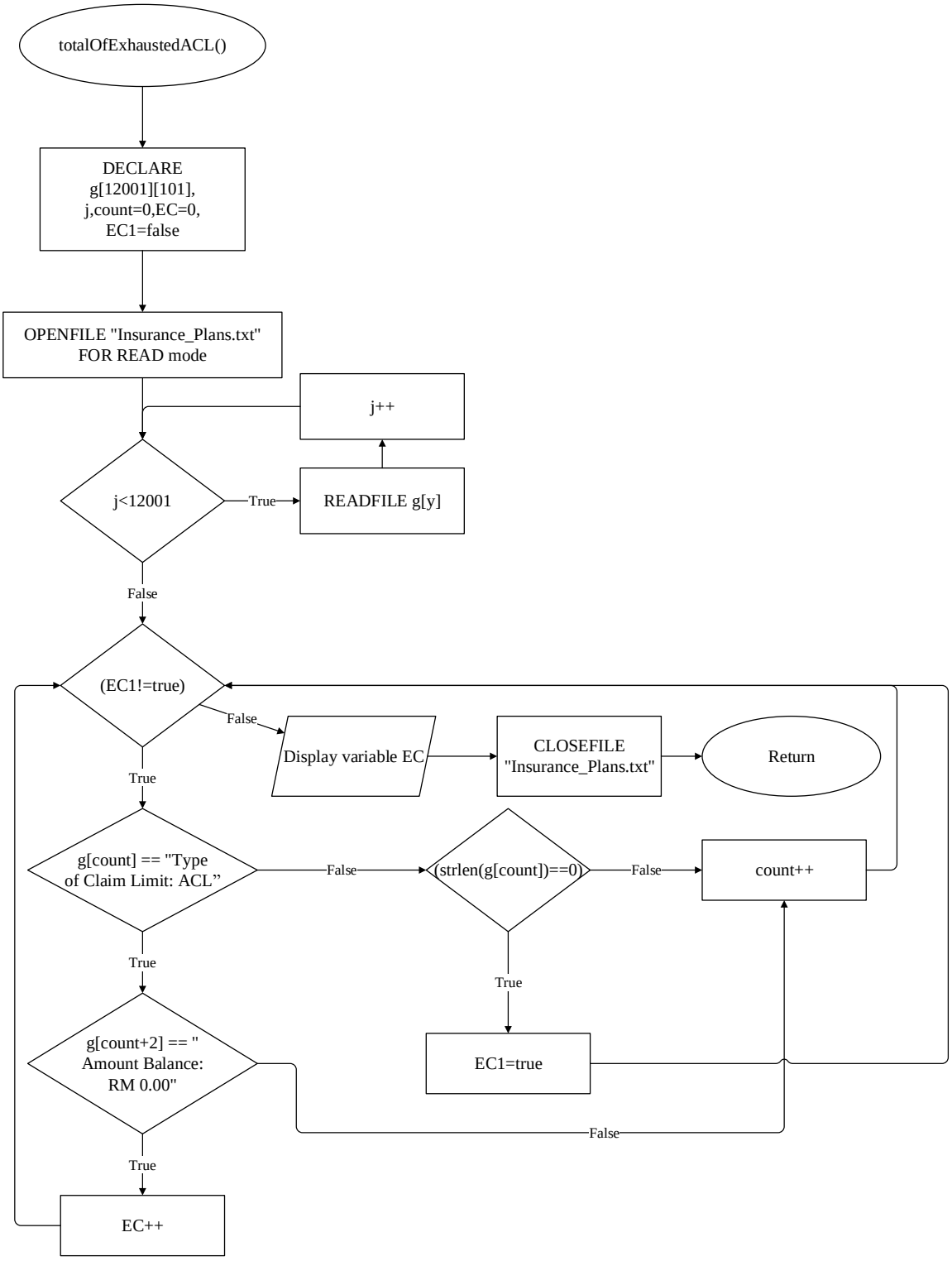
Flowchart for function displayOptions



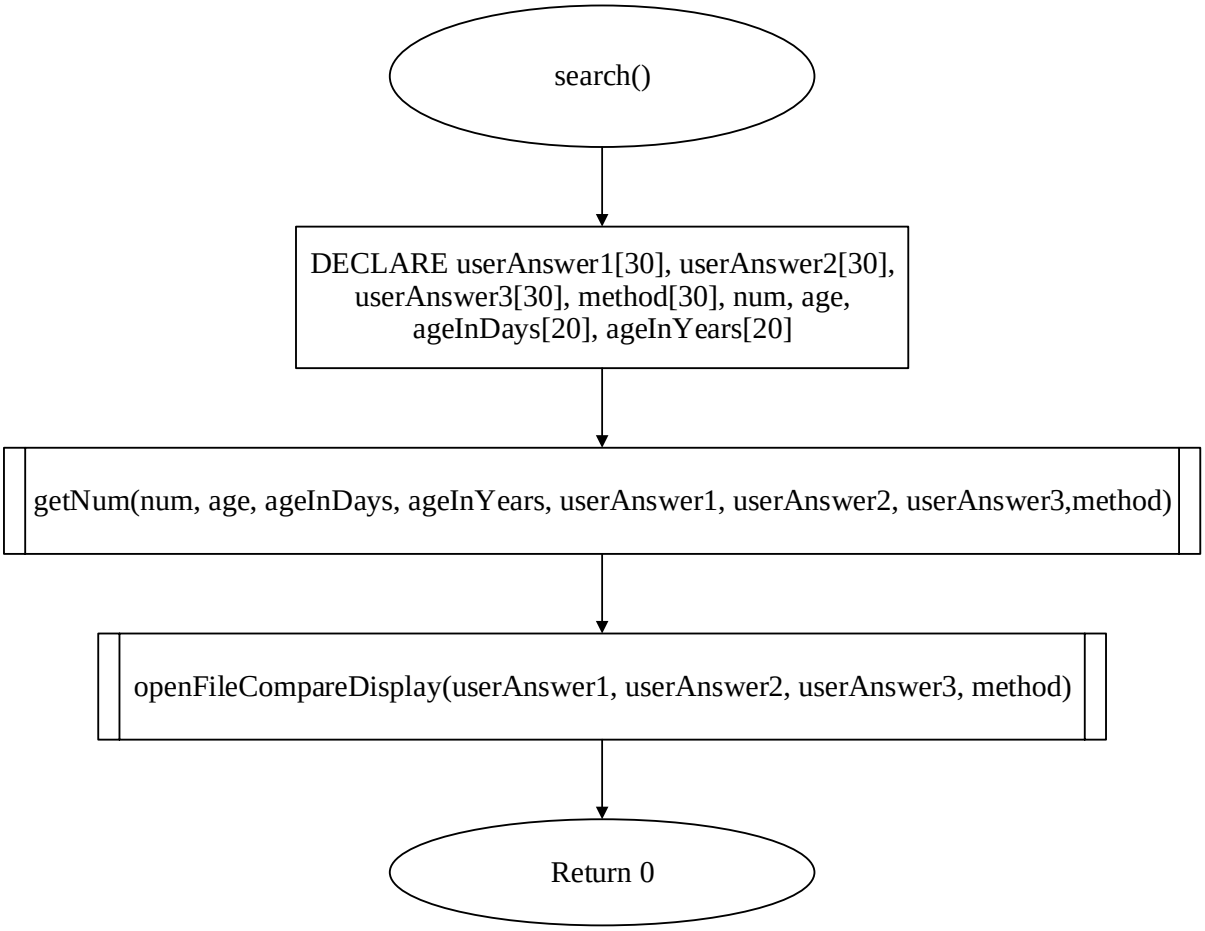
Flowchart for function displayTotalLCLAmountClaim



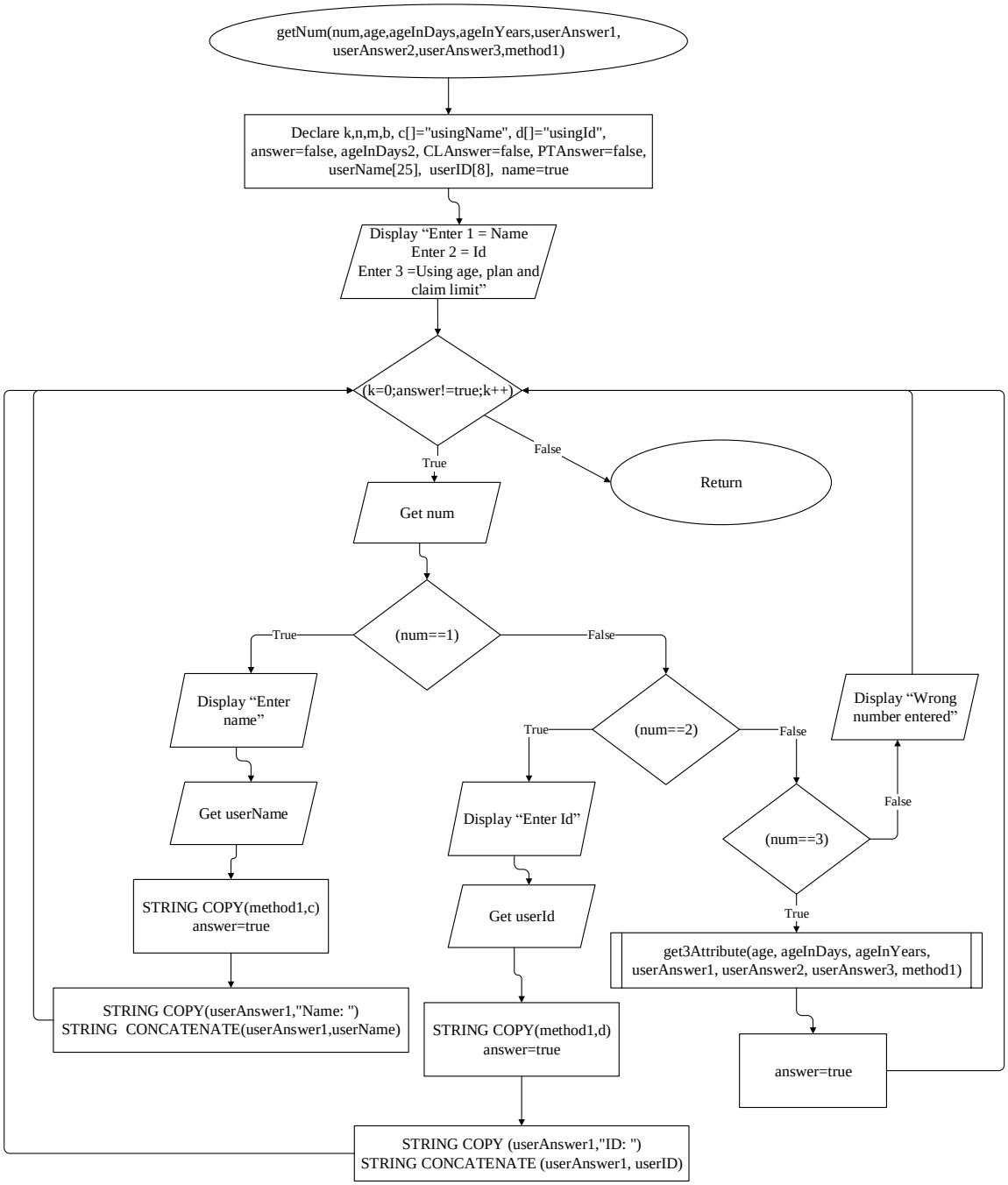
Flowchart for function totalOfExhaustedACL



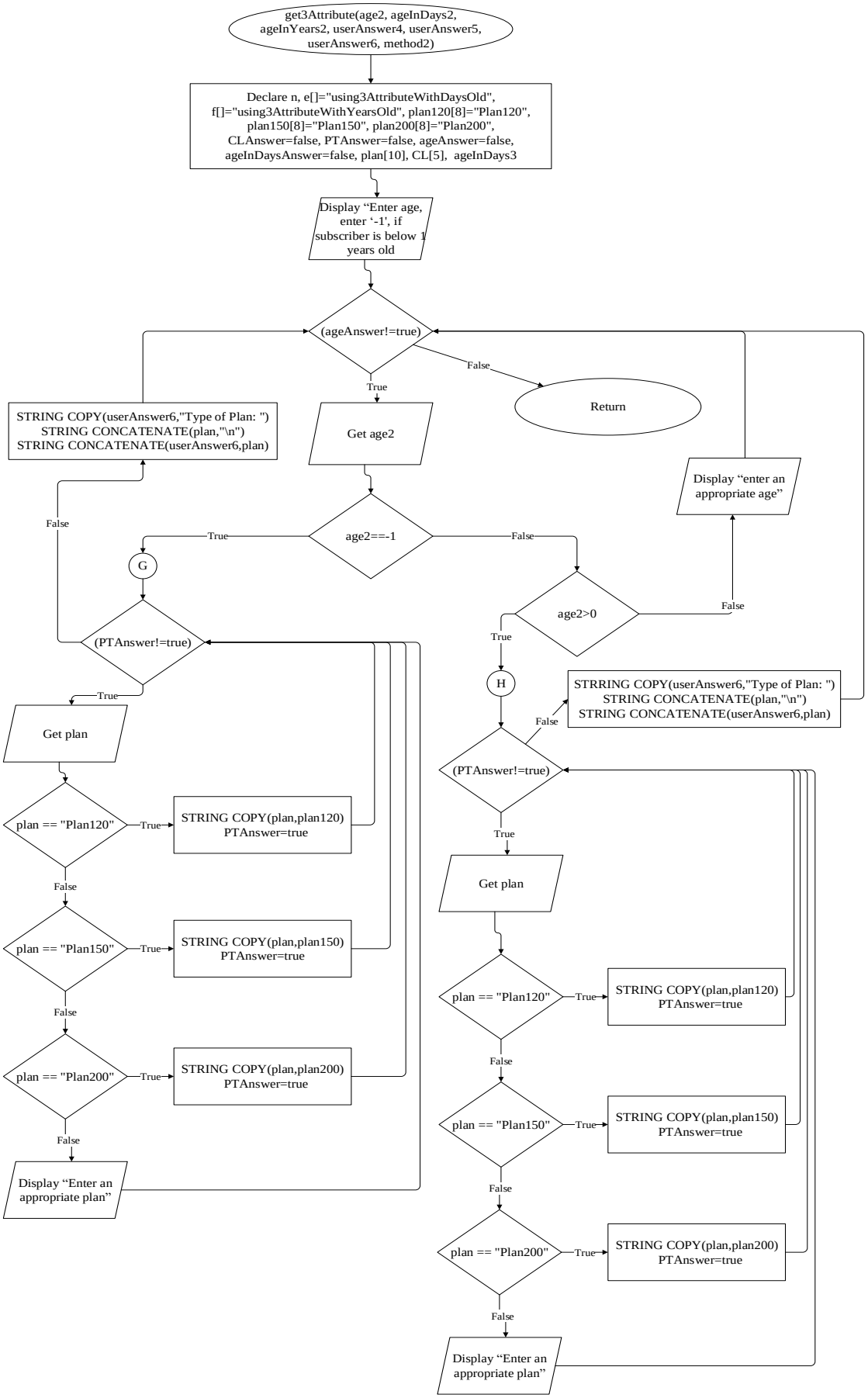
Flowchart for function search



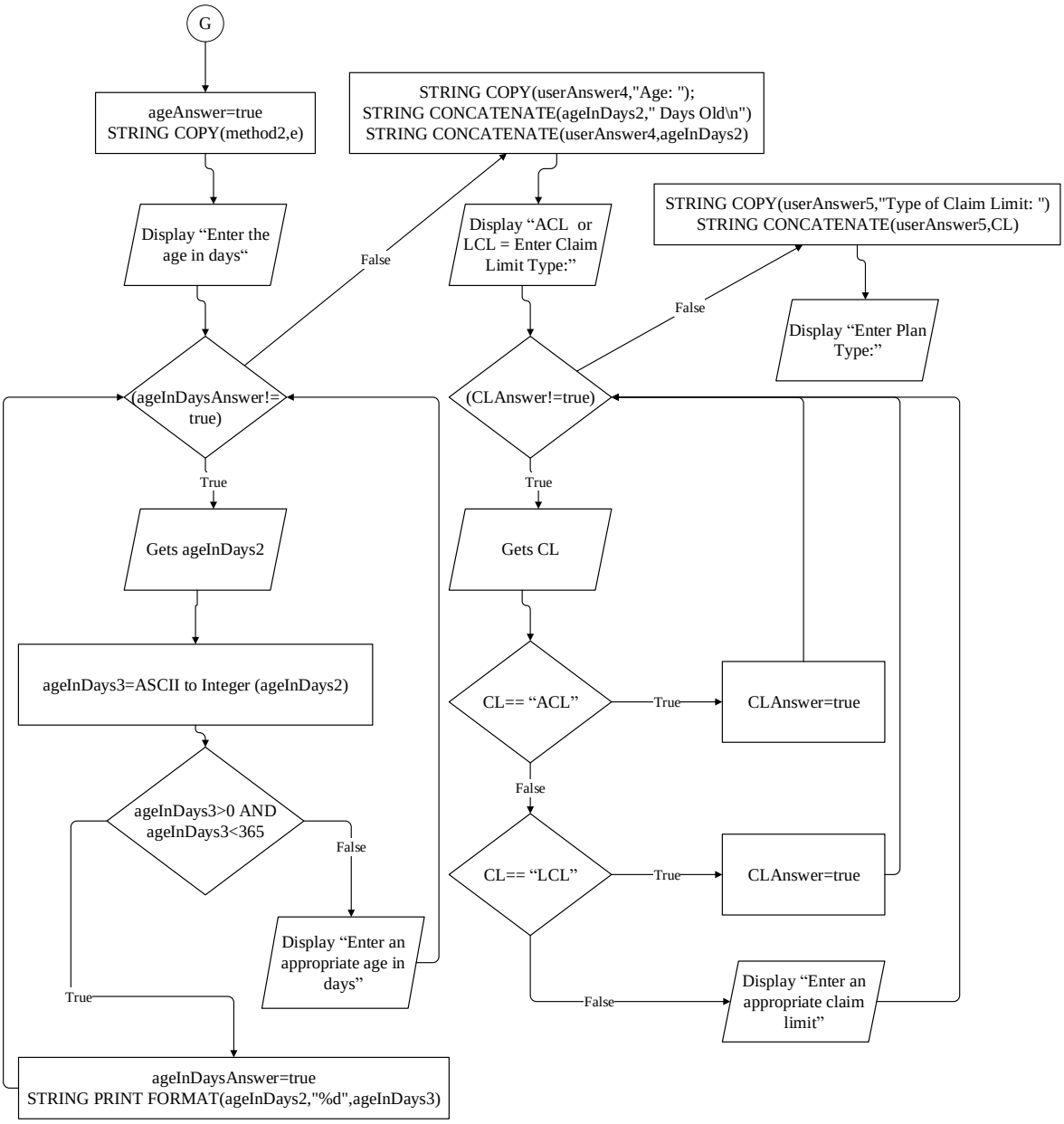
Flowchart for function getNum



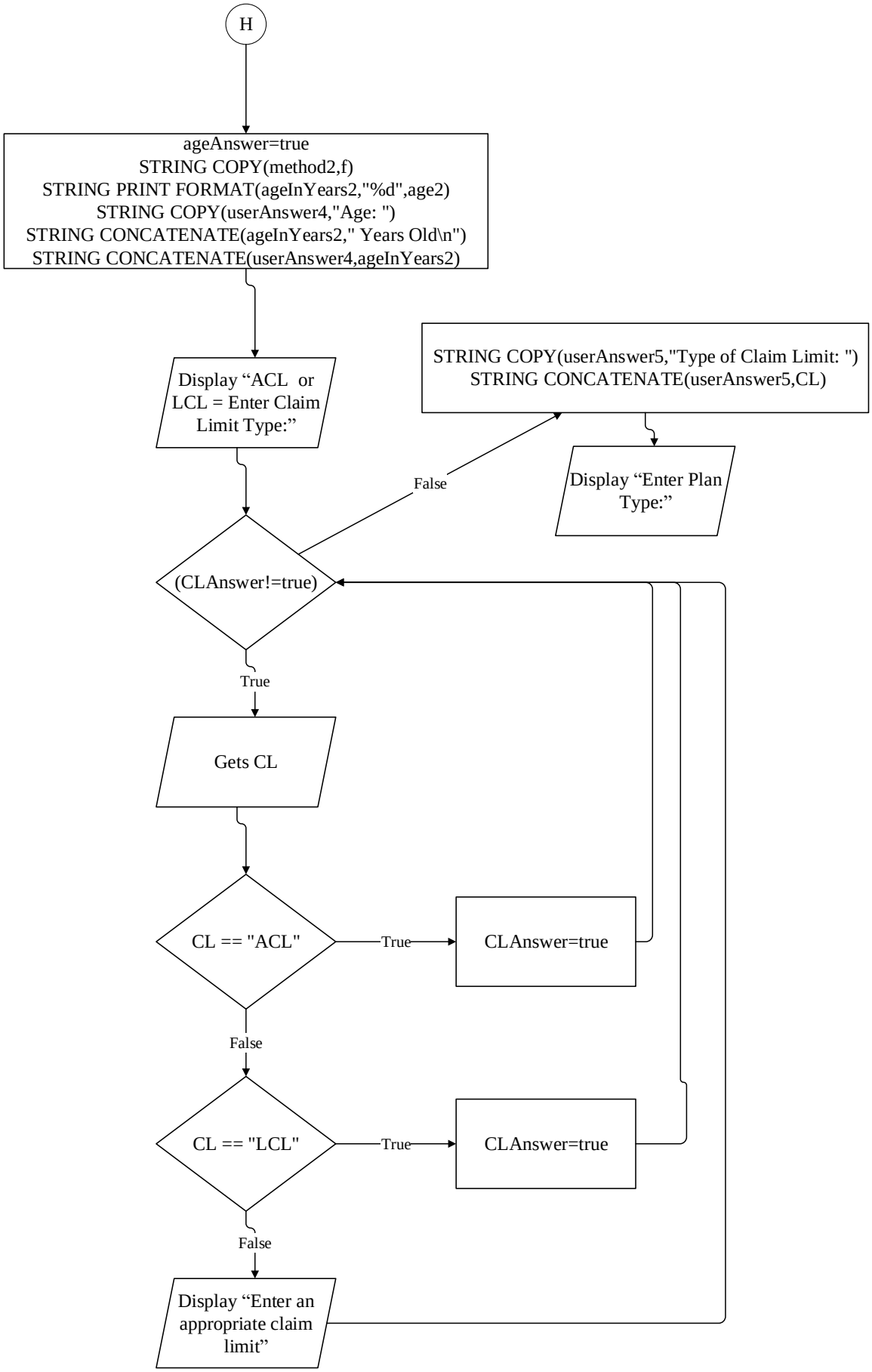
Flowchart for function get3Attribute



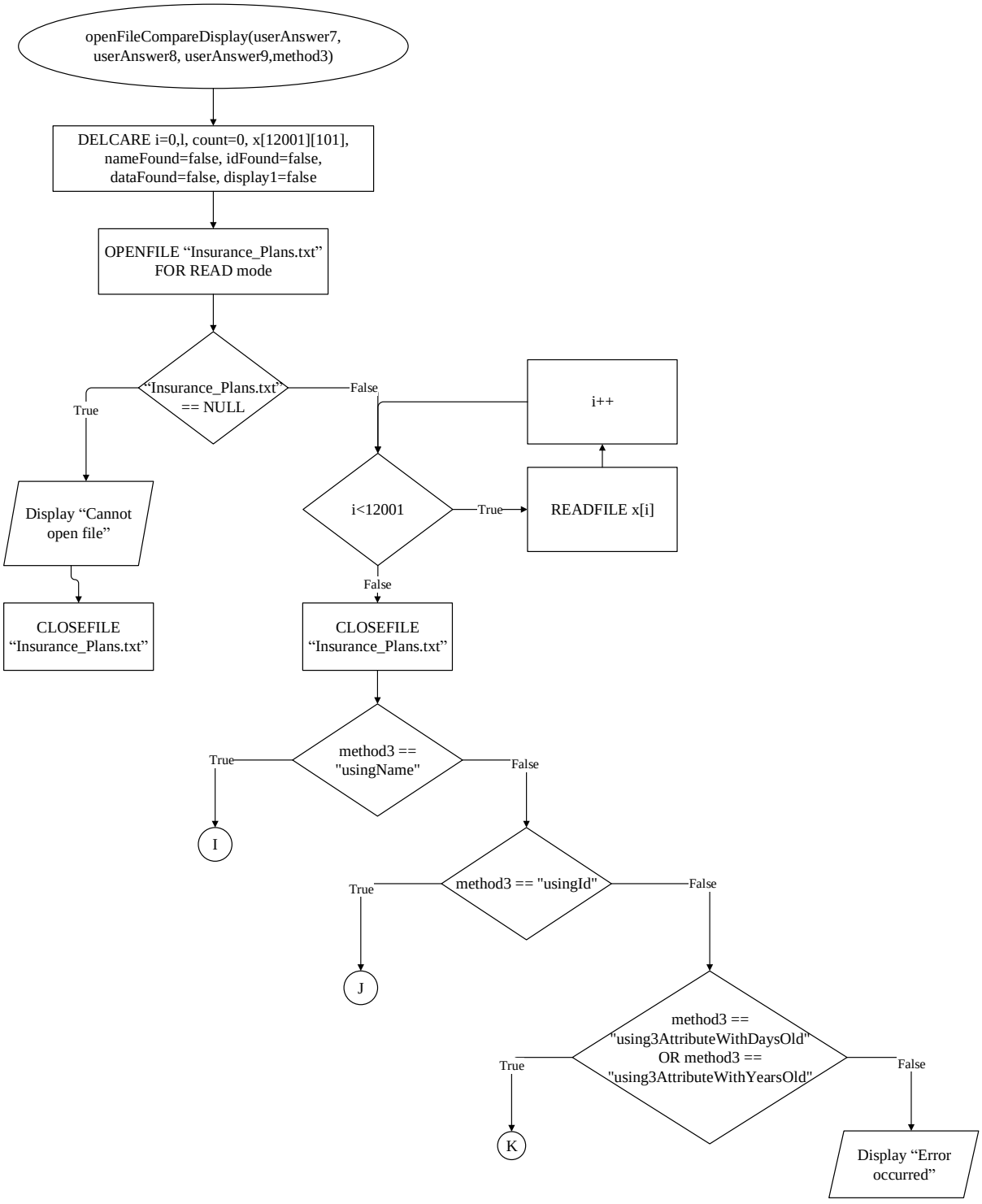
Same page connector G for function get3Attribute



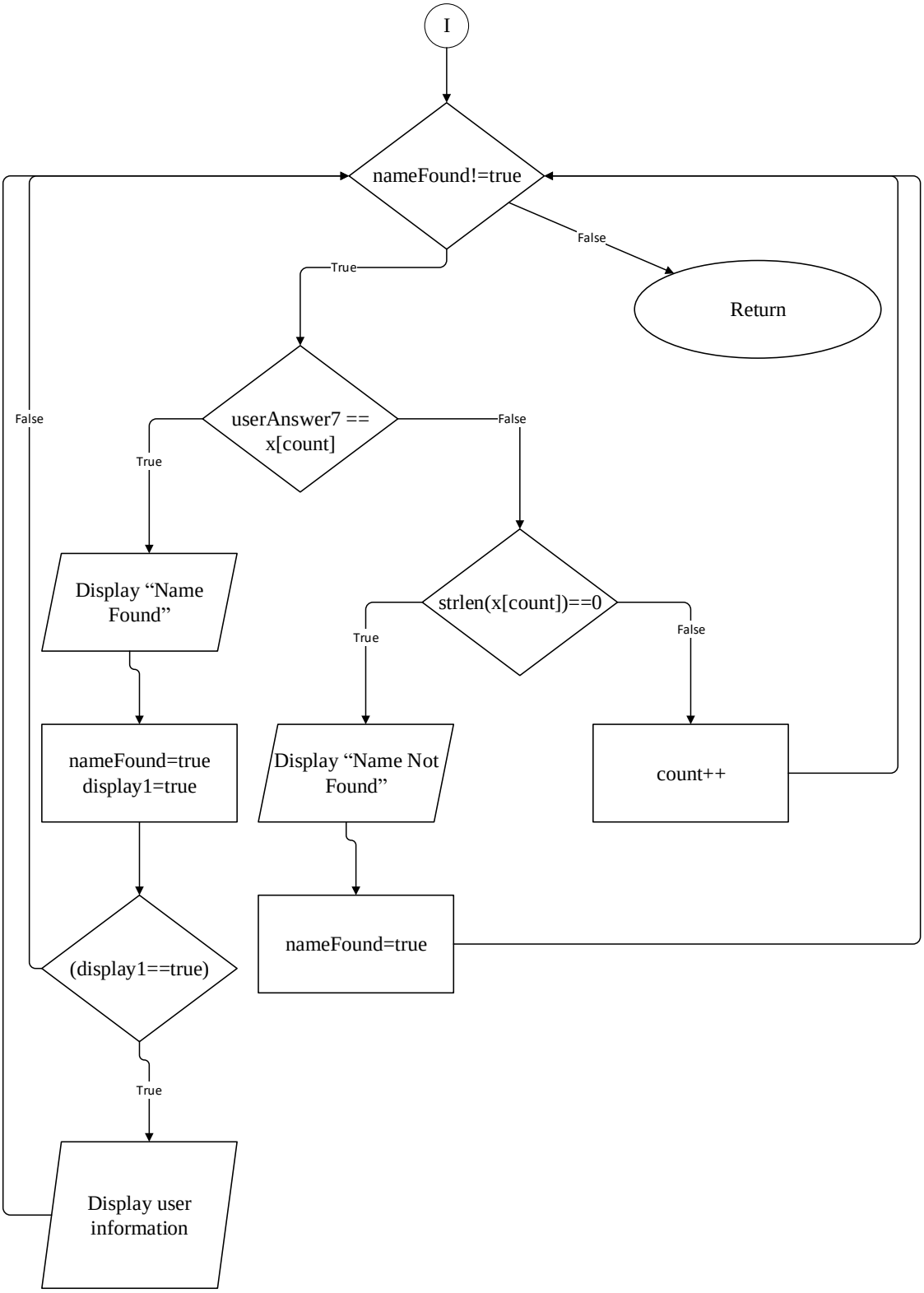
Same page connector H for function get3Attribute



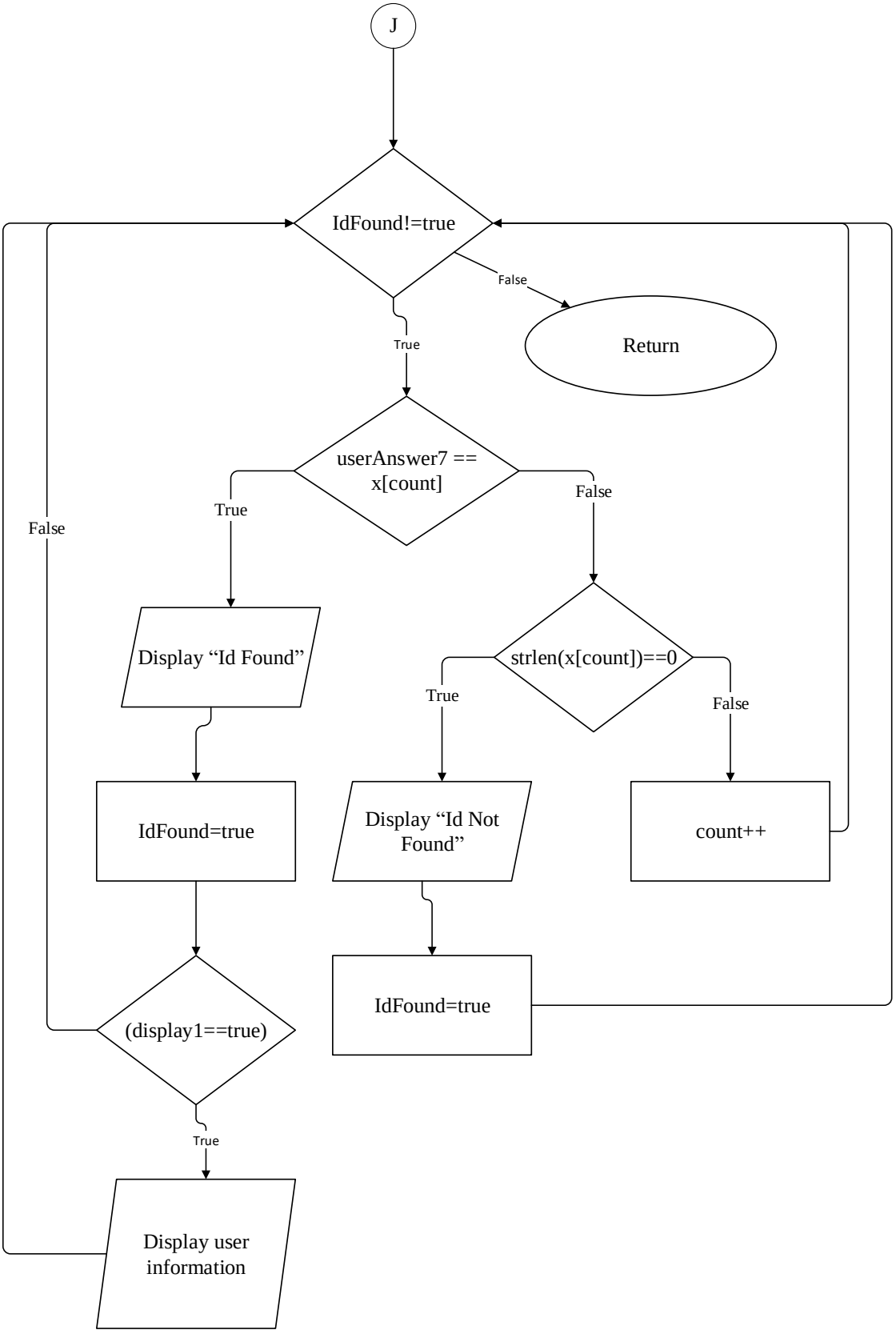
Flowchart for function openFileCompareDisplay



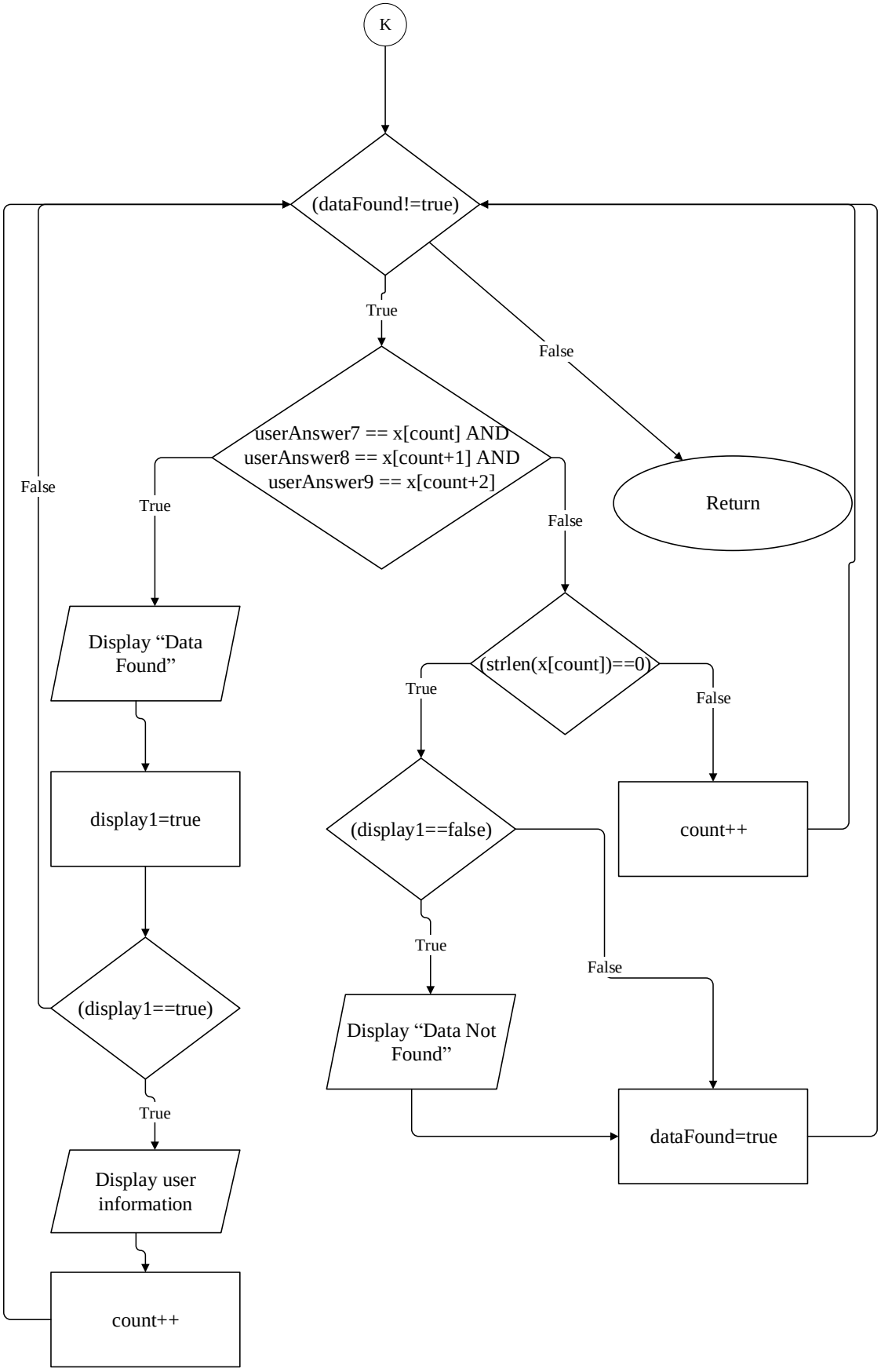
Same page connector I for function openFileCompareDisplay



Same page connector J for function openFileCompareDisplay



Same page connector K for function openFileCompareDisplay



4.0 Explanation of portable ANSI C

ANSI C or American National Standards Institute for C programming language is the successor standard of the C programming language. Although C is the oldest and most used programming language, it has its limitations. This drove the American National Standards Institute (ANSI) in 1983, formed a committee to provide a comprehensive definition to the C programming language, and this is how the new ANSI C language with better features came into existence. (Goswami, n.d.)

The new features include support for structure programming, recursion, and lexical variable scope that was not available in the previous C Language. Other improvements that were absent from previous C Languages is the inclusion of long int and unsigned int data type, the addition of compound assignment operators, standard input/output library, and the use of void data type in its library. (Goswami, n.d.)

The significant advantages of using C programming languages include:

- Many available build-in functions that are ready to use
- Fast and efficient in running programs
- A highly portable language
- Easy to compile C programs by different compilers

In C programming, it consists of 6 items in a program to execute a program. The six items are the documentation section, link section, definition section, global declaration section, main function, and executable functions outside the main function. The figure below is an example of the layout of a program. (Goswami, n.d.)

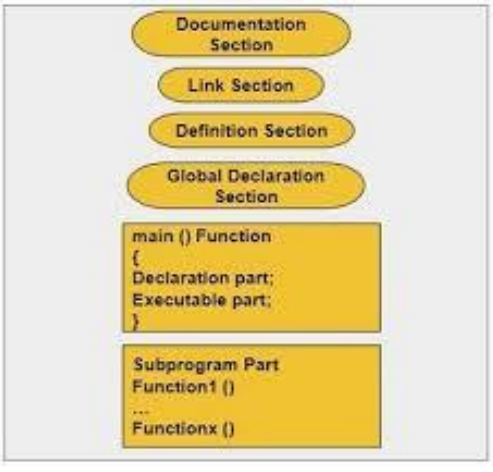


Figure 1: Structure of ANSI C Programming Language (Goswami, n.d.)

In the documentation section, the developer usually writes an introduction of the program, such as the name of the developer, name of program, file name, and copyright. The image below is an example of the use of the documentation section.

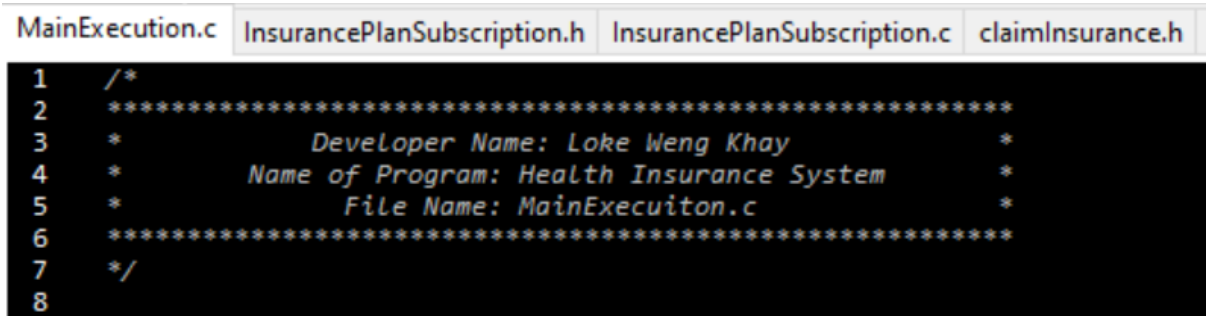


Figure 2: Example of a comment at the top of the program source code

This is an example of how a developer states all the required information before starting to do the code itself.

In the link section, usually is provide an instruction to the compiler to link functions from the system library such as standard input/output header file or “#include<stdio.h>”. The image below is an example of declaring all the header file that is needed in the “menu.c” file.

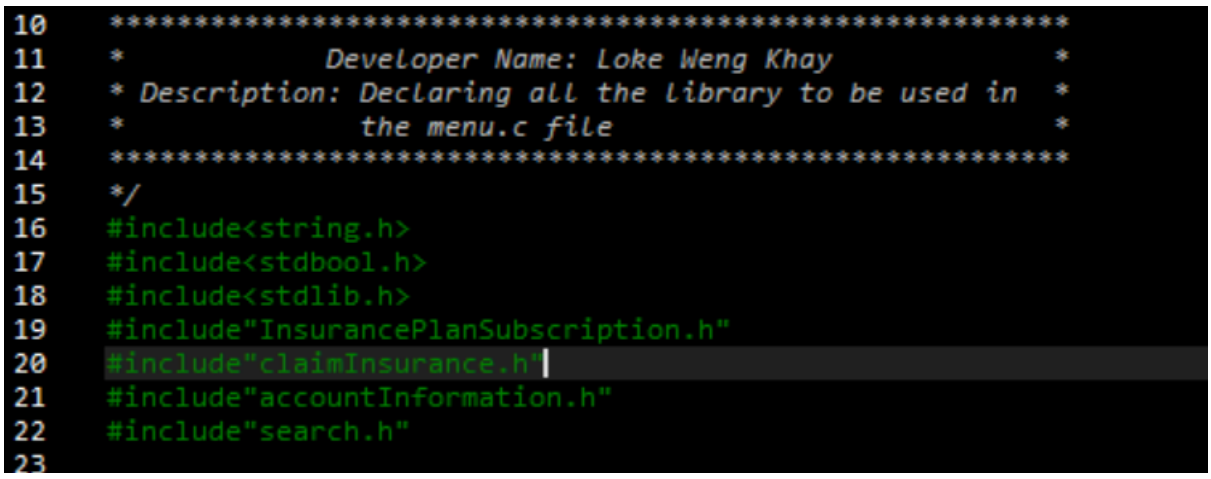


Figure 3: Example declaration of the header file that will be used in the “menu.c” file

In the definition section, all constant variables will be declared here to allow the variable to be accessed in any function, but the value of the variable will not change. The image below is an example of a defined variable in a program.

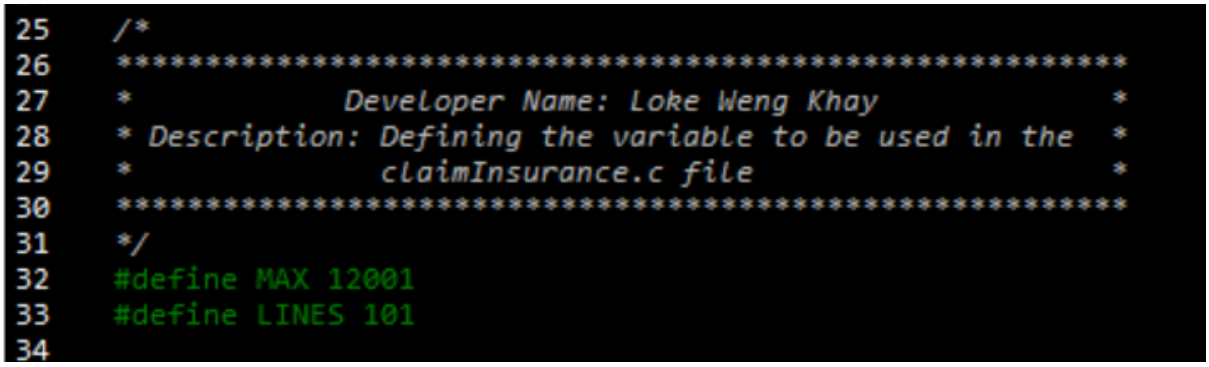


Figure 4: Defining variable MAX and LINES in the “claimInsurance.h” file

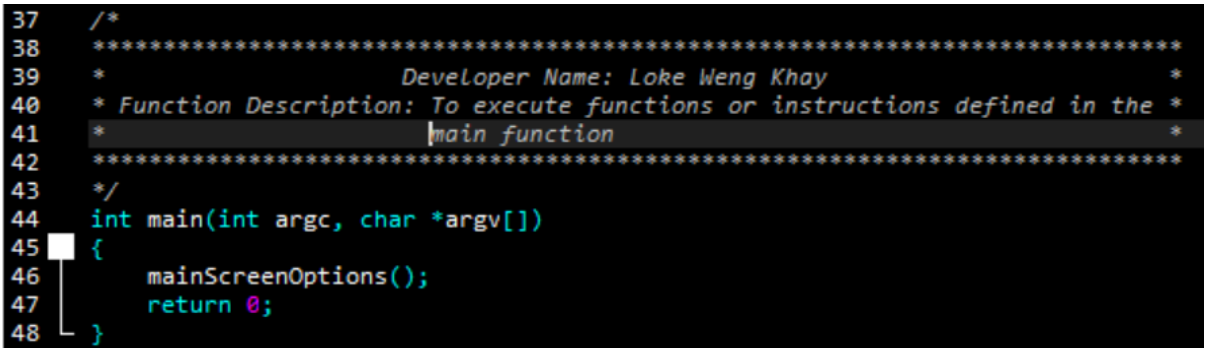
The function prototype also falls under the definition section. The function prototype usually gives information to the compiler that the function may be called in the main function. This is to prevent the compiler from getting confused when it is being called in the main function. Usually, the function prototype will give information to the compiler, such as the name of the function, type of return value, and the parameters. Figure 5 is an example of a function prototype. (Anon., n.d.)

Figure 5: Declaration of function prototype in the “InsurancePlanSubscription.c” file

In the global declaration section, variables are declared here as users will be using the variable more than once, and the variable can be accessed anywhere in the program. The developer will usually not declared a global variable, except it is mandatory. This is because the data integrity of the variable will be compromised as any function in the program itself can modify it.

In the main function, which is the most critical part of the whole program as this function executes all the other functions that are defined outside the main function will be

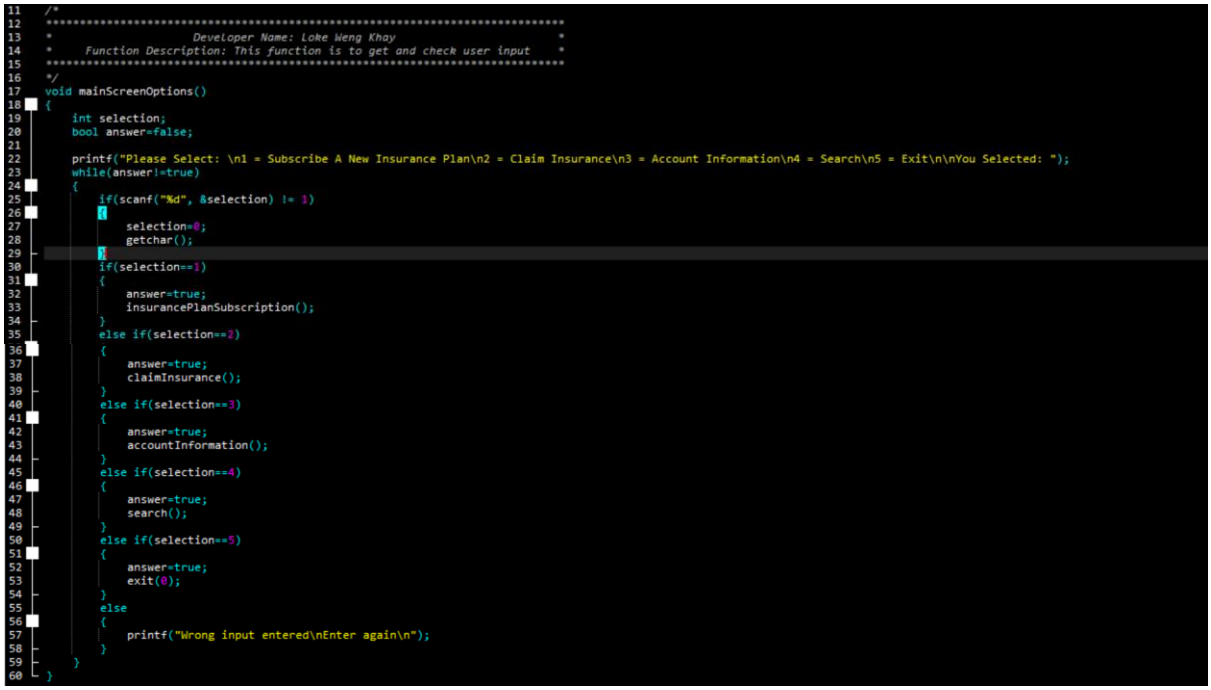
executed and to get the output. The main function is different from the other function. This is because the main function is the first code that is executed in a C program. When a program is running, the operating system passes control of the computer to the program to execute the instructions or functions defined in the main function. (Goswami, n.d.)



```
/*
*********************************************************************************
*                        Developer Name: Loke Weng Khay                         *
* Function Description: To execute functions or instructions defined in the     *
*                              main function                                    *
*********************************************************************************
*/
int main(int argc, char *argv[])
{
    mainScreenOptions();
    return 0;
}
```

Figure 6: Example of the use of the main function in “MainExecution.c” file

The last item in a program is the subprogram section. This is where a function definition is placed, which provides the actual body of the function. In the function declaration, it provides full instruction like the return type, type of parameter, and block codes to perform a specific task that is to be executed when it is called in the main function.



```
/*
*************************************************************************
*                    Developer Name: Loke Weng Khay                     *
*    Function Description: This function is to get and check user input *
*************************************************************************
*/
void mainScreenOptions()
{
    int selection;
    bool answer=false;

    printf("Please Select: \n1 = Subscribe A New Insurance Plan\n2 = Claim Insurance\n3 = Account Information\n4 = Search\n5 = Exit\n\nYou Selected: ");
    while(answer!=true)
    {
        if(scanf("%d", &selection) != 1)
        {
            selection=0;
            getchar();
        }
        if(selection==1)
        {
            answer=true;
            insurancePlanSubscription();
        }
        else if(selection==2)
        {
            answer=true;
            claimInsurance();
        }
        else if(selection==3)
        {
            answer=true;
            accountInformation();
        }
        else if(selection==4)
        {
            answer=true;
            search();
        }
        else if(selection==5)
        {
            answer=true;
            exit(0);
        }
        else
        {
            printf("Wrong input entered\nEnter again\n");
        }
    }
}
```

Figure 7: Example of a function definition “void mainScreenOption()” in “menu.c" file

5.0 Additional features

5.1 Error detection

This program has a feature added which is error detection. This is to prevent the user from entering the wrong information into the program. Error detection has been placed to avoid the user from entering wrong values such as symbols into the program. Error detection has been placed in areas like menu options, age in days, age in years, type of plan, and type of claim limit and many more. The figure below is a concept design in the flowchart on how the error detection works.

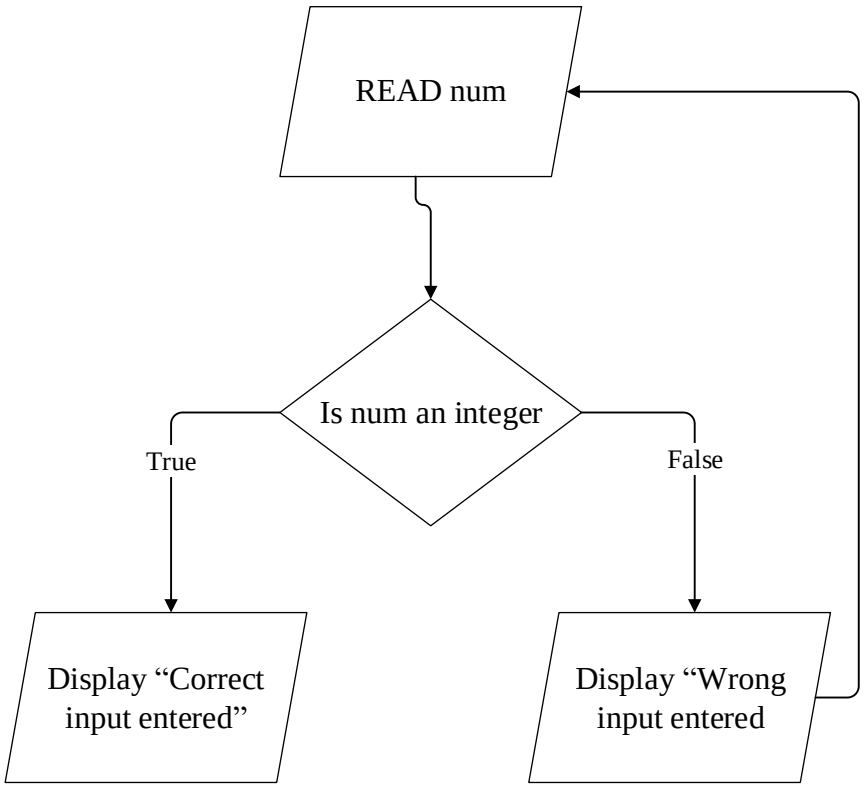


Figure 5.1.1: Concept design in the flowchart of error detection to check whether the input is an integer or not

```
23      while(answer!=true)
24      {
25          if(scanf("%d", &selection) != 1)
26          {
27              selection=0;
28              getchar();
29          }
30          if(selection==1)
31          {
32              answer=true;
33              insurancePlanSubscription();
34          }
35          else if(selection==2)
36          {
37              answer=true;
38              claimInsurance();
39          }
40          else if(selection==3)
41          {
42              answer=true;
43              accountInformation();
44          }
45          else if(selection==4)
46          {
47              answer=true;
48              search();
49          }
50          else if(selection==5)
51          {
52              answer=true;
53              exit(0);
54          }
55          else
56          {
57              printf("Wrong input entered\nEnter again\n");
58          }
59      }
```

Figure 5.1.2: Source code of error detection in the menu.c

As shown the Figure 5.1.2, the segment code inline 25 to line 29 is a conditional statement that checks the user input whether it is an integer or not. If the data is not an integer, the variable “selection” will be set to the value 0 and will loop again to ask the user to enter a correct input. The figure below is an example of the output result of error detection.

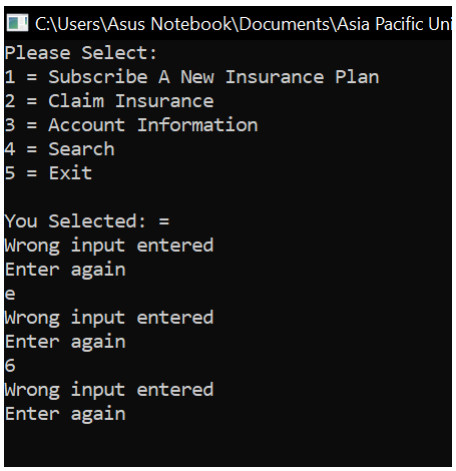


Figure 5.1.3: Error detection in progress when the user enters a character or values outside the range in menu screen options

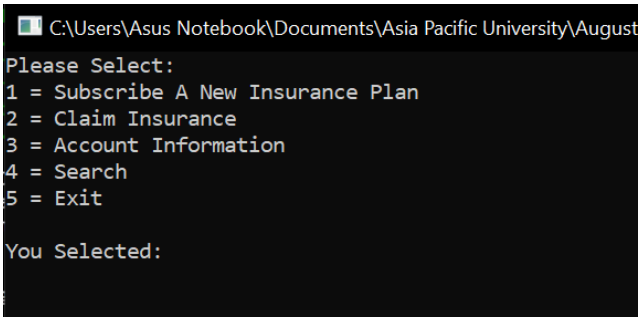
With the help of error detection, it will help to prevent the application from behaving differently than expected. This is because when a user enters a character to an integer variable, the program behaves different and makes the program to end instantly and return rubbish value. This is because the program does not know how to handle a situation like this. With added error detection, it helps to cover the loophole and increase the user-friendliness and satisfaction of this program.

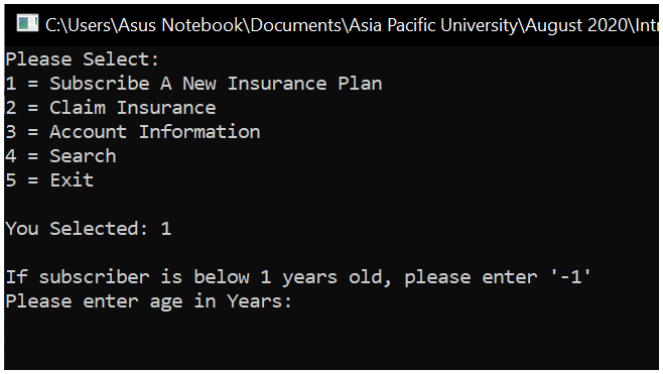
6.0 Test specification table

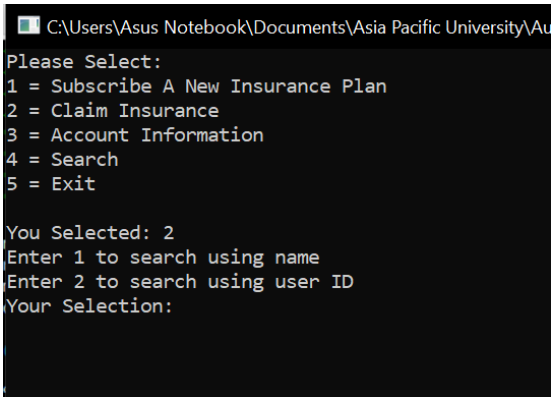
No.	Test Objective	Status
1.	To check whether the program can run without error and can display all the menu options	Success
2.	To check if the “insurancePlanSubscription” function is working or not when the user enters “1”	Success
3.	To check if the “claimInsurance” function is working or not when the user enters “2”	Success
4.	To check if the “accountInformation” function is working or not when the user enters “3”	Success
5.	To check if the “search” function is working or not when the user enter “4”	Success
6.	To check if the “Exit” function is working or not when the user enter “5”	Success
7.	To check if the program will prompt the user to enter again if the user enters the wrong number in the menu screen	Success
8.	To check whether the program is able to collect all information entered by the user in the “insurancePlanSubscription” function and display the data inserted to the user	Success
9.	To check whether the program is able to collect all information entered by the user in the “insurancePlanSubscription” function and store the data in a text file	Success
10.	To check if the program will prompt the user to enter again in “claimInsurance” function if the user enters the wrong number or symbols	Success
11.	To check whether the program is able to collect all information entered by the user in the “claimInsurance” function and display the data inserted to the user	Success
12.	To check whether the program is able to collect all information entered by the user in the “claimInsurance” function and store the data in a text file name “Insurance_Claims.txt”	Success
13.	To check if the program will prompt the user to enter again in “accountInformation” function if the user enters the wrong number or symbols	Success
14.	To check whether the program is able to read data from a text file which is “Insurance_Plans.txt” and display the total amount claimed by Lifetime Claim Limit user in the “accountInformation” function	Success
15.	To check whether the program is able to read data from a text file, which is “Insurance_Plans.txt” and display the number of Annual Claim Limit users that exhausted their amount balance in the “accountInformation” function	Success
16.	To check if the program will prompt the user to enter again in the “search” function if the user enters the wrong number or symbols	Success
17.	To check whether the program is able to read data from a text file, which is “Insurance_Plans.txt”, and find a particular subscriber that the user is looking for using the name entered by the user and displaying the subscriber info once it is found.	Success
18.	To check whether the program is able to read data from a text file, which is “Insurance_Plans.txt”, and find a particular subscriber that the user is looking for using the id entered by the user and displaying the subscriber info once it is found.	Success

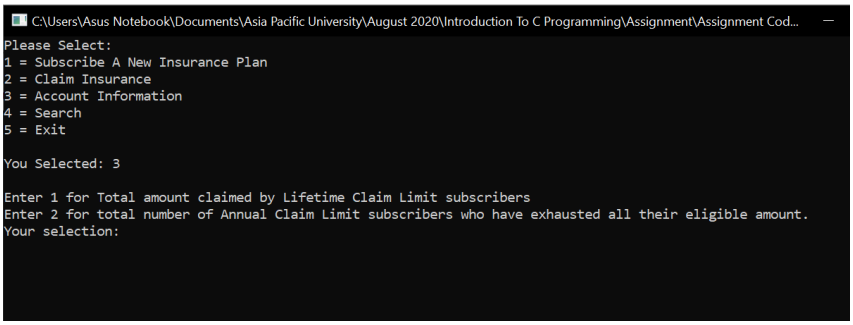
19.	To check whether the program is able to read data from a text file which is "Insurance_Plans.txt", and find a particular subscriber that the user is looking for using the combination of age, plan type, and type of claim limit entered by the user and displaying the subscriber info, once it is found.	Success

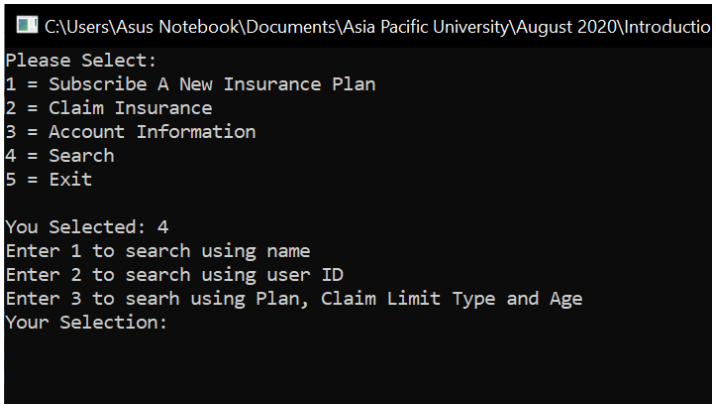
6.1 Test and Result

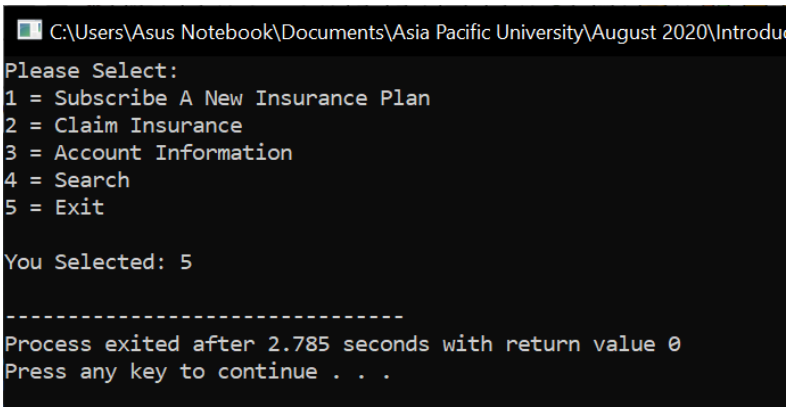
Test Case	1
Test Objective	To check whether the program can run without error and can display all the menu options
Function	mainScreenOptions()
Input	-
Expected Output	The program runs and displays the menu screen
Output	
Description of Output	The program is able to execute the "mainScreenOptions" function without any errors

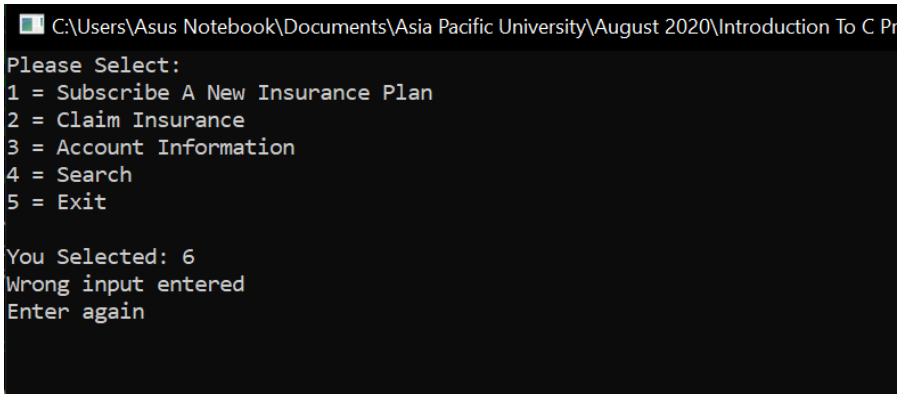
Test Plan	2
Test Objective	To check if the "Insurance Subscription" function is working or not when the user enters "1"
Function	insurancePlanSubscription()
Input	1
Expected Output	The program runs successfully and prompts the user to enter information to subscribe for an insurance
Output	
Description of Output	The program executes the "insurancePlanSubscription" function when it is called

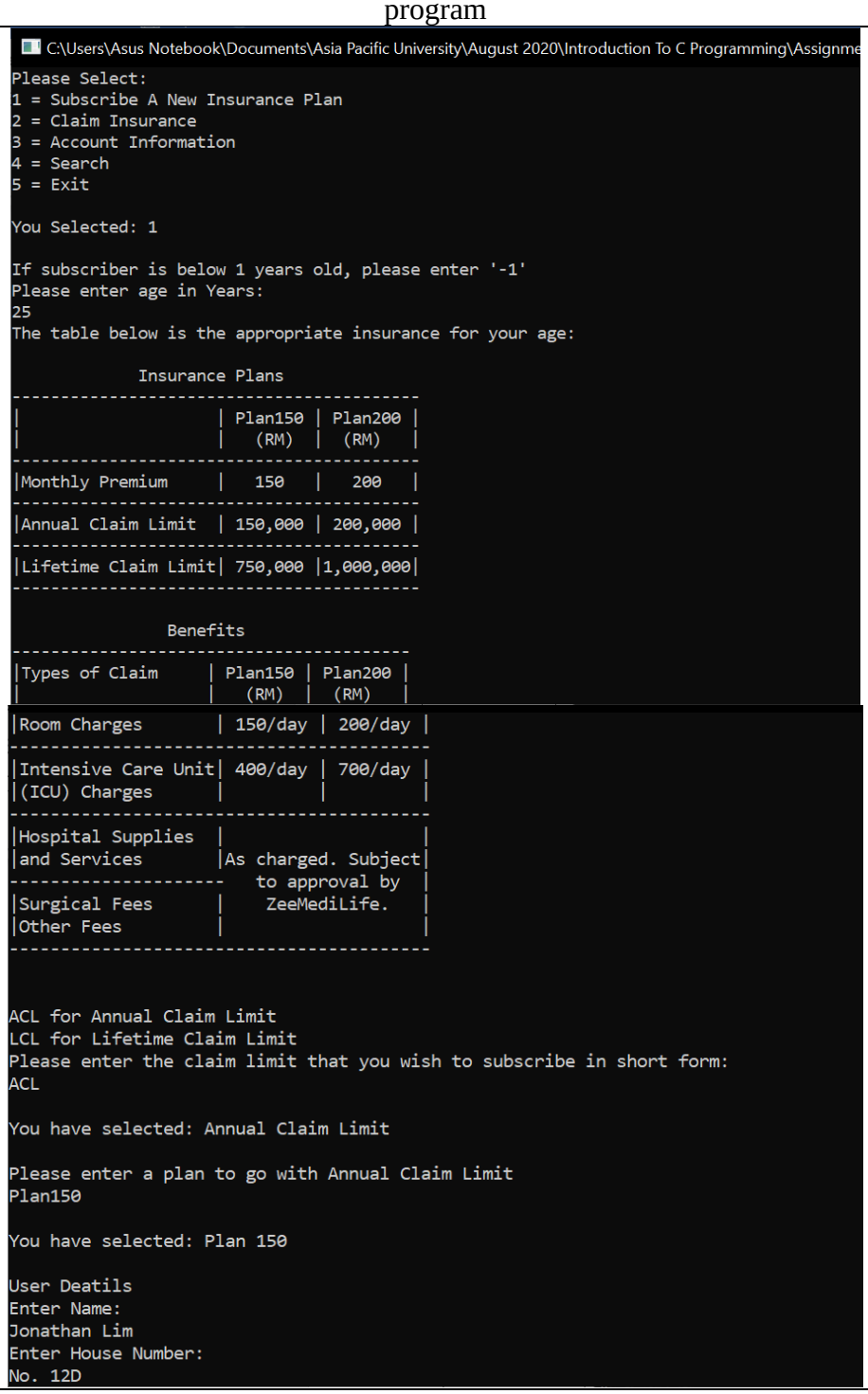
Test Plan	3
Test Objective	To check if the “claimInsurance” function is working or not when the user enters “2”
Function	claimInsurance()
Input	2
Expected Output	The program should be able to call the “claimInsurance” function
Output	
Description of Output	The program executes the “claimInsurance” function when it is called

Test Plan	4
Test Objective	To check if the “accountInformation” function is working or not when the user enters “3”
Function	accountInformation()
Input	3
Expected Output	The program should be able to call the “accountInformation” function
Output	
Description of Output	The program executes the “accountInformation” function when it is called

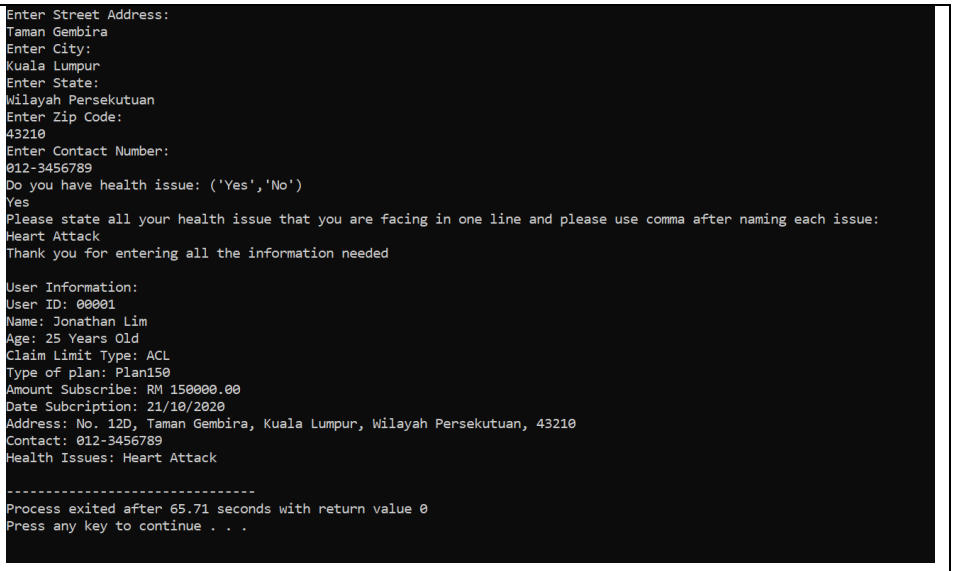
Test Plan	5
Test Objective	To check if the “search” function is working or not when the user enter “4”
Function	search()
Input	4
Expected Output	The program should be able to call the “search” function
Output	
Description of Output	The program executes the “search” function when it is called

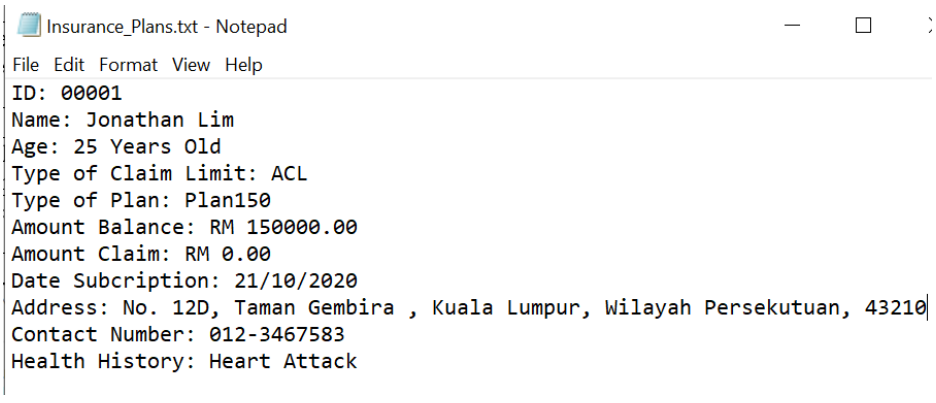
Test Plan	6
Test Objective	To check if the “Exit” function is working or not when the user enter “5”
Function	exit(0)
Input	5
Expected Output	The program should end the program without any error
Output	
Description of Output	The program automatically exit the program without any errors

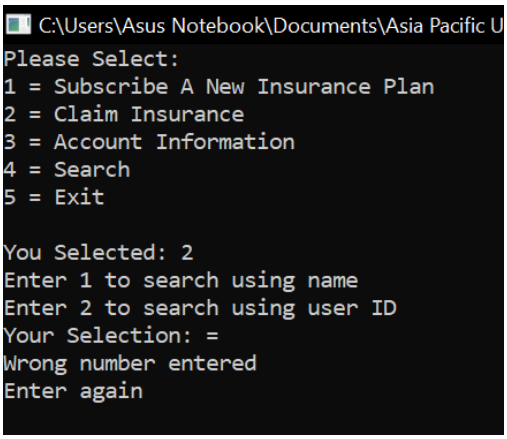
Test Plan	7
Test Objective	To check if the program will prompt the user to enter again if the user enters the wrong number
Function	mainScreenOptions()
Input	6
Expected Output	The program should prompt the user to enter again
Output	
Description of Output	The program was able to prompt the user that the user has entered a wrong input and ask the user to enter again

Test Plan	8
Test Objective	To check whether the program is able to collect all information entered by the user in the “insurancePlanSubscription” function and display the data inserted to the user
Function	insurancePlanSubscription()
Inputs	25, ACL, Plan150, Jonathan Lim, 12D, Taman Gembira, Kuala Lumpur, Wilayah Persekutuan, 43210, 012-3456789, Yes, Heart Attack
Expected Output	The program collects all user information and displays at the end of the program
Output	

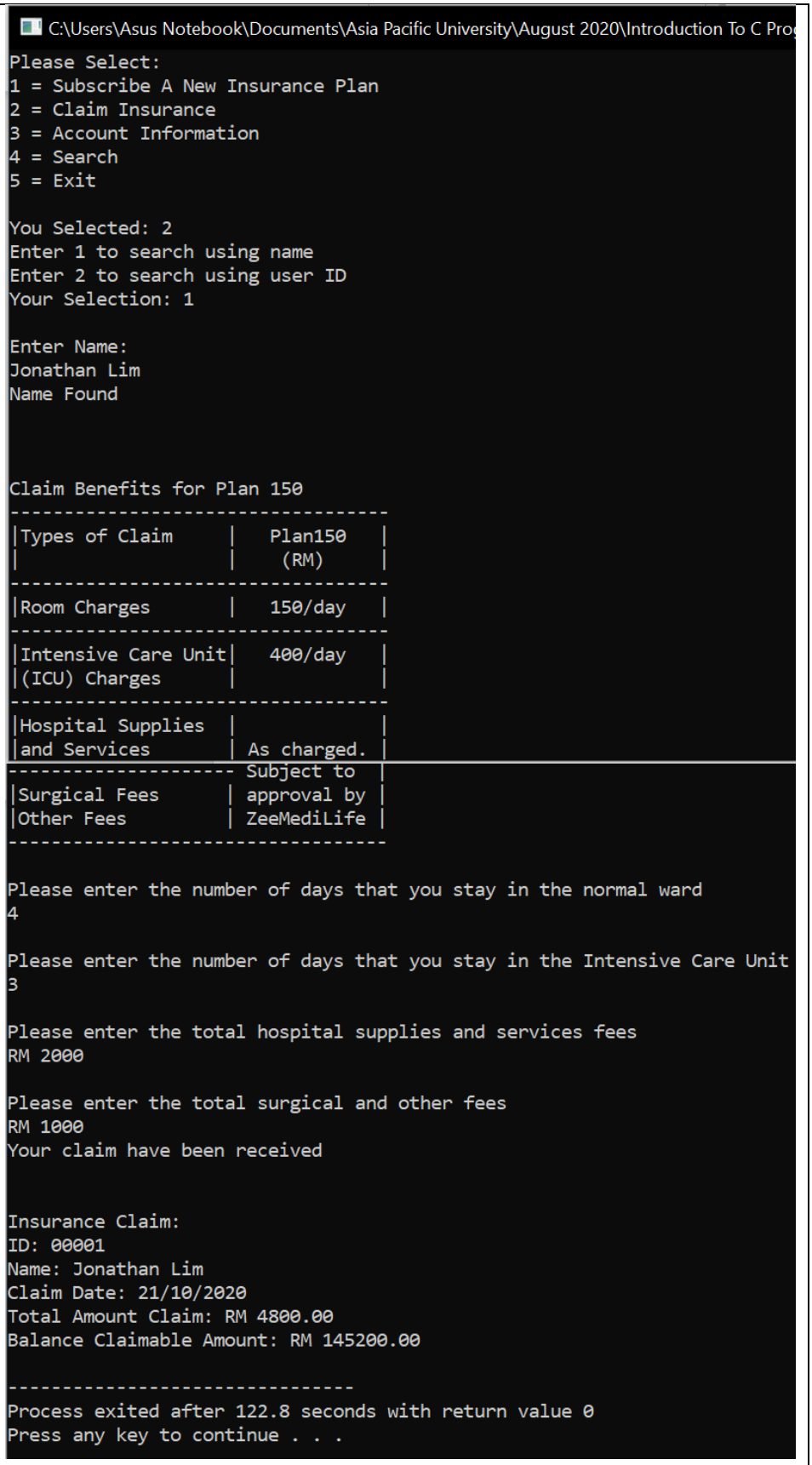
```
C:\Users\Asus Notebook\Documents\Asia Pacific University\August 2020\Introduction To C Programming\Assignme
Please Select:
1 = Subscribe A New Insurance Plan
2 = Claim Insurance
3 = Account Information
4 = Search
5 = Exit

You Selected: 1

If subscriber is below 1 years old, please enter '-1'
Please enter age in Years:
25
The table below is the appropriate insurance for your age:

            Insurance Plans
-----------------------------------------
|                   | Plan150 | Plan200 |
|                   |   (RM)  |  (RM)   |
-----------------------------------------
|Monthly Premium    |   150   |   200   |
-----------------------------------------
|Annual Claim Limit | 150,000 | 200,000 |
-----------------------------------------
|Lifetime Claim Limit| 750,000 |1,000,000|
-----------------------------------------

               Benefits
-----------------------------------------
|Types of Claim     | Plan150 | Plan200 |
|                   |   (RM)  |  (RM)   |
-----------------------------------------
|Room Charges       | 150/day | 200/day |
-----------------------------------------
|Intensive Care Unit| 400/day | 700/day |
|(ICU) Charges      |         |         |
-----------------------------------------
|Hospital Supplies  |                   |
|and Services       |As charged. Subject|
--------------------   to approval by   |
|Surgical Fees      |    ZeeMediLife.   |
|Other Fees         |                   |
-----------------------------------------


ACL for Annual Claim Limit
LCL for Lifetime Claim Limit
Please enter the claim limit that you wish to subscribe in short form:
ACL

You have selected: Annual Claim Limit

Please enter a plan to go with Annual Claim Limit
Plan150

You have selected: Plan 150

User Deatils
Enter Name:
Jonathan Lim
Enter House Number:
No. 12D
```

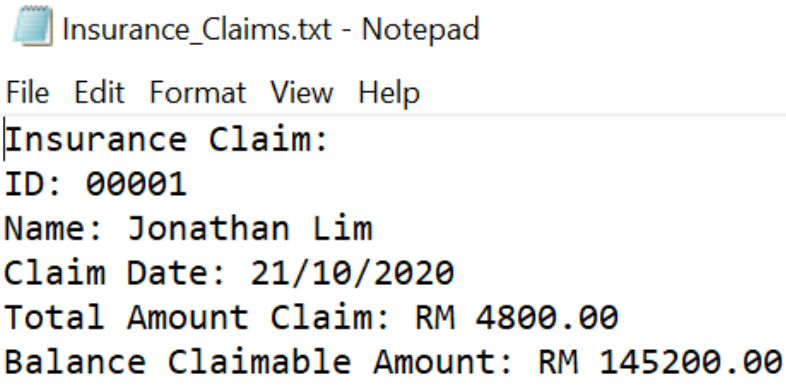
	
Description of Output	The program is able to collect all the user's data and display it out to the user

Test Plan	9
Test Objective	To check whether the program is able to collect all information entered by the user in the “insurancePlanSubscription” function and store the data in a text file
Function	insurancePlanSubscription()
Input	25, ACL, Plan150, Jonathan Lim, 12D, Taman Gembira, Kuala Lumpur, Wilayah Persekutuan, 43210, 012-3467583, Yes, Heart Attack
Expected Output	The program collects all user information and stores the data in the “Insurance_Plans.txt” file
Output	
Description of Output	The program is able to collect all the user's data and store it in the text file called “Insurance_Plans.txt”

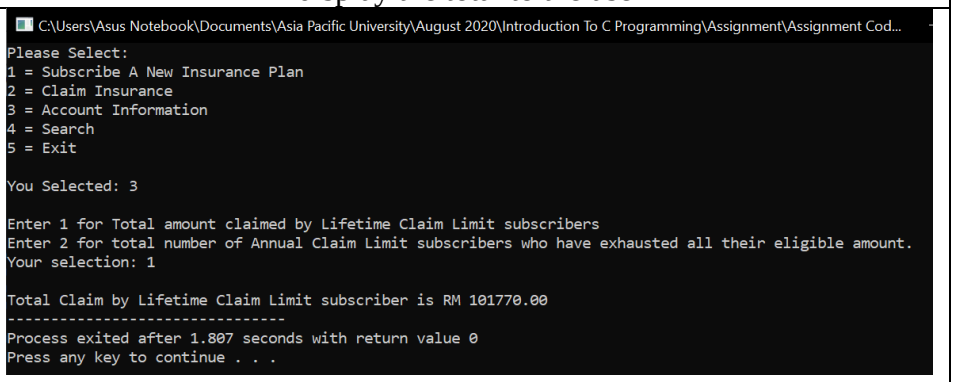
Test Plan	10
Test Objective	To check if the program will prompt the user to enter again in "claimInsurance" function if the user enters the wrong number or symbols
Function	claimInsurance()
Input	=
Expected Output	The program should prompt the user to enter again
Output	
Description of Output	The program was able to prompt the user that the user has entered a wrong input and ask the user to enter again

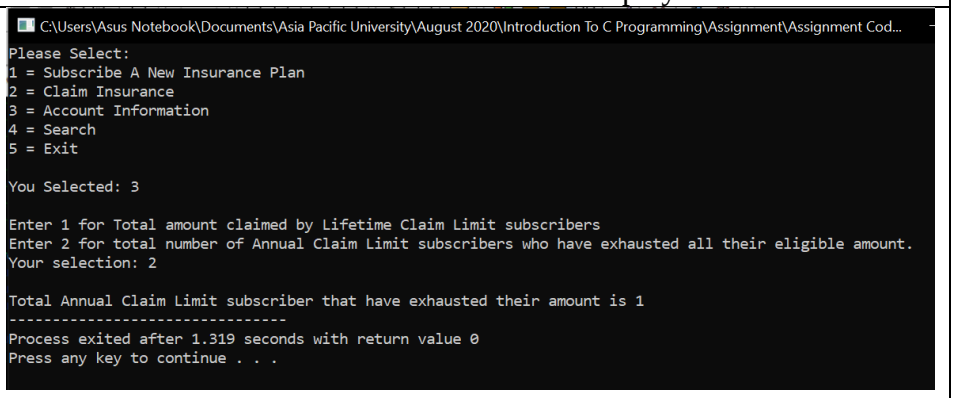
Test Plan	11
Test Objective	To check whether the program is able to collect all information entered by the user in the "claimInsurance" function and display the data inserted to the user
Function	claimInsurance()
Input	1, Jonathan Lim, 4, 4, 3000, 2000 Or 2, 00001, 5, 4, 3000, 2000
Expected Output	The program collects all user information and displays at the end of the program

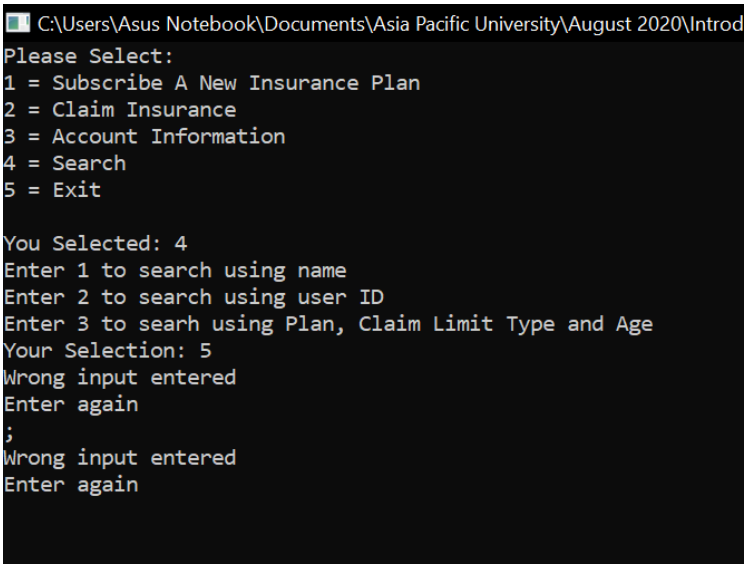
Output	
Description of Output	The program was able to collect all the information from the user, calculate the total and display out the total requested claim made by the user

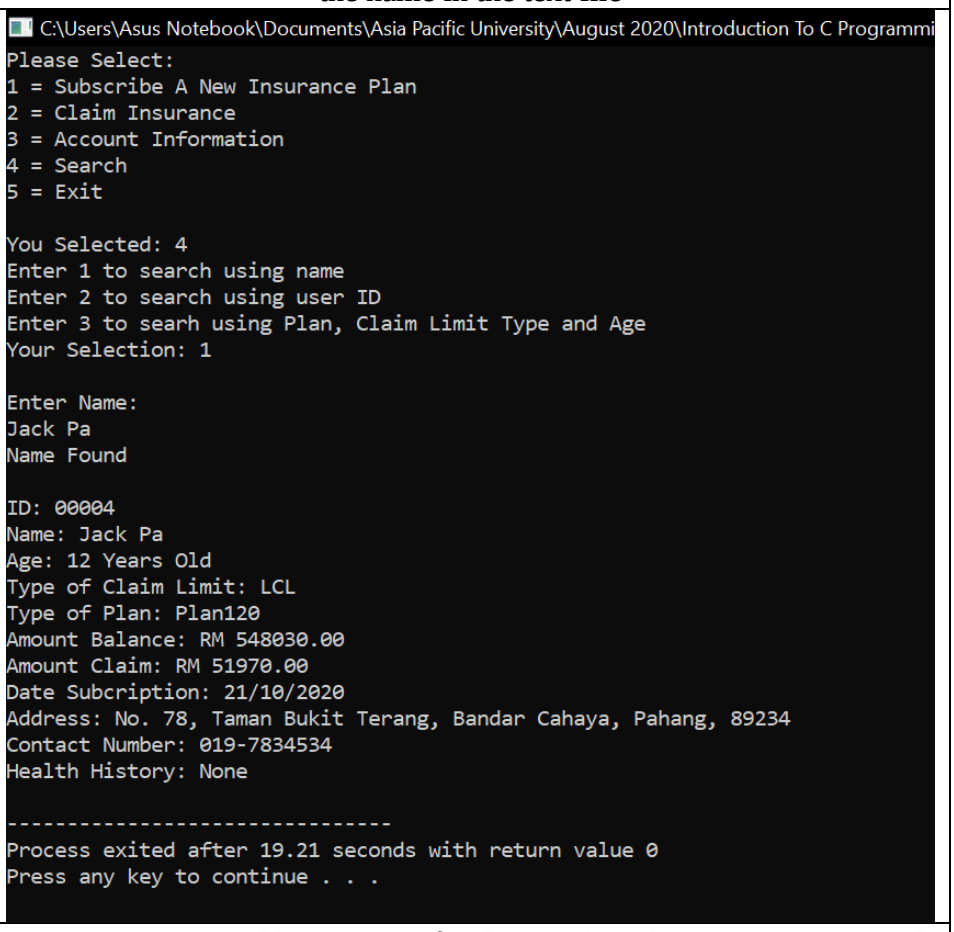
Test Plan	12
Test Objective	To check whether the program is able to collect all information entered by the user in the “claimInsurance” function and store the data in a text file name as “Insurance_Claims.txt”
Function	claimInsurance()
Input	1, Jonathan Lim, 4, 3, 2000, 1000 Or 2, 00001, 4, 3, 2000, 1000
Expected Output	The program collects all user information and stores the data in the text file named “Insurance_Claims.txt”
Output	
Description of Output	The program is able to collect all the user's data, calculate the total requested claim and store the data in the text file called “Insurance_Claims.txt”

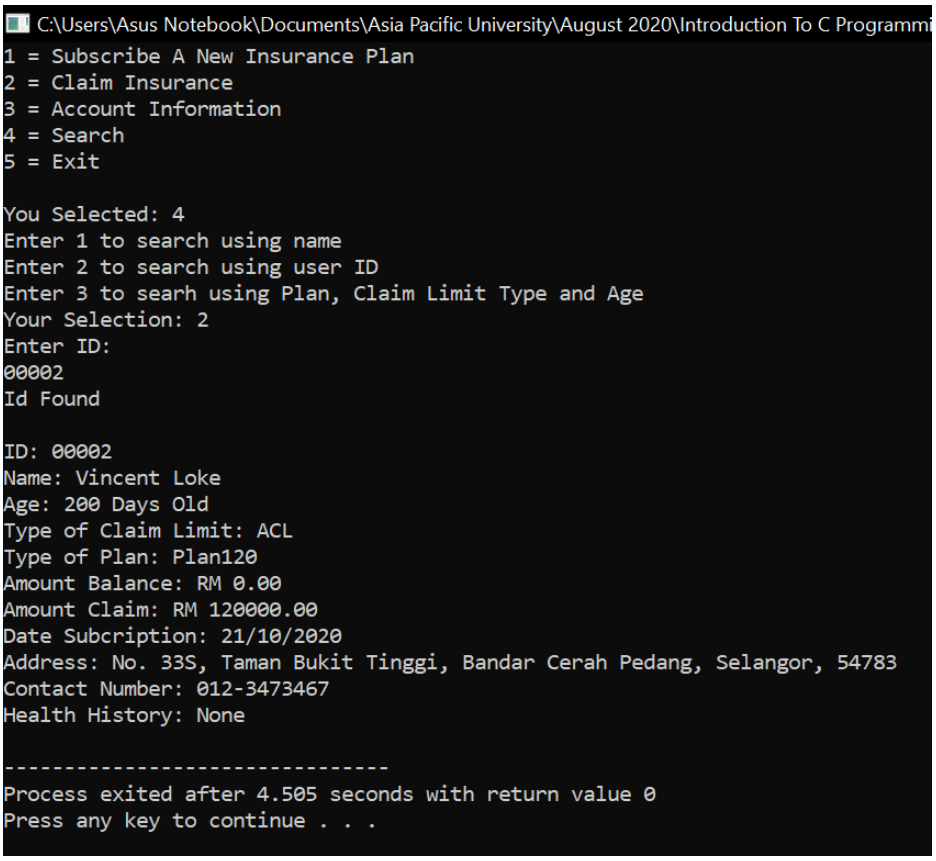
Test Plan	13
Test Objective	To check if the program will prompt the user to enter again in “accountInformation” function if the user enters the wrong number or symbols
Function	accountInformation()
Input	4 OR [
Expected Output	The program should prompt the user to enter again
Output	
Description of Output	The program was able to prompt the user that the user has entered a wrong input and ask the user to enter again

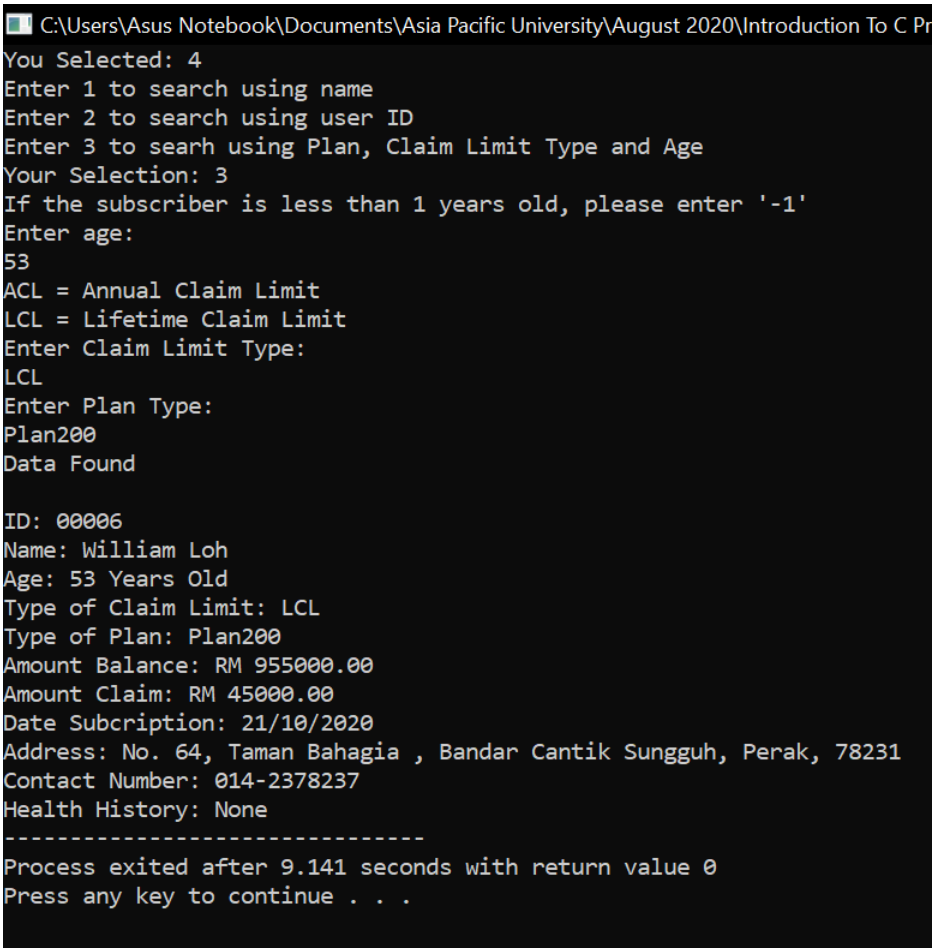
Test Plan	14
Test Objective	To check whether the program is able to read data from a text file which is “Insurance_Plans.txt” and display the total amount claimed by Lifetime Claim Limit user in the “accountInformation” function
Function	accountInformation()
Input	1
Expected Output	The program should be able to read the data in the “Insurance_Plan.txt” and add all the Lifetime Claim Limit user’s total insurance claim and display the total to the user
Output	
Description of Output	The program is able to check the “Insurance_Plans.txt” file and get the total of all the Lifetime Claim Limit Subscriber’s Claim Amount and display the total to the user
Prove of Text File Contents	Insurance_Plans.txt - Notepad File Edit Format View Help ID: 00004 Name: Jack Pa Age: 12 Years Old Type of Claim Limit: LCL Type of Plan: Plan120 Amount Balance: RM 548030.00 Amount Claim: RM 51970.00 Date Subcription: 21/10/2020 Address: No. 78, Taman Bukit Terang, Bandar Cahaya, Pahang, 89234 Contact Number: 019-7834534 Health History: None ID: 00005 Name: Roger Smith Age: 27 Years Old Type of Claim Limit: LCL Type of Plan: Plan150 Amount Balance: RM 745200.00 Amount Claim: RM 4800.00 Date Subcription: 21/10/2020 Address: No. 77F, Jalan Radin Bagus, Bandar Sri Petaling, Terrenganu, 13892 Contact Number: 016-2378237 Health History: Kidney Stones ID: 00006 Name: William Loh Age: 53 Years Old Type of Claim Limit: LCL Type of Plan: Plan200 Amount Balance: RM 955000.00 Amount Claim: RM 45000.00 Date Subcription: 21/10/2020 Address: No. 64, Taman Bahagia , Bandar Cantik Sungguh, Perak, 78231 Contact Number: 014-2378237 Health History: None
Calculation	Total = RM 51,970.00 + RM4,800.00 + RM45,000.00 = RM101,770.00

Test Plan	15
Test Objective	To check whether the program is able to read data from a text file, which is “Insurance_Plans.txt” and display the number of Annual Claim Limit users that exhausted their amount balance in the “accountInformation” function
Function	accountInformation()
Input	2
Expected Output	The program should be able to read the data in the “Insurance_Plans.txt” and add all the Annual Claim Limit users who have exhausted all their amount balance and display the total to the user
Output	
Description of Output	The program is able to check the “Insurance_Plans.txt” file and get the total number of Annual Claim Limit Subscriber’s who have exhausted all their amount balance and display the total to the user
Prove of Text File Contents	

Test Plan	16
Test Objective	To check if the program will prompt the user to enter again in “search” function, if the user enters the wrong number or symbols
Function	search()
Input	5 OR ;
Expected Output	The program should prompt user to enter again
Output	
Description of Output	The program was able to prompt user that user have entered a wrong input and ask user to enter again

Test Plan	17
Test Objective	To check whether the program is able to read data from a text file, which is “Insurance_Plans.txt”, and find a particular subscriber that the user is looking for using the name entered by the user and displaying the subscriber info once it is found.
Function	search()
Input	4, 1, Jack Pa
Expected Output	The program should be able to read the data in the “Insurance_Plans.txt” and compare the name entered by the user with the name in the text file
Prove of Output	
Description of Output	The program was able to search for the user based on the name entered by the user and the name in the text file. Once the user is found, the user record is displayed

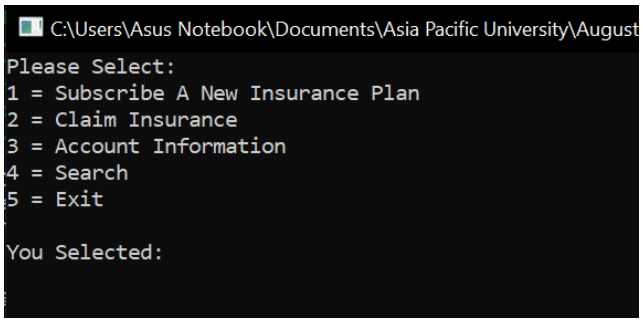
Test Plan	18
Test Objective	To check whether the program is able to read data from a text file, which is “Insurance_Plans.txt”, and find a particular subscriber that the user is looking for using the id entered by the user and displaying the subscriber info once it is found.
Function	search()
Input	4, 2, 00002
Expected Output	The program should be able to read the data in the “Insurance_Plans.txt” and compare the id entered by the user with the id in the text file
Prove of Output	
Description of Output	The program was able to search for the user based on the id entered by the user and the id in the text file. Once the user is found, the user record is displayed

Test Plan	19
Test Objective	To check whether the program is able to read data from a text file which is “Insurance_Plans.txt”, and find a particular subscriber that the user is looking for using the combination of age, plan type, and type of claim limit entered by the user and displaying the subscriber info, once it is found.
Function	search()
Input	4, 2, 53, LCL, Plan200
Expected Output	The program should be able to read the data in the “Insurance_Plans.txt” and compare the combination of age, plan type, and type of claim limit entered by the user with the age, plan type, and type of claim limit in the text file
Prove of Output	
Description of Output	The program was able to search for the user using the combination of age, claim limit and plan type entered by the user and compare the data in the text file. Once the user is found, the user record is displayed

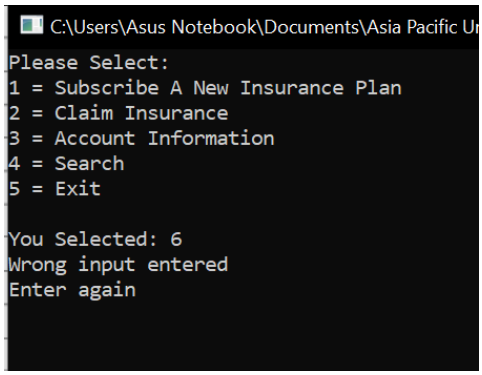
7.0 Sample outputs

Sample output 1

When the program first executed, the program will display a menu for the user to select:

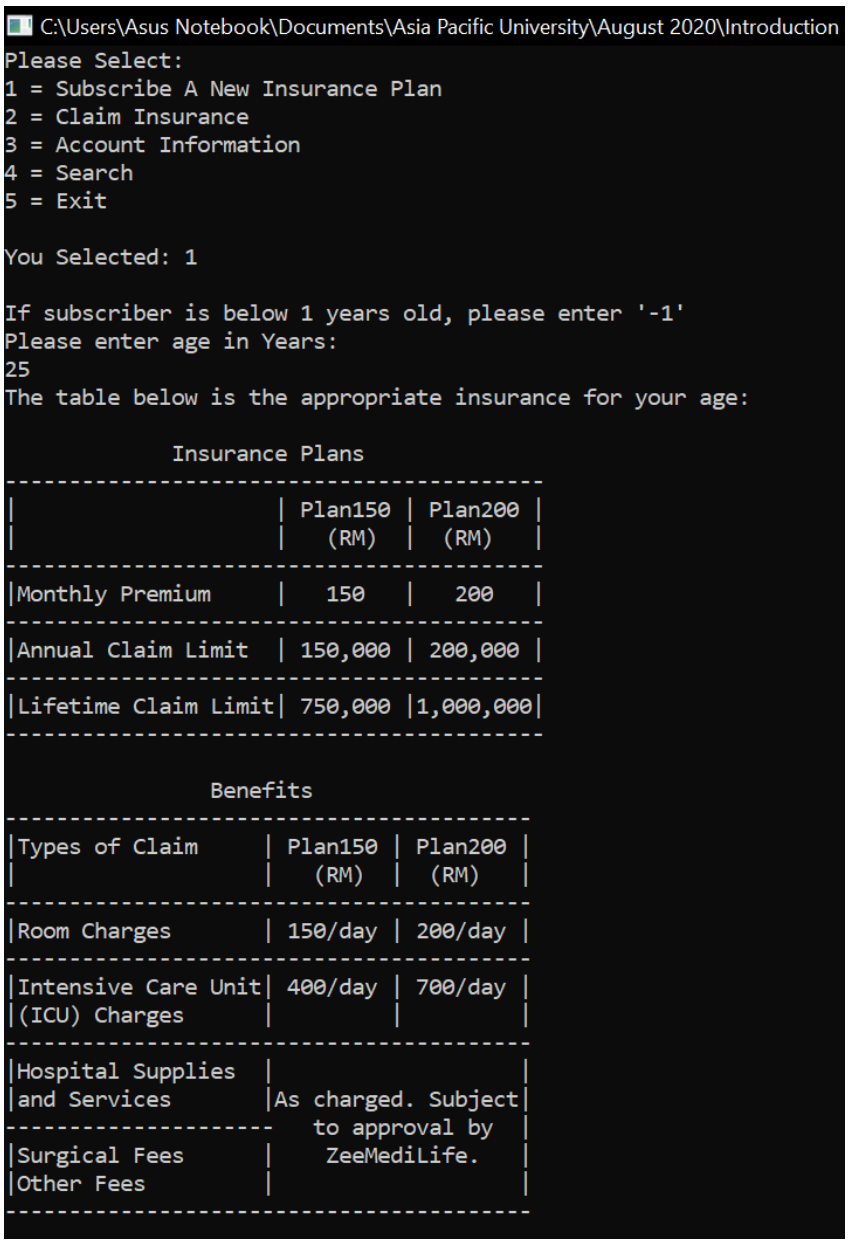


Users can select from the range of 1 to 5. If the user accidentally enters a number, not from the range, the program will prompt the user to enter again like the image below.

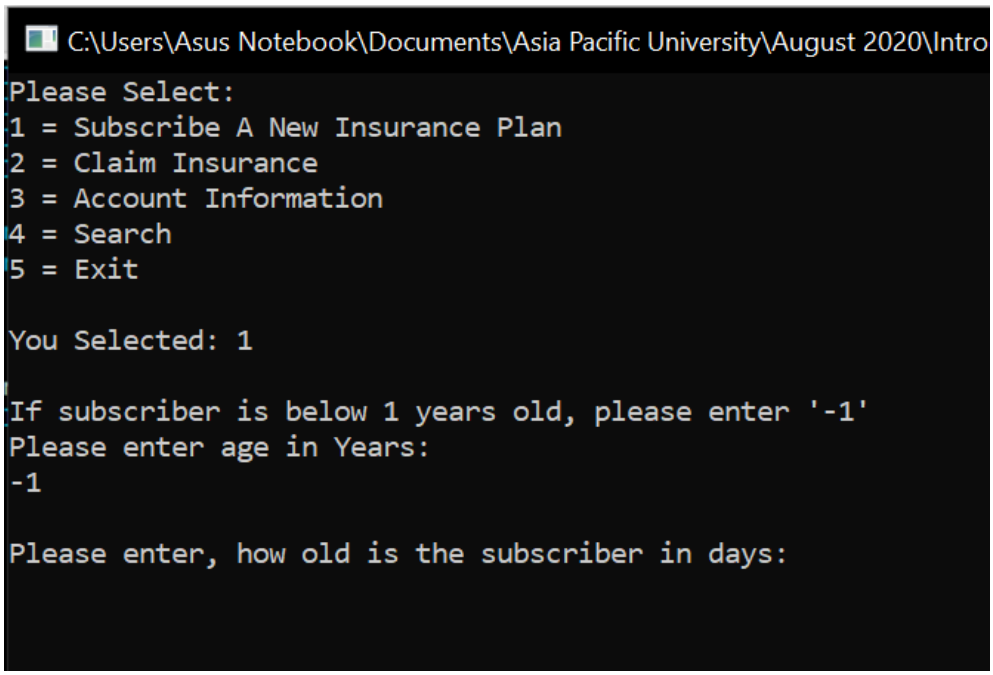


Sample Output 2

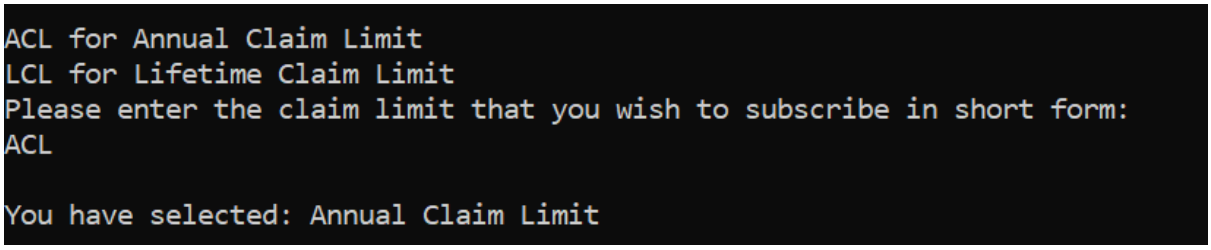
If the user enters “1”, which is the subscription to a new insurance plan, the user will be redirected to enter their age, plan type, and type of claim limit to subscribe from. Once the age is entered, an insurance plan that is suitable for the user age is displayed. The figure below is an example.



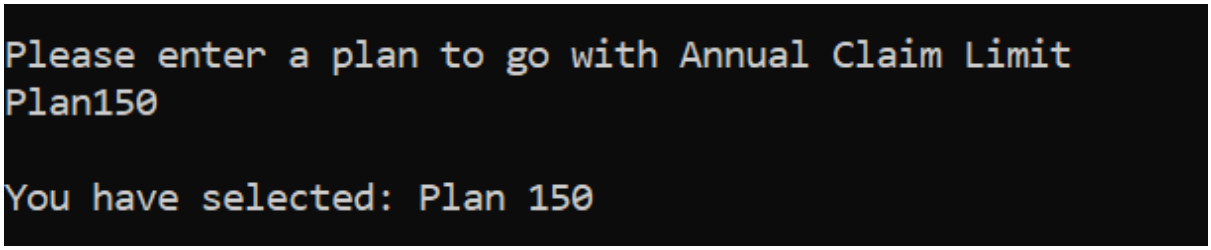
The user also has an option to enter their age in days if the subscriber is below one year old.



The user will then be prompt to enter the type of claim limit, either “ACL” or “LCL”.



After that user will have to enter the plan that the user wishes to apply



If the user enters an incorrect age, plan type, and type of claim, the program will prompt the user to enter an appropriate answer. This is to prevent the user from entering wrong or incorrect information into the program.

After the user entered the age, plan type, and type of claim limit, the user will be asked to enter all user information such as name, address, contact number, and any health history or diseases.

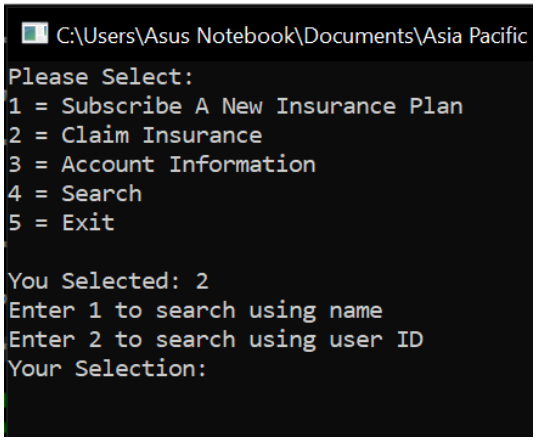
```
User Deatils
Enter Full Name:
Loke Weng Khay
Enter House Number:
No. 23G
Enter Street Address:
Taman Gembira
Enter City:
Kuala Lumpur
Enter State:
Wilayah Persekutuan
Enter Zip Code:
12189
Enter Contact Number:
012-347 8394
Do you have health issue: ('Yes','No')
Yes
Please state all your health issue that you are facing in one line and please use comma after naming each issue:
Kidney Stone
Thank you for entering all the information needed
```

After entering all the information, the data will be shown to the user and stored in a text file

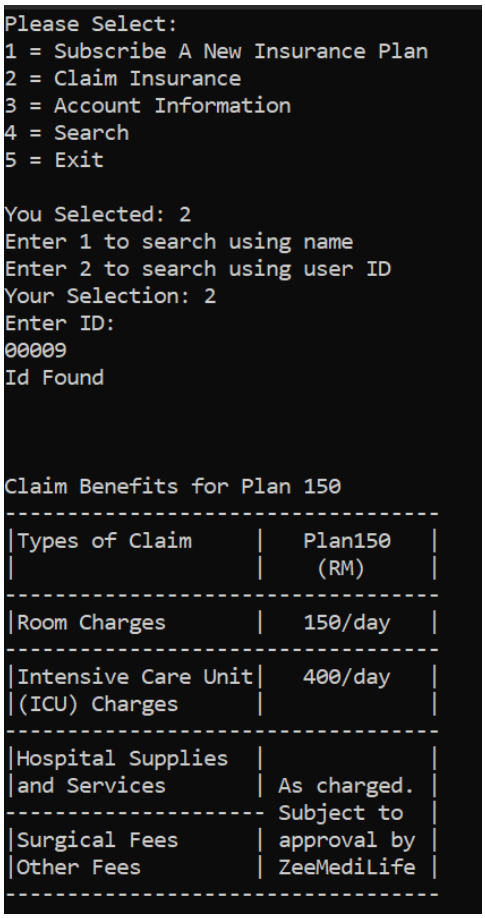
```
User Information:
User ID: 00009
Name: Loke Weng Khay
Age: 25 Years Old
Claim Limit Type: ACL
Type of plan: Plan150
Amount Subscribe: RM 150000.00
Date Subcription: 31/10/2020
Address: No. 23G, Taman Gembira, Kuala Lumpur, Wilayah Persekutuan, 12189
Contact: 012-347 8394
Health Issues: Kidney Stone

--------------------------------
Process exited after 87.36 seconds with return value 0
Press any key to continue . . .
```

Sample Output 3

If the user enters “2”, which is the claim insurance option, the user will be redirected to enter either to search the user by name or id.



Once the user has entered the user’s name or user’s id, the program will show if the program is able to search for a user record or not. If the record is found, the program will prompt the benefits table according to the user’s plan type



After that, the program will ask the user to enter the number of days in the normal wardroom, ICU room, the total amount of hospital supplies and service fees, and the total amount of surgical and other fees.

```
Please enter the number of days that you stay in the normal ward
4

Please enter the number of days that you stay in the Intensive Care Unit
4

Please enter the total hospital supplies and services fees
RM 4000

Please enter the total surgical and other fees
RM 400
Your claim have been received
```

After that, the program will display the user claims and store the record into a text file

```
Insurance Claim:
ID: 00009
Name: Loke Weng Khay
Claim Date: 31/10/2020
Total Amount Claim: RM 6600.00
Balance Claimable Amount: RM 143400.00

--------------------------------
Process exited after 8.869 seconds with return value 0
Press any key to continue . . .
```

Sample Output 4

If the user enters “3”, which is the account information, the user will be prompt with another menu to either select to see all Lifetime Claim Limit subscriber amount claimed or to see the total number of annual Claim Limit subscriber who has exhausted all their amount Balance

```
C:\Users\Asus Notebook\Documents\Asia Pacific University\August 2020\Introduction To C Programming\Assignment\Assignment Code\Final_
Please Select:
1 = Subscribe A New Insurance Plan
2 = Claim Insurance
3 = Account Information
4 = Search
5 = Exit

You Selected: 3

Enter 1 for Total amount claimed by Lifetime Claim Limit subscribers
Enter 2 for total number of Annual Claim Limit subscribers who have exhausted all their eligible amount.
Your selection:
```

If the user selects “1”, which is to see the total amount claimed by Lifetime Claim Limit subscriber, the program will display the total amount claimed

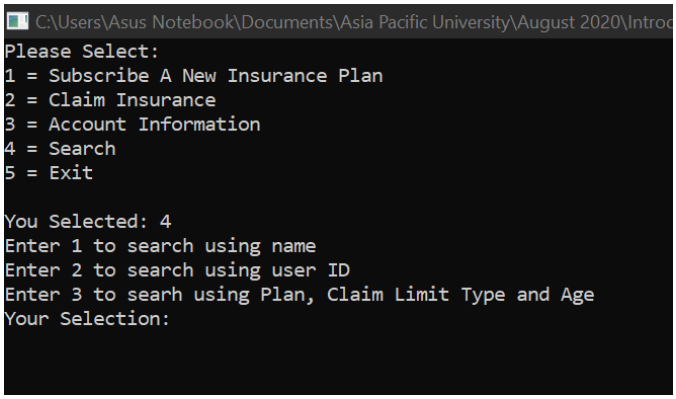
```
Enter 1 for Total amount claimed by Lifetime Claim Limit subscribers
Enter 2 for total number of Annual Claim Limit subscribers who have exhausted all their eligible amount.
Your selection: 1

Total Claim by Lifetime Claim Limit subscriber is RM 160040.00
--------------------------------
Process exited after 63.28 seconds with return value 0
Press any key to continue . . .
```

If the user selects “2”, which is to see the total number of Annual Claim Limit subscriber who has exhausted their amount balance, the program will display the total number of Annual Claim Limit.

```
Enter 1 for Total amount claimed by Lifetime Claim Limit subscribers
Enter 2 for total number of Annual Claim Limit subscribers who have exhausted all their eligible amount.
Your selection: 2

Total Annual Claim Limit subscriber that have exhausted their amount is 2
--------------------------------
Process exited after 0.8338 seconds with return value 0
Press any key to continue . . .
```

Sample output 5

If the user enters “4”, which is the search option, the user will be prompt with another menu to either search user by name, id, or the combination of age, type of plan, and type of claim limit.



If the user enters “1”, which is searching by using a name, the user will be asked to enter the user’s full name to find the user record

```
Enter 1 to search using name
Enter 2 to search using user ID
Enter 3 to searh using Plan, Claim Limit Type and Age
Your Selection: 1

Enter Full Name:
Malika Benitez
Name Found

ID: 00004
Name: Malika Benitez
Age: 16 Years Old
Type of Claim Limit: LCL
Type of Plan: Plan150
Amount Balance: RM 739800.00
Amount Claim: RM 10200.00
Date Subcription: 31/10/2020
Address: No. 34, Taman Gembira, Kuala Lumpur, Wilayah Persekutuan, 82372
Contact Number: 012-4538734
Health History: None

--------------------------------
Process exited after 15.1 seconds with return value 0
Press any key to continue . . .
```

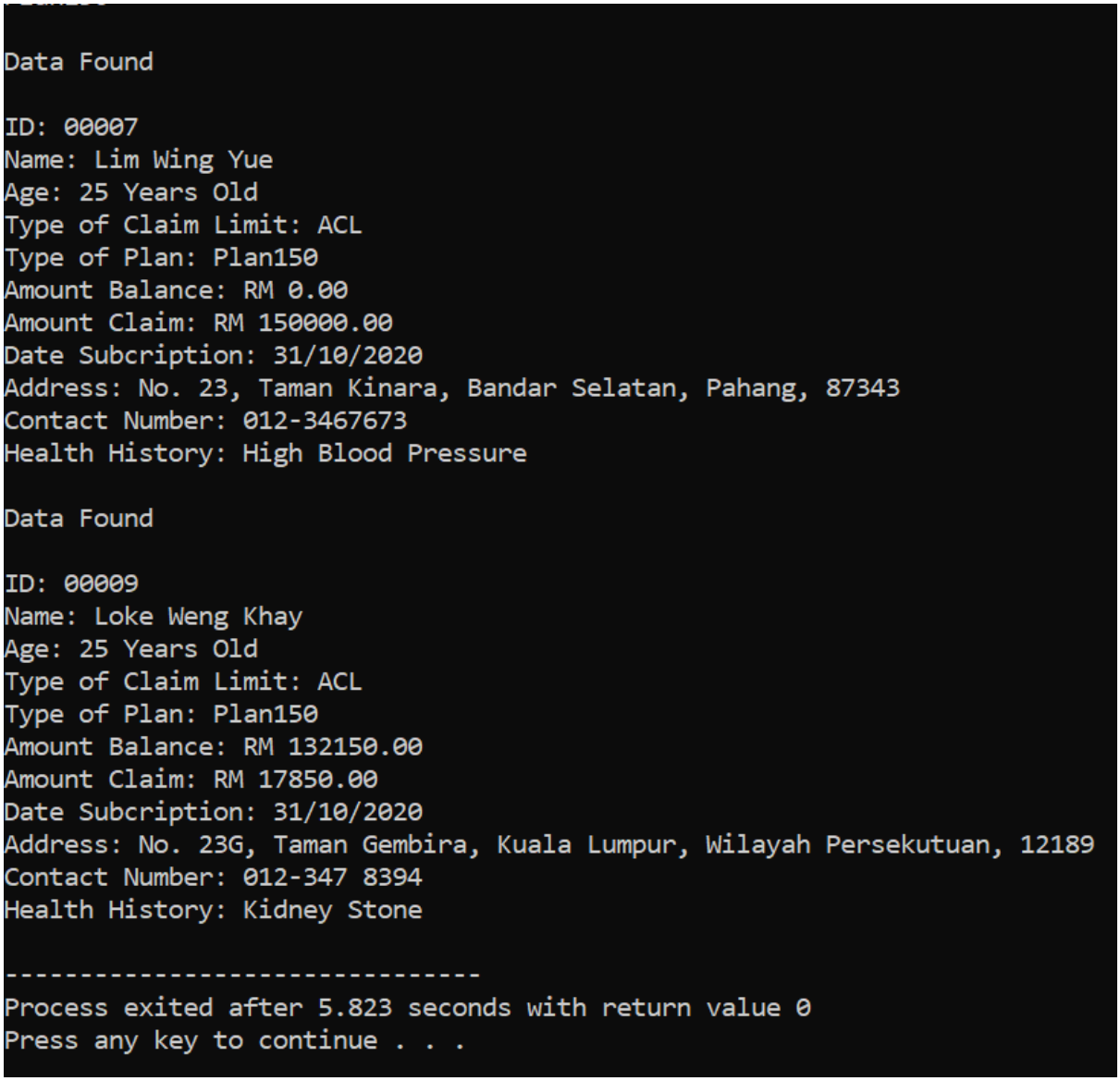
If the user enters “2”, which is searching by using id, the user will be asked to enter the user id to find the user record

```
Enter 1 to search using name
Enter 2 to search using user ID
Enter 3 to searh using Plan, Claim Limit Type and Age
Your Selection: 2
Enter ID:
00009
Id Found

ID: 00009
Name: Loke Weng Khay
Age: 25 Years Old
Type of Claim Limit: ACL
Type of Plan: Plan150
Amount Balance: RM 143400.00
Amount Claim: RM 6600.00
Date Subcription: 31/10/2020
Address: No. 23G, Taman Gembira, Kuala Lumpur, Wilayah Persekutuan, 12189
Contact Number: 012-347 8394
Health History: Kidney Stone
--------------------------------
Process exited after 3.419 seconds with return value 0
Press any key to continue . . .
```

If the user enters “3”, which is searching by using user age, plan type, and type of claim limit, the user will be asked to enter the user age, plan type, and type of claim limit to find the user record.

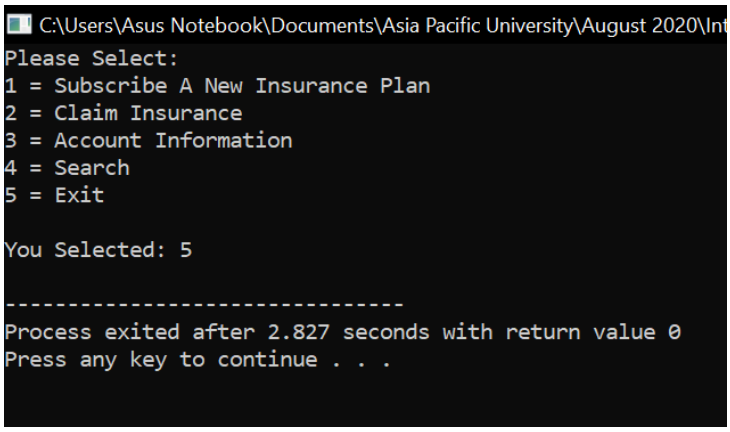
```
Enter 1 to search using name
Enter 2 to search using user ID
Enter 3 to searh using Plan, Claim Limit Type and Age
Your Selection: 3
If the subscriber is less than 1 years old, please enter '-1'
Enter age:
25
ACL = Annual Claim Limit
LCL = Lifetime Claim Limit
Enter Claim Limit Type:
ACL
Enter Plan Type:
Plan150
```

But with this option, more than one user record will be displayed as it has the possibility to have a user with the same age, plan type, and type of claim limit. With this, the program will be printing all the user record that matches the age, plan type, and type of claim limit entered by the user. The figure below is an example of this happening.



Sample output 6

If the user enters “5”, which is the exit option, the program will immediately exit without any errors.



8.0 Conclusion

This health insurance system was a challenge for me as it was a test of all the skills that I have learned and acquired in my Introduction to C Programming classes. During the process of making this application, this program has taught me to think out of the box more often, and it also gives a good overview look at how industry programming standards look like.

Furthermore, C programming has been an essential skill to obtain, as it is the necessary skills or baseline requirement for today's standards. Every Institution of Higher Learning will mandate students to learn C programming in the first semester when they enrol in a Diploma or Degree in Computer Science course.

Although I have met most of the requirements of this program, I felt I could have done this program better, but I was faced with multiple challenges when doing this assignment.

One of the challenges was lack of time. This is because this program has to be completed in 3 and a half weeks, and this was a challenge as this project was a massive program to code and document in three and a half weeks. The other challenge was this was not the only assignment I have to complete as I have other course assignments to completed in a given time.

In the end, with the help of my time management skills, I was able to manage my time between assignment projects loads and able hand in the work or assignment in a timely manner.

Last but not least, C programming has given me a good basic knowledge of C programming and its logic. With this, it will further help me to improve in the future when I progress studying C++ and C# programming language or other programming languages such as Java, Perl, Python, and many more.

9.0 References

Anon., 2020. *Wikibooks.* [Online]
Available at:
https://en.wikibooks.org/wiki/A_Level_Computer_Science_Programming_Guide/File_Handling
[Accessed 25 October 2020].

Anon., n.d. *Beginners Book.* [Online]
Available at: https://beginnersbook.com/2017/11/c-strcat-function-with-example/
[Accessed 25 October 2020].

Anon., n.d. *Programiz.* [Online]
Available at: https://www.programiz.com/c-programming/c-user-defined-functions
[Accessed 23 October 2020].

Goswami, M., n.d. *Study.com.* [Online]
Available at: https://study.com/academy/lesson/ansi-c-history-formation-structure.html#:~:text=Brief%20History%20of%20C&text=He%20added%20many%20powerful%20features,C%20language%20with%20better%20features.
[Accessed 22 October 2020].

Manley, G., 2020. *Section.* [Online]
Available at: https://www.section.io/engineering-education/history-of-c-programming-language/
[Accessed 20 October 2020].

Munoz, D., 2020. *Developers.* [Online]
Available at: https://www.toptal.com/c/after-all-these-years-the-world-is-still-powered-by-c-programming
[Accessed 20 October 2020].