Table of Contents

1.0 Introduction and Assumptions

The designed C programming is a Health Insurance Management System. The health insurance is introduced by ZeeMediLife insurance company. The primary purpose of this system is to record the insurance plan subscription automatically instead of collecting the information and storing using Microsoft Excel or Word. The designed system provides the function of registering new subscription, claiming process, storing information about customers' details and plan details, showing customers' account information, and searching functionalities.

The designed system is expected to accept the registration of new plan subscription and record the related information into three different text files (Plan120.txt, Plan150.txt, Plan200.txt) according to their registered plan. Other text files (Claim_plan120.txt, Claim_plan150.txt, Claim_plan200.txt) are used to store the account balance of customers to claim insurance.

In this program, there are three types of health insurances. Every customer is allowed to subscribe to one of the plans and choose an annual claim limit or lifetime claim limit. The system only accepts the subscription of people from 15 days old baby until 54 years old adult. All the eligible subscriber can subscribe Plan 200, while Plan 150 only accepts people from 15 days old until 40 years old and Plan 120 only accepts people from 15 days old until 20 years old.

A subscriber with an annual claim limit can only claim his or her subscribed insurance until 60 years old. For example, a 54 years old adult who subscribes Plan 200 with an annual claim limit in the year 2020 can claim his insurance until the year 2026. Meanwhile, a subscriber with a lifetime claim limit can claim his insurance until the end of his life. After reached the claim limit of each claim limit types, subscribers are not allowed to claim any insurances. Every customer should subscribe to a plan before claiming any insurance. After the subscription, a customer ID will be generated, and it is unique.

2.0 Design of Program

2.1 Pseudocode

```
TYPE DEFINITION STRUCTURE plan_type
    DECLARE name, monthly_premium, annual_limit, lifetime_limit, room_charges, icu_charges AS
INTEGER
ENDSTRUCTURE
plan_type plan120 = { 120,120,12000000,60000000,120,250 }
plan_type plan150 = { 150,150,15000000,75000000,150,400 }
plan_type plan200 = { 200,200,20000000,100000000,200,700 }
p1 AS plan120 POINTER
p2 AS plan150 POINTER
p3 AS plan200 POINTER

FUNCTION plan_name(plan_type POINTER)
    PRINT newline
    PRINT(“Plan Name: Plan”+(plan_type POINTER).name)
    PRINT newline
    PRINT("Monthly Premium: "+(plan_type POINTER).monthly_premium)
    PRINT newline
    PRINT("Annual Claim Limit: "+annual_limit)
    PRINT newline
    PRINT("Lifetime Claim Limit: "+lifetime_limit)
    PRINT newline
    PRINT("Room Charges : "+room_charges+" / day")
    PRINT newline
    PRINT("Intensive Care Unit(ICU) Charge : "+icu_charges+" / day")
    PRINT newline
    PRINT("Hospital Supplies and Services: As charged. Subject to approval by ZeeMediLife")
    PRINT newline
    PRINT("Surgical Fees: As charged. Subject to approval by ZeeMediLife.")
    PRINT newline
    PRINT("Other Fees: As charged. Subject to approval by ZeeMediLife.")
ENDFUNCTION

FUNCTION subscription()
    DECLARE plan_type chosen_plan AS STRUCTURE
    year = 'NULL'
    main_repeat=1
    WHILE (main_repeat = 1) THEN
        repeat=1
        WHILE (repeat=1) THEN
            repeat=0
            PRINT newline
            age = READ ("Enter your age, (999) to exit:")
            IF (age=0) THEN
                PRINT newline
                days = READ "How many days old the baby is:"
                IF (days>=15) THEN
                    age_group=1
                ELSE
                    PRINT "Your baby is not eligible for applying any plans."
                    repeat=1
                ENDIF
            ELSEIF ((age>=1) AND (age<=20)) THEN
                age_group=1
            ELSEIF ((age>=21) AND (age<=40)) THEN
                age_group=2
        ELSEIF ((age>=41) AND (age<=54)) THEN
                age_group=3
        ELSEIF (age=999) THEN
                PRINT newline
                RETURN
        ELSE
                PRINT newline
                PRINT "Your age is not eligible for applying any plans."
                repeat=1
        ENDIF
ENDWHILE

repeat=1
IF (age_group=1) THEN
        PRINT newline
        PRINT "You are eligible for applying the following plans:"
        PRINT newline
        PRINT "1. Plan120"
        CALL plan_name(p1)
        PRINT "2. Plan150"
        CALL plan_name(p2)
        PRINT "3. Plan200"
        CALL plan_name(p3)
        WHILE (repeat=1) THEN
                repeat=0
                PRINT newline
                plan = READ "Enter your choice:"
                IF (plan=1) THEN
                        chosen_plan=plan120
                ELSEIF (plan=2) THEN
                        chosen_plan=plan150
                ELSEIF (plan=3) THEN
                        chosen_plan=plan200
                ELSE
                        PRINT newline
                        PRINT "Please choose a listed plan."
                        repeat=1
                ENDIF
        ENDWHILE
ELSEIF (age_group=2) THEN
        PRINT newline
        PRINT "You are eligible for applying the following plans:"
        PRINT newline
        PRINT "1. Plan150"
        CALL plan_name(p2)
        PRINT "2. Plan200"
                CALL plan_name(p3)
                WHILE (repeat=1) THEN
                repeat=0
                PRINT newline
                plan = READ "Enter your choice:"
                IF (plan=1) THEN
                        chosen_plan=plan150
                ELSEIF (plan=2) THEN
                        chosen_plan=plan200
                ELSE
                        PRINT newline
                        PRINT "Please choose a listed plan."
                        repeat=1
                ENDIF
        ENDWHILE
ELSEIF (age_group=3) THEN
        PRINT newline
        PRINT "You are eligible for applying the following plans:"
        PRINT newline
        PRINT "1. Plan200"
        CALL plan_name(p3)
        WHILE (repeat=1) THEN
                repeat=0
                PRINT newline
                plan = READ "Enter your choice:"
                IF (plan=1) THEN
                        chosen_plan=plan200
                ELSE
                        PRINT newline
                        PRINT "Please choose a listed plan."
                        repeat=1
                ENDIF
        ENDWHILE
ENDIF

repeat=1
PRINT newline
PRINT("Please choose a claim limit type:")
PRINT newline
PRINT("1. Annual Claim Limit")
PRINT newline
PRINT("2. Lifetime Claim Limit")
WHILE (repeat=1) THEN
        repeat=0
        claim= READ "Enter your choice:"
        IF (claim=1) THEN
                chosen_claim="Annual Claim Limit"
                chosen_plan.lifetime_limit IS NULL
        ELSEIF (claim=2) THEN
                chosen_claim="Lifetime Claim Limit"
                chosen_plan.annual_limit IS NULL
        ELSE
                PRINT newline
                PRINT("Please choose a listed claim limit.")
                repeat=1
        ENDIF
ENDWHILE

cus_name = READ("Name: ")
CALL no_spaces(cus_name,name)
cus_contact_number = READ("Contact Number: ")
CALL no_spaces(cus_contact_number,contact_number)
cus_house_number = READ("House Number: ")
CALL no_spaces(cus_house_number,house_number)
cus_street = READ("Street: ")
CALL no_spaces(cus_street,street)
cus_city = READ("City: ")
CALL no_spaces(cus_city,city)
cus_state = READ("State: ")
CALL no_spaces(cus_state,state)
cus_health_history = READ("Health History: ")
CALL no_spaces(cus_health_history,health_history)

count=0
```

```
CASE based on chosen_plan.name
        CASE 120
                OPEN and READ Plan120.txt AS f_subscribe
                OPEN and READ Claim_plan120.txt AS f_claim
                BREAK
        CASE150
                OPEN and READ Plan150.txt AS f_subscribe
                OPEN and READ Claim_plan150.txt AS f_claim
                BREAK
        CASE 200
                OPEN and READ Plan200.txt AS f_subscribe
                OPEN and READ Claim_plan200.txt AS f_claim
                BREAK
        DEFAULT
                PRINT "File cannot be opened."
ENDCASE
WHILE EACH line IN f_subscribe
        count = count +1
ENDWHILE
IF (count<10) THEN
        id = "00" + count
ELSEIF (count-2<100) THEN
        id = "0" + count
ELSE
        id = count
ENDIF
customer_id = age_group + chosen_plan.name + id
add customer_id into f_subscribe
add name into f_subscribe
add age into f_subscribe
add contact_number into f_subscribe
add house_number into f_subscribe
add street into f_subscribe
add city into f_subscribe
add state into f_subscribe
add health_history into f_subscribe
add chosen_plan.name into f_subscribe
add chosen_claim into f_subscribe

add customer_id into f_claim
add year into f_claim
add (chosen_plan.annual_limit/100) into f_claim
add (chosen_plan.annual_limit%100) into f_claim
add (chosen_plan.lifetime_limit/100) into f_claim
add (chosen_plan.lifetime_limit%100) into f_claim

CLOSE f_subscribe
CLOSE f_claim

PRINT newline
PRINT("Customer ID: "+customer_id)
PRINT newline
PRINT("Name: "+cus_name)
PRINT newline
PRINT("Age: "+age)
PRINT newline
PRINT("Phone Number: "+cus_contact_number)
PRINT newline
PRINT("Address:"+cus_house_number+cus_street+cus_city+cus_state)
PRINT newline
```

```
        PRINT("Health History:"+cus_health_history)
        PRINT newline
        PRINT("Chosen Plan: Plan"+chosen_plan.name)
        PRINT newline
        PRINT("Claim Limit Type: "+chosen_claim)

        main_repeat = 1
    ENDWHILE

ENDFUNCTION

-----------------------------------------------------------------------------------------------

FUNCTION claim()
    DECLARE plan_type chosen_plan AS STRUCTURE
    year=2020
    day_ward=0
    day_icu=0
    services=0
    surgery=0
    others=0
    main_repeat=1
    WHILE (main_repeat=1) THEN
        repeat=1
        WHILE (repeat=1) THEN
            customer_id= READ ("Enter customer ID, 999 to exit:")
            IF (customer_id=999) THEN
                RETURN
            ENDIF
            FOR     a FROM 0 TO 2 STEP 1
                CASE based on a
                    CASE 0
                        OPEN and READ Claim_plan120.txt AS f_claim
                        chosen_plan=plan120
                        BREAK
                    CASE 1
                        OPEN and READ Claim_plan150.txt AS f_claim
                        chosen_plan=plan150
                        BREAK
                    CASE 2
                        OPEN and READ Claim_plan200.txt AS f_claim
                        chosen_plan=plan200
                        BREAK
                    DEFAULT
                        PRINT "File cannot be opened."
                ENDCASE

                FOR EACH line IN f_claim
                    FOR EACH index,line IN ENUMERATE(f_claim)
                        input_id = f_claim[index]
                        IF (input_id = customer_id) THEN
                            claim_in_rm=f_claim[index]
                            claim_in_sen=f_claim[index]
                            yearleft_rm=f_claim[index]
                            yearleft_sen=f_claim[index]
                            lifeleft_rm=f_claim[index]
                            lifeleft_sen=f_claim[index]
                            amount_claim = claim_in_rm * 100 + claim_in_sen
                            amount_year = yearleft_rm * 100 + yearleft_sen
                            amount_life = lifeleft_rm * 100 + lifeleft_sen
                            CLOSE f_claim
                            BREAK
                        ELSE
                            NEXT line
                        ENDIF
                    ENDFOR
                ENDFOR

                CASE based on chosen_plan.name
                    CASE 120
                        OPEN and READ Plan120.txt AS f_subscribe
                        BREAK
                    CASE 150
                        OPEN and READ Plan150.txt AS f_subscribe
                        BREAK
                    CASE 200
                        OPEN and READ Plan200.txt AS f_subscribe
                ENDCASE

                FOR EACH line IN f_subscribe
                    FOR EACH index,line IN ENUMERATE(f_subscribe)
                        input_id = f_subscribe[index]
                        IF (input_id = customer_id) THEN
                            chosen_claim=f_subscribe[index+9]
                            CLOSE f_subscribe
                            repeat=0
                            BREAK
                        ELSE
                            NEXT line
                        ENDIF
                    ENDFOR
                ENDFOR
            ENDFOR

            day_ward = READ "How many days do you stay in normal wards: "
            day_icu = READ "How many days do you stay in ICU: "
            services = READ "Enter the amount of hospital supplies and services in RM: "
            surgery = READ "Enter the amount of surgical fees in RM: "
            others = READ "Enter the amount of other fees in RM: "

            sum = (day_ward * chosen_plan.room_charges + day_icu * chosen_plan.icu_charges + services + surgery + others) * 100

            IF (chosen_claim="Lifetime Claim Limit") THEN
                IF (sum > amount_life)
                    PRINT "Your available balance for current plan is not enough."
                    sum = amount_life
                ENDIF
            ELSE
                IF (sum > amount_year)
                    PRINT "Your available balance for current plan is not enough."
                    sum = amount_year
                ENDIF
            ENDIF

            CASE based on chosen_plan.name
                CASE 120
                    OPEN and READ Claim_plan120.txt AS f_claim
                    BREAK
```

```
                CASE 150
                    OPEN and READ Claim_plan150.txt AS f_claim
                    BREAK
                CASE 200
                    OPEN and READ Claim_plan200.txt AS f_claim
            ENDCASE

            updated_amount_claim_rm = (amount_claim + sum) / 100
            updated_amount_claim_sen = (amount_claim + sum) % 100
            updated_amount_life_rm = (amount_life - sum) / 100
            updated_amount_life_sen = (amount_life - sum) % 100
            updated_amount_year_rm = (amount_year - sum) / 100
            updated_amount_year_sen = (amount_year - sum) % 100

            IF (chosen_claim="Lifetime Claim Limit") THEN
                add customer_id into f_claim
                add year into f_claim
                add updated_amount_claim_rm into f_claim
                add updated_amount_claim_sen into f_claim
                add updated_amount_life_rm into f_claim
                add updated_amount_life_sen into f_claim
            ELSE
                add customer_id into f_claim
                add year into f_claim
                add updated_amount_claim_rm into f_claim
                add updated_amount_claim_sen into f_claim
                add updated_amount_year_rm into f_claim
                add updated_amount_year_sen into f_claim
            ENDIF

            CLOSE f_claim
            main_repeat=1

        ENDWHILE
    ENDWHILE
ENDFUNCTION

-----------------------------------------------------------------------------------------------

FUNCTION information()
    a=0
    b=0
    total_amount_claimed = 0
    num_exsubscriber = 0
    DECLARE num_lifesubscriber[1000][10]
    DECLARE num_yearsubscriber[1000][10]
    FOR c FROM 0 TO 2 STEP 1
        CASE based on c
            CASE 0
                OPEN and READ Plan120.txt AS f_subscribe
                BREAK
            CASE 1
                OPEN and READ Plan150.txt AS f_subscribe
                BREAK
            CASE 2
                OPEN and READ Plan200.txt AS f_subscribe
        ENDCASE
        FOR EACH line IN f_subscribe
            GET chosen_claim FROM f_subscribe
            GET customer_id FROM f_subscribe
            IF (chosen_claim="Lifetime Claim Limit") THEN
                num_lifesubscriber[a]=customer_id
                a=a+1
            ELSE
                num_yearsubscriber[a]= customer_id
                b=b+1
            ENDIF
        ENDFOR
        CLOSE f_subscribe
    ENDFOR

    FOR d FROM 0 TO 2 STEP 1
        CASE based on d
            CASE 0
                FOR c FROM 0 TO 2 STEP 1
                    CASE based on c
                        CASE 0
                            OPEN and READ Claim_plan120.txt AS f_claim
                            BREAK
                        CASE 1
                            OPEN and READ Claim_plan150.txt AS f_claim
                            BREAK
                        CASE 2
                            OPEN and READ Claim_plan200.txt AS f_claim
                    ENDCASE
                    FOR EACH line IN f_claim
                        GET customer_id FROM f_claim
                        GET claim_in_rm FROM f_claim
                        GET claim_in_sen FROM f_claim
                        FOR a FROM 0 TO 999 STEP 1
                            IF ( num_lifesubscriber[a] = customer_id ) THEN
                                amount_claim = claim_in_rm * 100 + claim_in_sen
                                total_amount_claimed = total_amount_claimed + amount_claim
                            ENDIF
                        ENDFOR
                    ENDFOR
                ENDFOR
                BREAK
            CASE 1
                FOR c FROM 0 TO 2 STEP 1
                    CASE based on c
                        CASE 0
                            OPEN and READ Claim_plan120.txt AS f_claim
                            BREAK
                        CASE 1
                            OPEN and READ Claim_plan150.txt AS f_claim
                            BREAK
                        CASE 2
                            OPEN and READ Claim_plan200.txt AS f_claim
                    ENDCASE
                    FOR EACH line IN f_claim
                        GET customer_id FROM f_claim
                        GET yearleft_rm FROM f_claim
                        GET yearleft_sen FROM f_claim
                        FOR a FROM 0 TO 999 STEP 1
                            IF ( num_yearsubscriber[a] = customer_id ) THEN
```

```
                                amount_year = yearleft_rm * 100 + yearleft_sen
                                IF (amount_year=0) THEN
                                    num_exsubscriber = num_exsubscriber + 1
                                ENDIF
                            ENDIF
                        ENDFOR
                    ENDFOR
                ENDFOR
        ENDCASE
    ENDFOR
    CLOSE f_claim
    PRINT newline
    PRINT ("Total amount claimed by Lifetime Claim Limit subscribers(RM): "+ total_amount_claimed / 100 + total_amount_claimed % 100)
    PRINT newline
    PRINT ("Total number of Annual Claim Limit subscribers who have exhausted all their eligible amount: " + num_exsubscriber)
ENDFUNCTION

------------------------------------------------------------------------------------------------------

FUNCTION search()
    main_repeat=1
    WHILE (main_repeat=1) THEN
        main_repeat=0
        PRINT newline
        PRINT "Do you want use searching functionalities by entering: "
        PRINT newline
        PRINT "1. Customer ID or Name"
        PRINT newline
        PRINT "2. Plan, Claim Limit Type and Age"
        PRINT newline
        PRINT "999. Exit"
        choice= READ ("Enter your choice: ")
        CASE based on choice
            CASE 1
                repeat=1
                WHILE (repeat=1) THEN
                    repeat=0
                    PRINT newline
                    PRINT "Do you want use searching functionalities by entering: "
                    PRINT newline
                    PRINT "1. Customer ID"
                    PRINT newline
                    PRINT "2. Customer Name"
                    choice= READ ("Enter your choice: ")
                    CASE based on choice
                        CASE 1
                            name_or_id = READ "Enter Customer ID: "
                            FOR a FROM 0 TO 2 STEP 1
                                CASE based on a
                                    CASE 0
                                        OPEN and READ Plan120.txt AS f_subscribe
                                        OPEN and READ Claim_plan120.txt AS f_claim
                                        BREAK
                                    CASE 1
            OPEN and READ Plan150.txt AS f_subscribe
            OPEN and READ Claim_plan150.txt AS f_claim
            BREAK
        CASE 2
            OPEN and READ Plan200.txt AS f_subscribe
            OPEN and READ Claim_plan200.txt AS f_claim
    ENDCASE
    FOR EACH line IN f_subscribe
        GET customer_id FROM f_subscribe
        IF (customer_id = name_or_id ) THEN
            GET name FROM f_subscribe
            GET age FROM f_subscribe
            GET contact_number FROM f_subscribe
            GET house_number FROM f_subscribe
            GET street FROM f_subscribe
            GET city FROM f_subscribe
            GET state FROM f_subscribe
            GET health_history FROM f_subscribe
            GET plan FROM f_subscribe
            GET chosen_claim FROM f_subscribe
            FOR EACH line IN f_claim
                GET cus FROM f_claim
                GET name_or_id FROM f_claim
                GET amount_year FROM f_claim
                GET amount_claim FROM f_claim
                GET chosen_claim FROM f_claim
                GET balance FROM f_claim
            ENDFOR
            CLOSE f_claim
            PRINT newline
            PRINT ("Customer ID: " + customer_id)
            PRINT newline
            PRINT ("Name: " + name)
            PRINT newline
            PRINT ("Age: " + age)
```

```
                PRINT newline
                PRINT ("Contact Number: " + contact_number)
                PRINT newline
                PRINT ("Address:" + house_number + street + city + state)
                PRINT newline
                PRINT ("Health History: " + health_history)
                PRINT newline
                PRINT ("Plan: " + plan)
                PRINT newline
                PRINT ("Claim Limit Type: " + chosen_claim)
                PRINT newline
                PRINT ("Claimed Year: " + amount_year)
                PRINT newline
                PRINT ("Total of Claimed Insurance: " + amount_claim)
                PRINT newline
                PRINT ("Available Balance: " + balance)
                repeat=0
                BREAK
            ENDIF
        ENDIF
        CLOSE f_subscribe
        IF (repeat=0) THEN
            BREAK
        ENDIF
    ENDFOR
    IF (repeat=1) THEN
        PRINT "Please enter a valid customer ID."
    ENDIF
    BREAK
CASE 2
    repeat=1
    WHILE (repeat=1) THEN
        cus_name = READ ("Customer Name: ")
        CALL no_spaces(cus_name,name_or_id)
        FOR a FROM 0 TO 2 STEP 1
            CASE based on a
                CASE 0
                    OPEN and READ Plan120.txt AS f_subscribe
                    OPEN and READ Claim_plan120.txt AS f_claim
                    BREAK
                CASE 1
                    OPEN and READ Plan150.txt AS f_subscribe
                    OPEN and READ Claim_plan150.txt AS f_claim
                    BREAK
                CASE 2
                    OPEN and READ Plan200.txt AS f_subscribe
OPEN and READ Claim_plan200.txt AS f_claim
ENDCASE
FOR EACH line IN f_subscribe
    GET customer_id FROM f_subscribe
    GET name FROM f_subscribe
    IF (name=name_or_id) THEN
        GET name FROM f_subscribe
        GET age FROM f_subscribe
        GET contact_number FROM f_subscribe
        GET house_number FROM f_subscribe
        GET street FROM f_subscribe
        GET city FROM f_subscribe
        GET state FROM f_subscribe
        GET health_history FROM f_subscribe
        GET plan FROM f_subscribe
        GET chosen_claim FROM f_subscribe
        FOR EACH line IN f_claim
            GET cus FROM f_claim
            GET customer_id FROM f_claim
            GET amount_year FROM f_claim
            GET amount_claim FROM f_claim
            GET chosen_claim FROM f_claim
            GET balance FROM f_claim
        ENDFOR
        CLOSE f_claim
        PRINT newline
        PRINT ("Customer ID: " + customer_id)
        PRINT newline
        PRINT ("Name: " + name)
        PRINT newline
        PRINT ("Age: " + age)
```

```
                                            PRINT
newline
                                            PRINT
("Contact Number: " + contact_number)
                                            PRINT
newline
                                            PRINT
("Address:" + house_number + street + city + state)
                                            PRINT
newline
                                            PRINT
("Health History: " + health_history)
                                            PRINT
newline
                                            PRINT
("Plan: " + plan)
                                            PRINT
newline
                                            PRINT
("Claim Limit Type: " + chosen_claim)
                                            PRINT
newline
                                            PRINT
("Claimed Year: " + amount_year)
                                            PRINT
newline
                                            PRINT
("Total of Claimed Insurance: " + amount_claim)
                                            PRINT
newline
                                            PRINT
("Available Balance: " + balance)
                                            repeat=0
                                            BREAK
                                        ENDIF
                                        IF (repeat=1) THEN
                                            PRINT
"Please enter a valid customer ID."
                                        ENDIF
                                    ENDFOR
                                ENDFOR
                            ENDWHILE
                            BREAK
                        DEFAULT
                            PRINT "Please enter a valid input."
                            repeat = 1
                    ENDCASE
                ENDWHILE
                BREAK
            CASE 2
                repeat=1
                PRINT newline
                PRINT "Please select a plan:"
                PRINT newline
                PRINT "1. Plan120"
                PRINT newline
                PRINT "2. Plan150"
                PRINT newline
                PRINT "3. Plan200"
                WHILE (repeat=1) THEN
```

```
    repeat=0
    choice=READ "Enter Choice:"
    CASE based on choice
        CASE 1
            cus_plan = 120
            BREAK
        CASE 2
            cus_plan = 150
            BREAK
        CASE 3
            cus_plan = 200
            BREAK
        DEFAULT
            PRINT "Please select a valid plan."
            repeat=1
    ENDCASE
ENDWHILE
repeat=1
PRINT newline
PRINT "Please choose a claim limit type:"
PRINT newline
PRINT "1.Annual Claim Limit"
PRINT newline
PRINT "2.Lifetime Claim Limit"
WHILE (repeat=1) THEN
    repeat=0
    choice=READ "Enter Choice:"
    CASE based on choice
        CASE 1
            cus_chosen_plan = "Annual Claim Limit"
            BREAK
        CASE 2
            cus_chosen_plan = "Lifetime Claim Limit"
            BREAK
        DEFAULT
            PRINT "Please choose a listed claim limit type."
            repeat=1
    ENDCASE
ENDWHILE
cus_age = READ "Enter age: "
FOR a FROM 0 TO 2 STEP 1
    CASE based on a
        CASE 0
            OPEN and READ Plan120.txt AS f_subscribe
            OPEN and READ Claim_plan120.txt AS f_claim
            BREAK
        CASE 1
            OPEN and READ Plan150.txt AS f_subscribe
            OPEN and READ Claim_plan150.txt AS f_claim
            BREAK
        CASE 2
            OPEN and READ Plan200.txt AS f_subscribe
            OPEN and READ Claim_plan200.txt AS f_claim
    ENDCASE
    FOR EACH line IN f_subscribe
        GET customer_id FROM f_subscribe
        GET name FROM f_subscribe
        GET age FROM f_subscribe
        GET contact_number FROM f_subscribe
        GET house_number FROM f_subscribe
```

```
                        GET street FROM f_subscribe
                        GET city FROM f_subscribe
                        GET state FROM f_subscribe
                        GET health_history FROM f_subscribe
                        GET plan FROM f_subscribe
                        GET chosen_claim FROM f_subscribe
                        IF ( (cus_plan=plan) AND (cus_chosen_plan=chosen_claim) AND (cus_age=age) ) THEN
                            FOR EACH line IN f_claim
                                GET cus FROM f_claim
                                GET customer_id FROM f_claim
                                GET amount_year FROM f_claim
                                GET amount_claim FROM f_claim
                                GET chosen_claim FROM f_claim
                                GET balance FROM f_claim
                            ENDFOR
                            CLOSE f_claim
                            PRINT newline
                            PRINT ("Customer ID: " + customer_id)
                            PRINT newline
                            PRINT ("Name: " + name)
                            PRINT newline
                            PRINT ("Age: " + age)
                            PRINT newline
                            PRINT ("Contact Number: " + contact_number)
                            PRINT newline
                            PRINT ("Address:" + house_number + street + city + state)
                            PRINT newline
                            PRINT ("Health History: " + health_history)
                            PRINT newline
                            PRINT ("Plan: " + plan)
                            PRINT newline
                            PRINT ("Claim Limit Type: " + chosen_claim)
                            PRINT newline
                            PRINT ("Claimed Year: " + amount_year)
                            PRINT newline
                            PRINT ("Total of Claimed Insurance: " + amount_claim)
                            PRINT newline
                            PRINT ("Available Balance: " + balance)
                            repeat=0
                        ENDIF
                    ENDFOR
                ENDFOR
                IF (repeat=1) THEN
                    PRINT "Can't find the specific age."
                ENDIF
                BREAK
            CASE 999
                RETURN
                BREAK
            DEFAULT
                PRINT newline
                PRINT "Please select the available choice."
                main_repeat=1
        ENDCASE
    ENDWHILE
ENDFUNCTION
```

```
FUNCTION create_file()
    OPEN and CLOSE Plan120.txt
    OPEN and CLOSE Plan150.txt
    OPEN and CLOSE Plan200.txt
    OPEN and CLOSE Claim_plan120.txt
    OPEN and CLOSE Claim_plan150.txt
    OPEN and CLOSE Claim_plan200.txt
ENDFUNCTION

FUNCTION menu()
    statement=1
    WHILE statement EQUAL TO 1 THEN
        PRINT "HEALTH INSURANCE MANAGEMENT SYSTEM"
        PRINT newline
        PRINT "Please select one of the functions:"
        PRINT newline
        PRINT "1. Insurance Plan Subscription"
        PRINT newline
        PRINT "2. Claim Processing"
        PRINT newline
        PRINT "3. Accounts Information"
        PRINT newline
        PRINT "4. Searching Functionalities"
        PRINT newline
        PRINT "5. Exit"
        choice=READ("Enter your choice:")
        CASE based on choice
            CASE 1
                CALL subscription()
                BREAK
            CASE 2
                CALL claim()
                BREAK
            CASE 3
                CALL information()
                BREAK
            CASE 4
                CALL search()
                BREAK
            CASE 5
                RETURN
                BREAK
            DEFAULT
                PRINT newline
                PRINT("Invalid Input.")
                statement=1
        ENDCASE
    ENDWHILE
ENDFUNCTION

FUNCTION no_spaces(CHARACTER store[],CHARACTER filter[])
    a=0
    b=0
    FOR a FROM 0 TO (store[a] IS NOT EQUAL TO NULL) STEP 1
        IF (store[a] IS NOT EQUAL TO ' ')
            filter[b] = store[a]
            b = b + 1
        ENDIF
        ENDFOR
        filter[b] = NULL
ENDFUNCTION

PROGRAM HealthInsuranceManagementSystem
BEGIN
        CALL create_file()
        CALL menu()
END
```

2.2 Flowcharts

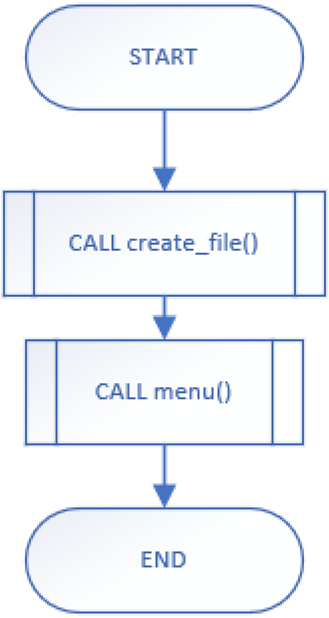


Figure 2.2.1: *Main*

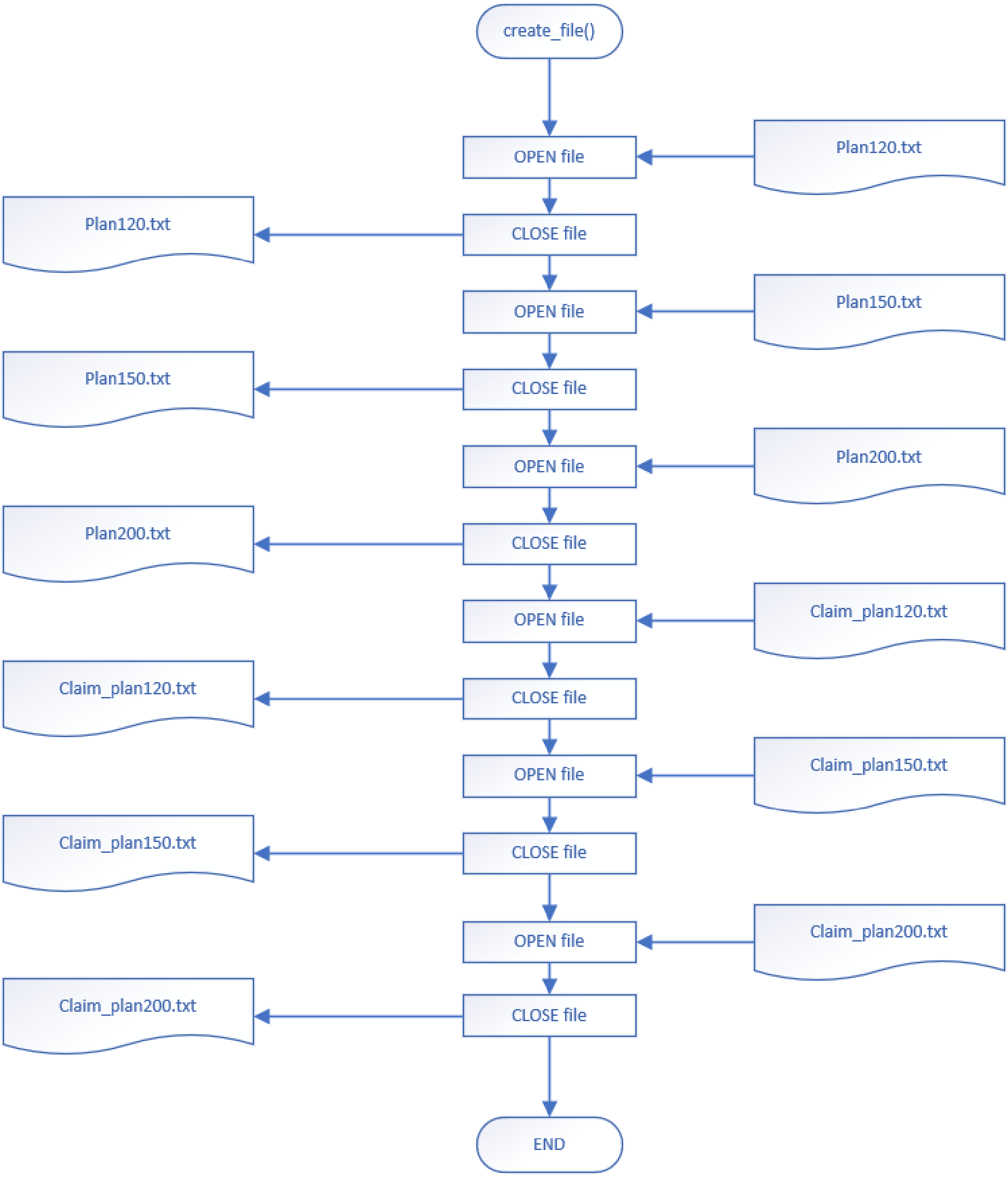


Figure 2.2.2: *Create File*

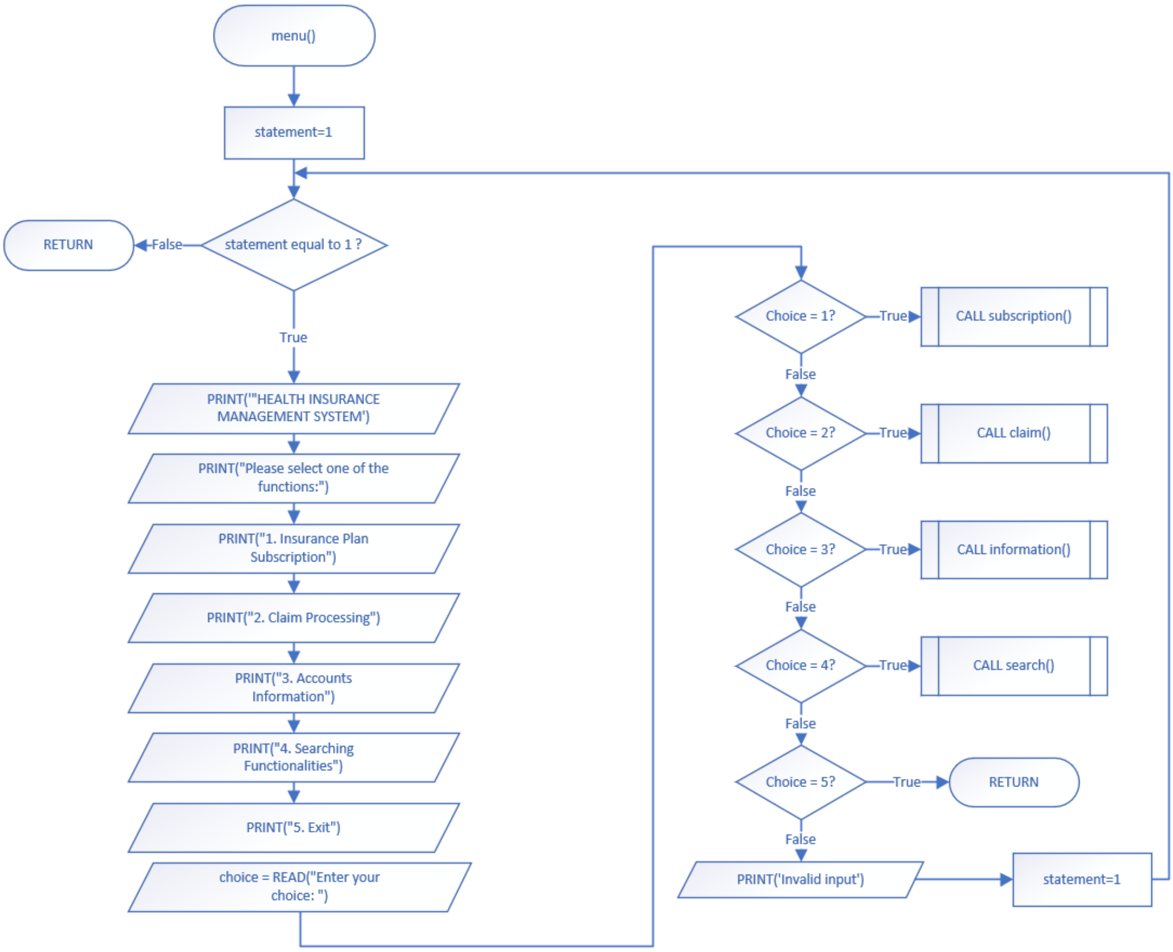


Figure 2.2.3: *Menu*

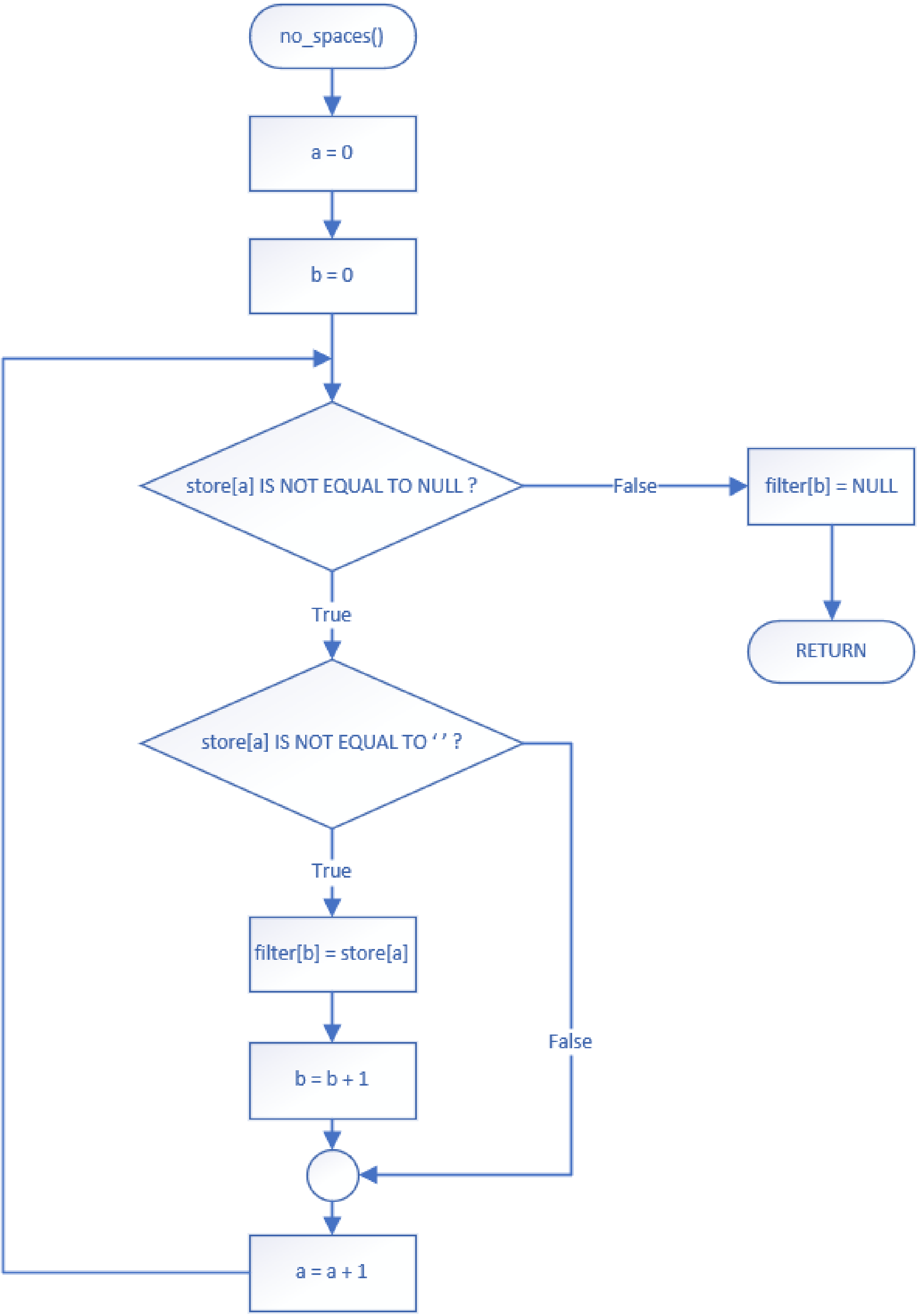


Figure 2.2.4: *Function to avoid space problem*

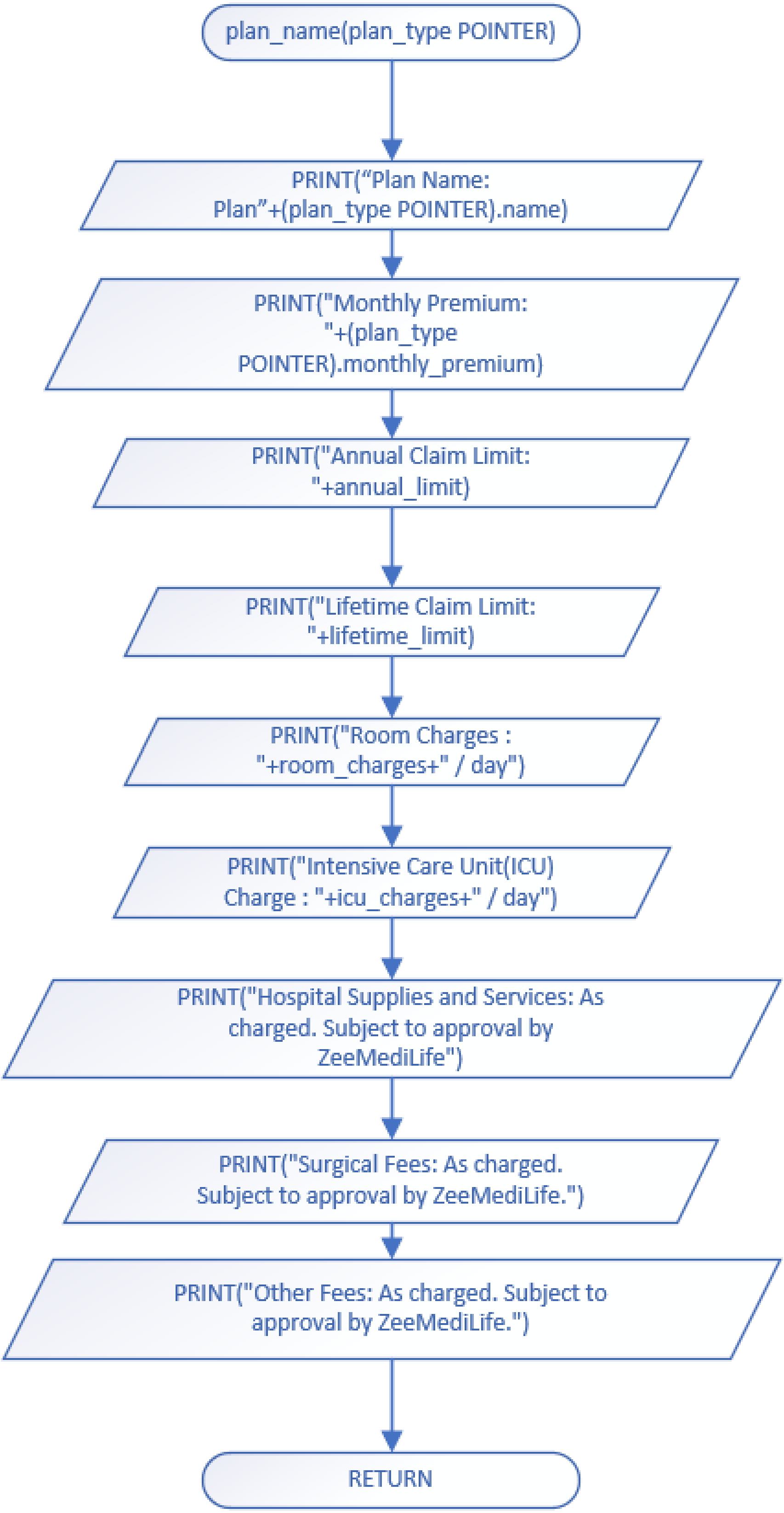

plan_name(plan_type POINTER)
PRINT(“Plan Name: Plan”+(plan_type POINTER).name)
PRINT("Monthly Premium: "+(plan_type POINTER).monthly_premium)
PRINT("Annual Claim Limit: "+annual_limit)
PRINT("Lifetime Claim Limit: "+lifetime_limit)
PRINT("Room Charges : "+room_charges+" / day")
PRINT("Intensive Care Unit(ICU) Charge : "+icu_charges+" / day")
PRINT("Hospital Supplies and Services: As charged. Subject to approval by ZeeMediLife")
PRINT("Surgical Fees: As charged. Subject to approval by ZeeMediLife.")
PRINT("Other Fees: As charged. Subject to approval by ZeeMediLife.")
RETURN

Figure 2.2.5: *Function to store plan details*

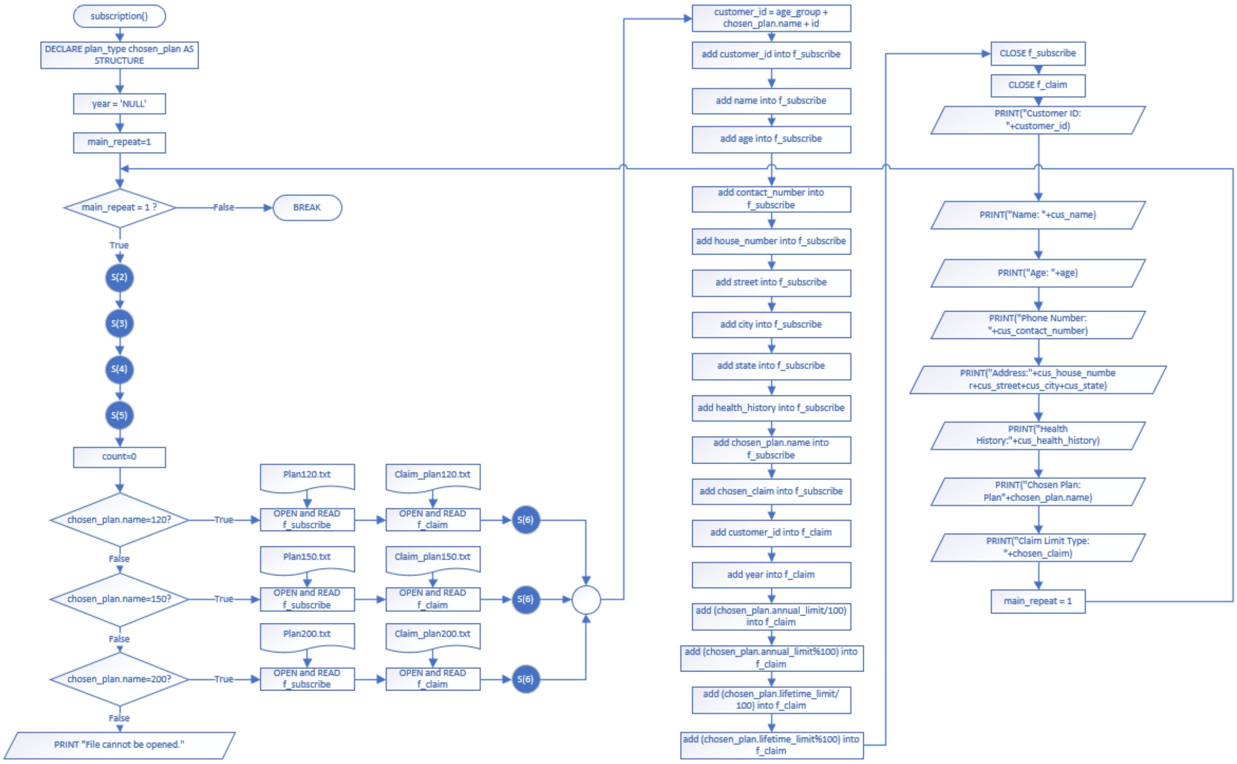

subscription()
DECLARE plan_type chosen_plan AS STRUCTURE
year = 'NULL'
main_repeat=1
main_repeat = 1 ?
False
BREAK
True
S(2)
S(3)
S(4)
S(5)
count=0
chosen_plan.name=120?
True
Plan120.txt
OPEN and READ f_subscribe
Claim_plan120.txt
OPEN and READ f_claim
S(6)
False
chosen_plan.name=150?
True
Plan150.txt
OPEN and READ f_subscribe
Claim_plan150.txt
OPEN and READ f_claim
S(6)
False
chosen_plan.name=200?
True
Plan200.txt
OPEN and READ f_subscribe
Claim_plan200.txt
OPEN and READ f_claim
S(6)
False
PRINT "File cannot be opened."
customer_id = age_group + chosen_plan.name + id
add customer_id into f_subscribe
add name into f_subscribe
add age into f_subscribe
add contact_number into f_subscribe
add house_number into f_subscribe
add street into f_subscribe
add city into f_subscribe
add state into f_subscribe
add health_history into f_subscribe
add chosen_plan.name into f_subscribe
add chosen_claim into f_subscribe
add customer_id into f_claim
add year into f_claim
add (chosen_plan.annual_limit/100) into f_claim
add (chosen_plan.annual_limit%100) into f_claim
add (chosen_plan.lifetime_limit/ 100) into f_claim
add (chosen_plan.lifetime_limit%100) into f_claim
CLOSE f_subscribe
CLOSE f_claim
PRINT("Customer ID: "+customer_id)
PRINT("Name: "+cus_name)
PRINT("Age: "+age)
PRINT("Phone Number: "+cus_contact_number)
PRINT("Address:"+cus_house_numbe r+cus_street+cus_city+cus_state)
PRINT("Health History:"+cus_health_history)
PRINT("Chosen Plan: Plan"+chosen_plan.name)
PRINT("Claim Limit Type: "+chosen_claim)
main_repeat = 1

Figure 2.2.6: *Subscription (Part 1)*

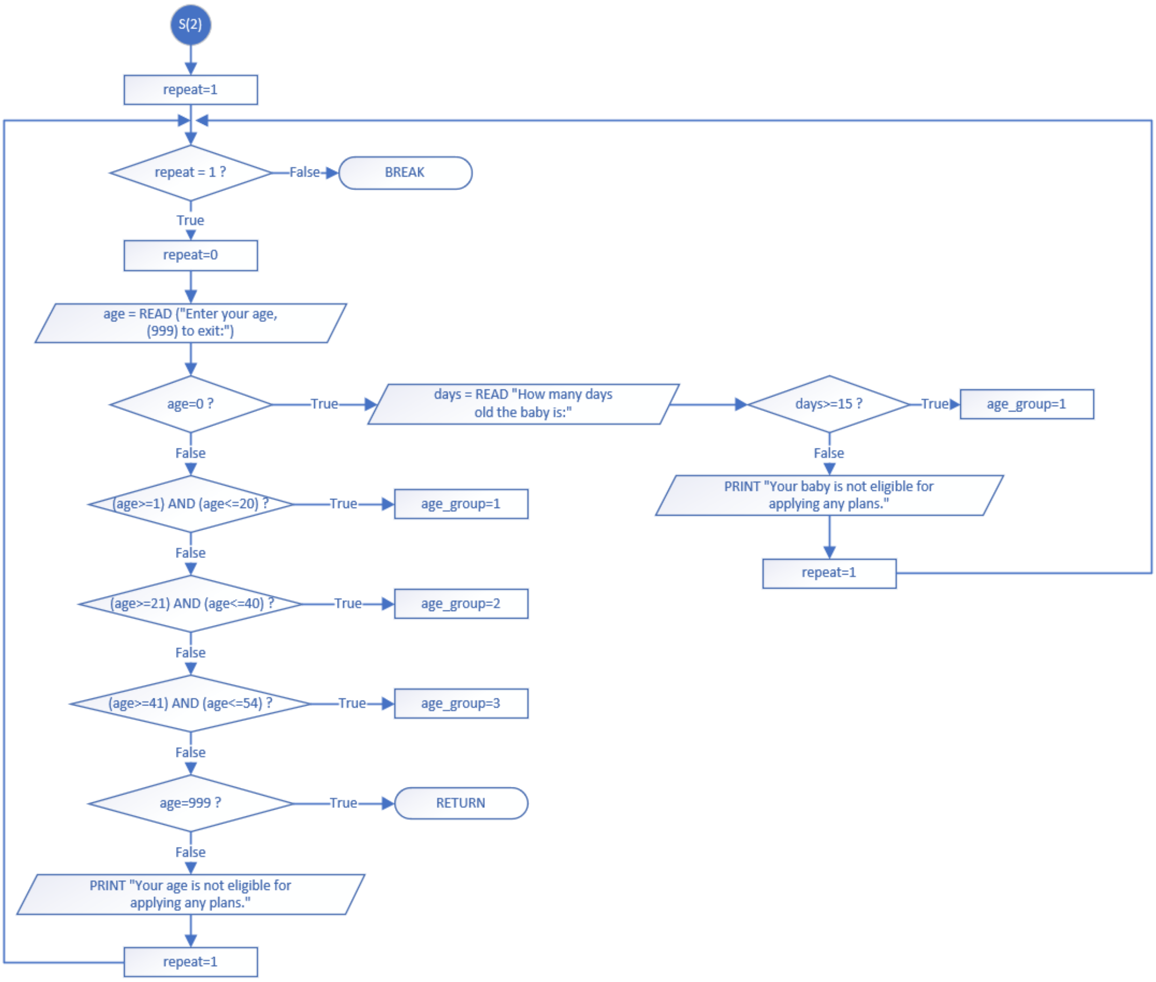

S(2)
repeat=1
repeat = 1 ?
False
BREAK
True
repeat=0
age = READ ("Enter your age, (999) to exit:")
age=0 ?
True
days = READ "How many days old the baby is:"
days>=15 ?
True
age_group=1
False
PRINT "Your baby is not eligible for applying any plans."
repeat=1
False
(age>=1) AND (age<=20) ?
True
age_group=1
False
(age>=21) AND (age<=40) ?
True
age_group=2
False
(age>=41) AND (age<=54) ?
True
age_group=3
False
age=999 ?
True
RETURN
False
PRINT "Your age is not eligible for applying any plans."
repeat=1

Figure 2.2.7: *Subscription (Part 2)*

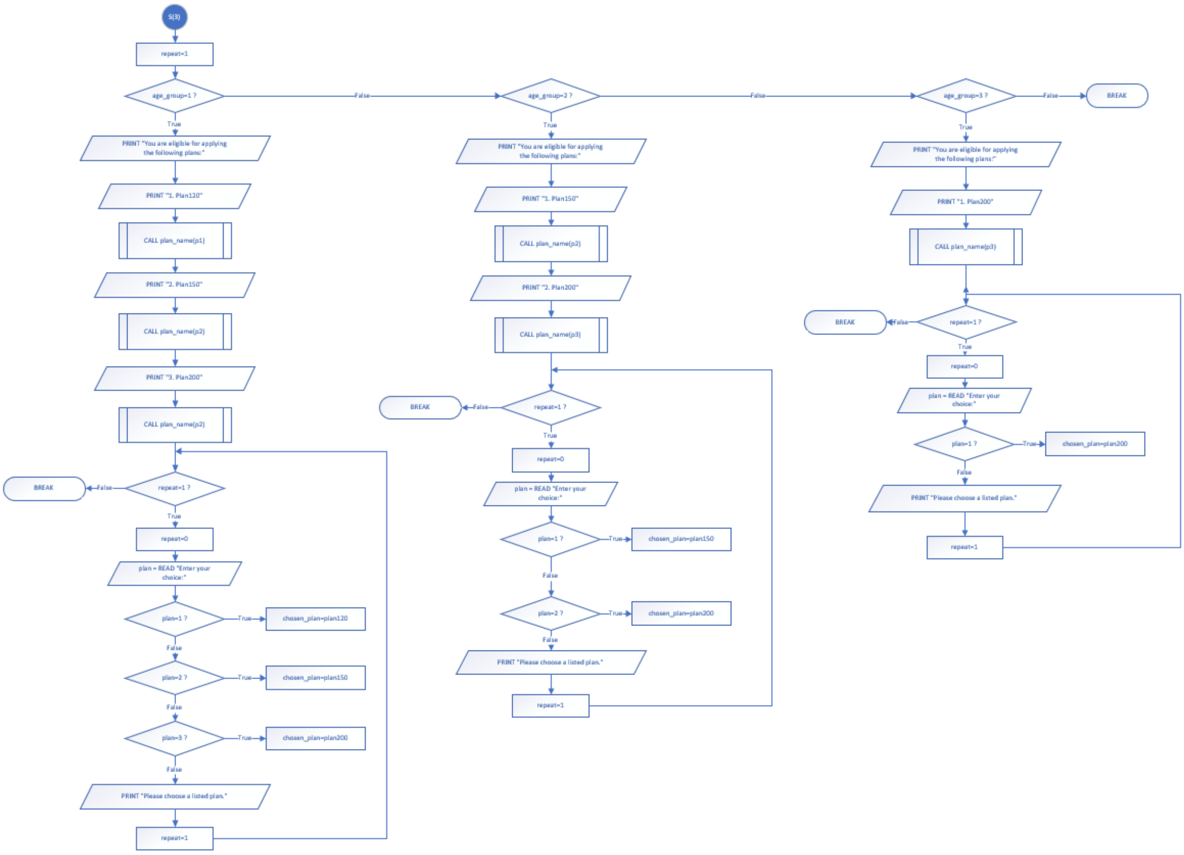


Figure 2.2.8: *Subscription (Part 3)*

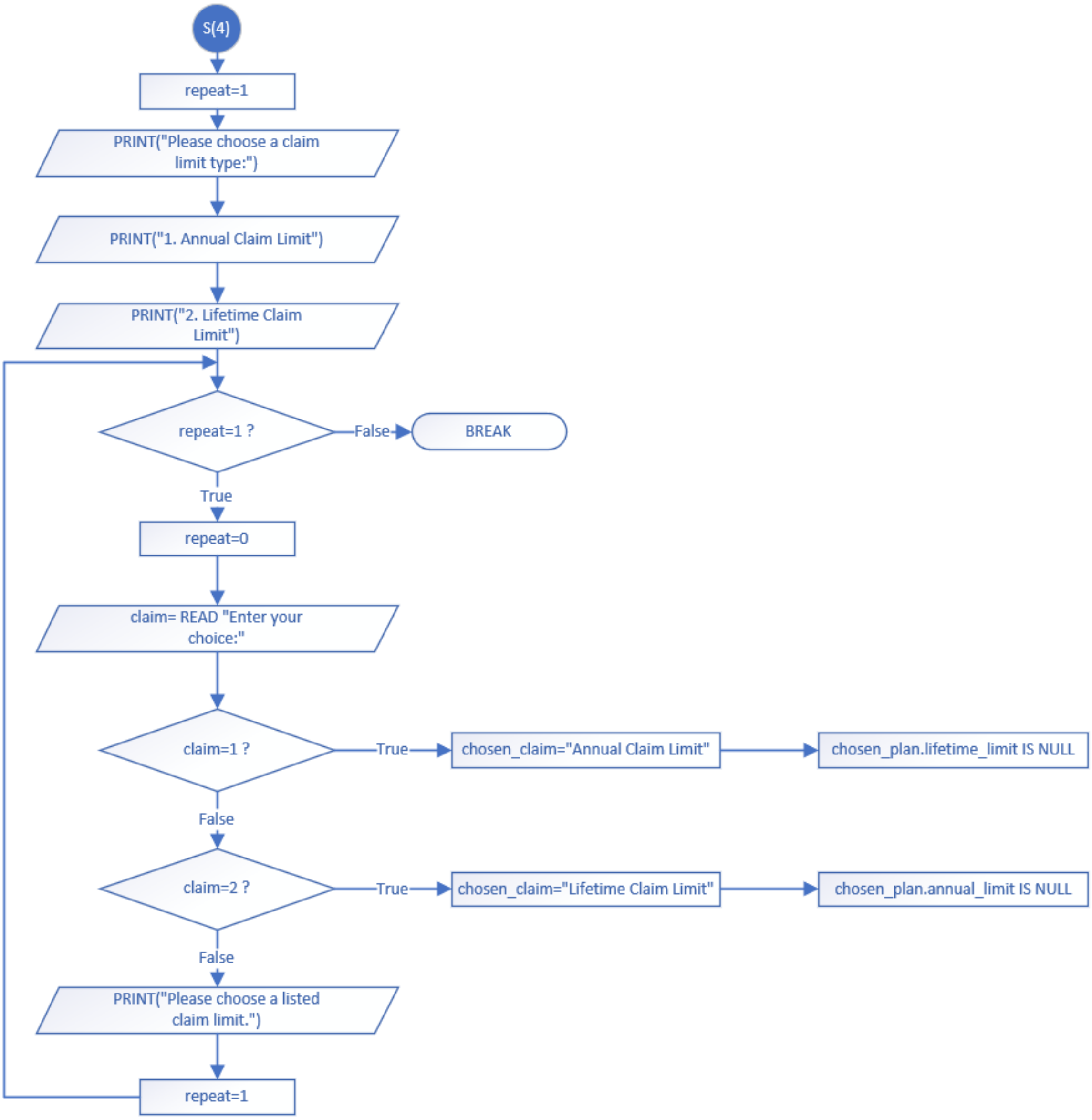

S(4)
repeat=1
PRINT("Please choose a claim limit type:")
PRINT("1. Annual Claim Limit")
PRINT("2. Lifetime Claim Limit")
repeat=1 ?
False
BREAK
True
repeat=0
claim= READ "Enter your choice:"
claim=1 ?
True
chosen_claim="Annual Claim Limit"
chosen_plan.lifetime_limit IS NULL
False
claim=2 ?
True
chosen_claim="Lifetime Claim Limit"
chosen_plan.annual_limit IS NULL
False
PRINT("Please choose a listed claim limit.")
repeat=1

Figure 2.2.9: *Subscription (Part 4)*

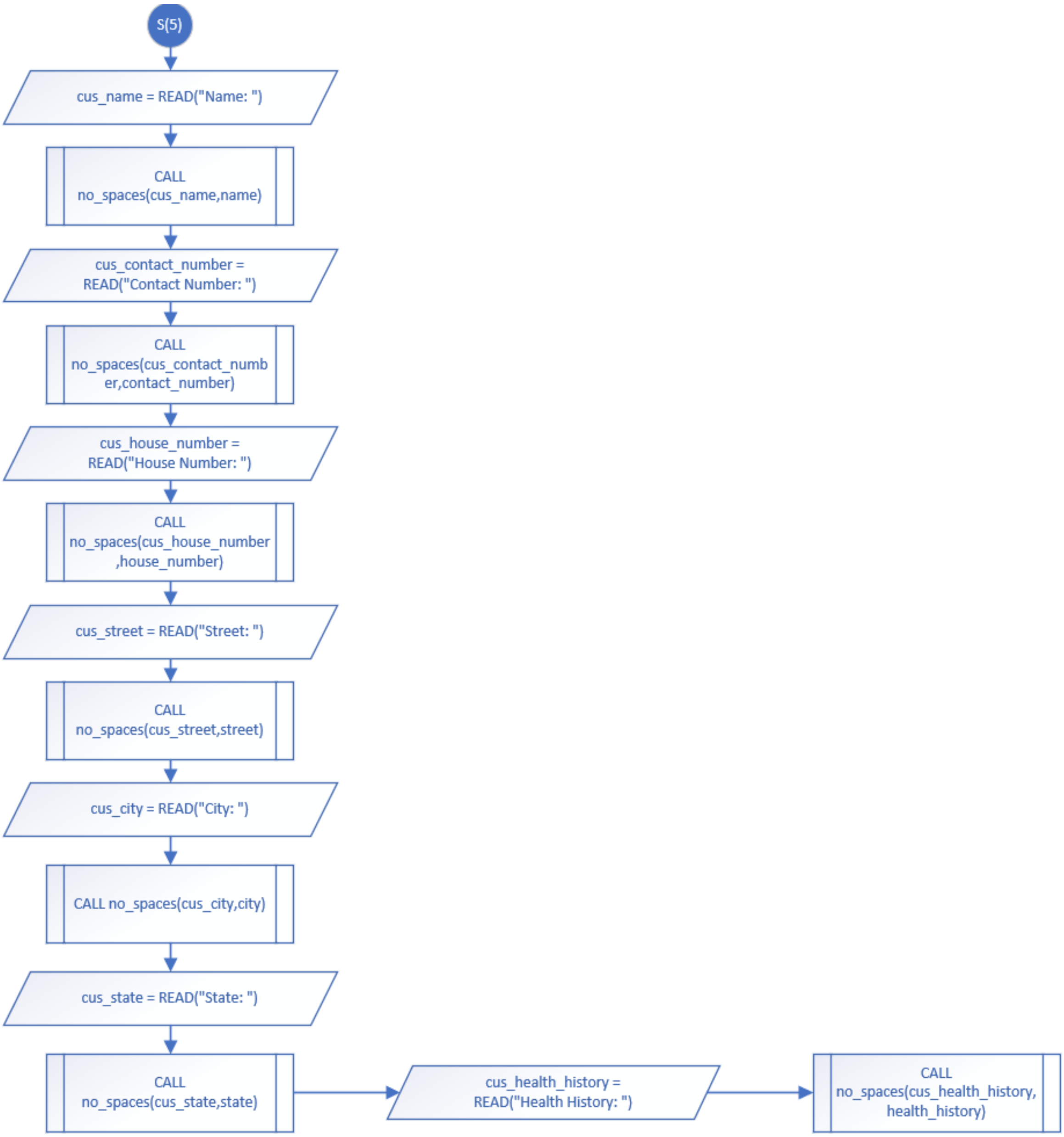

S(5)
cus_name = READ("Name: ")
CALL no_spaces(cus_name,name)
cus_contact_number = READ("Contact Number: ")
CALL no_spaces(cus_contact_number,contact_number)
cus_house_number = READ("House Number: ")
CALL no_spaces(cus_house_number,house_number)
cus_street = READ("Street: ")
CALL no_spaces(cus_street,street)
cus_city = READ("City: ")
CALL no_spaces(cus_city,city)
cus_state = READ("State: ")
CALL no_spaces(cus_state,state)
cus_health_history = READ("Health History: ")
CALL no_spaces(cus_health_history, health_history)

Figure 2.2.10: *Subscription (Part 5)*

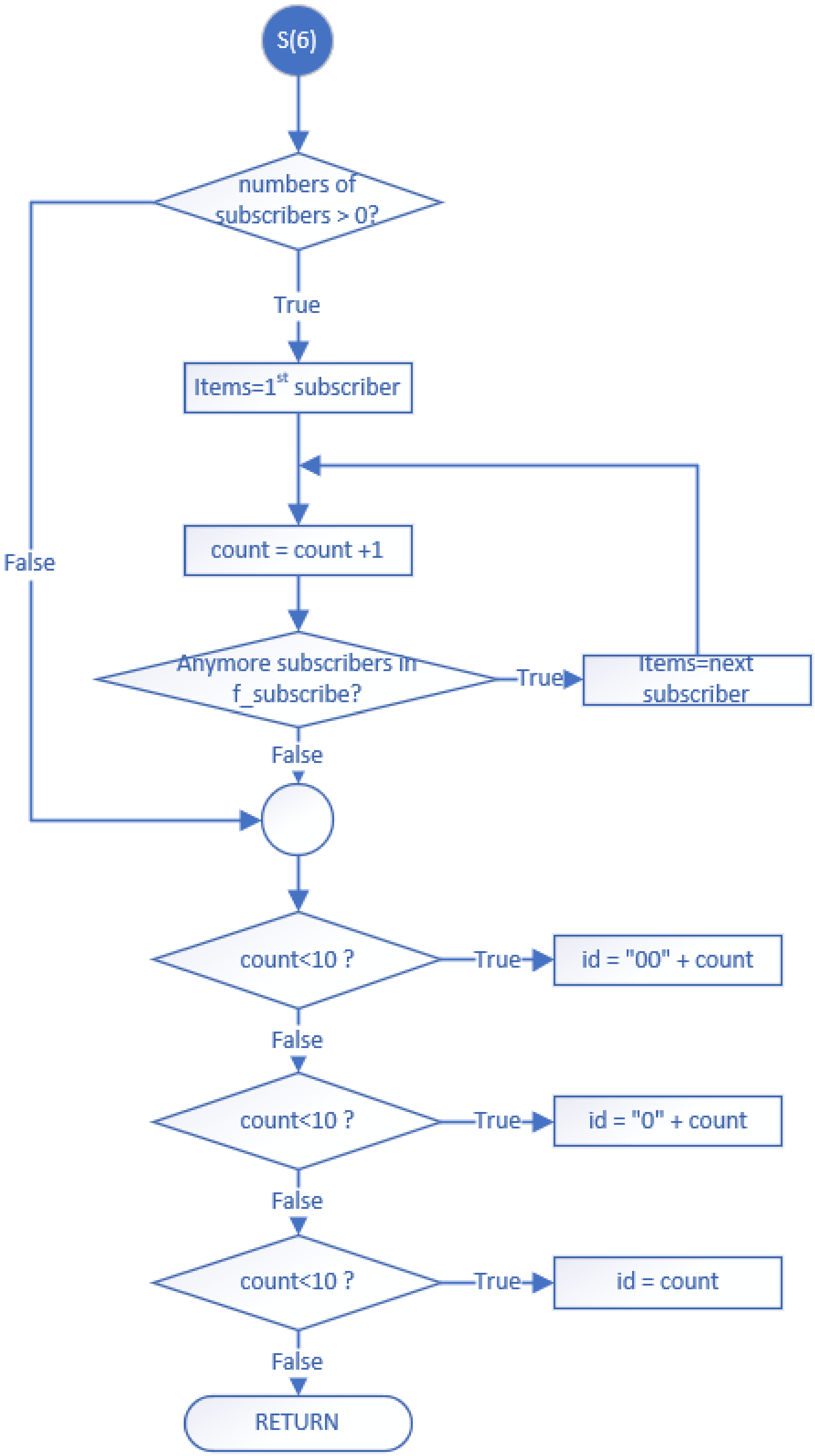
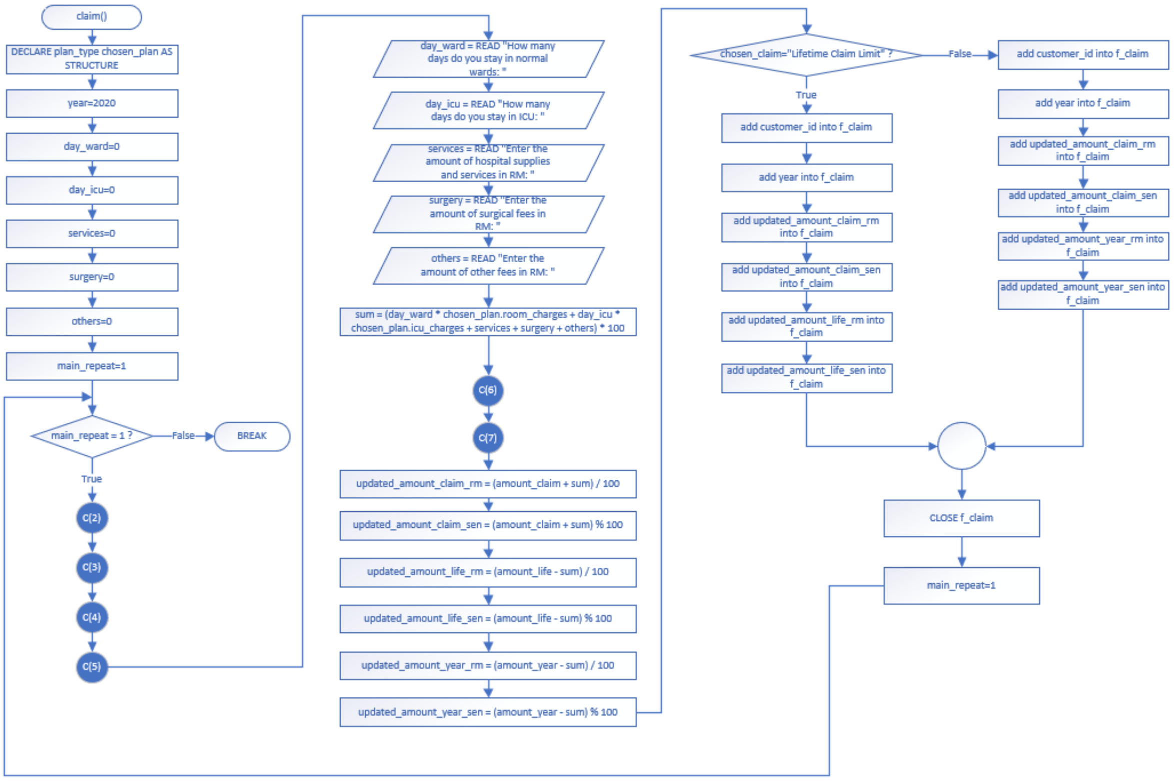


Figure 2.2.11: *Subscription (Part 6)*



claim()
DECLARE plan_type chosen_plan AS STRUCTURE
year=2020
day_ward=0
day_icu=0
services=0
surgery=0
others=0
main_repeat=1
main_repeat = 1 ?
False
BREAK
True
C(2)
C(3)
C(4)
C(5)
day_ward = READ "How many days do you stay in normal wards: "
day_icu = READ "How many days do you stay in ICU: "
services = READ "Enter the amount of hospital supplies and services in RM: "
surgery = READ "Enter the amount of surgical fees in RM: "
others = READ "Enter the amount of other fees in RM: "
sum = (day_ward * chosen_plan.room_charges + day_icu * chosen_plan.icu_charges + services + surgery + others) * 100
C(6)
C(7)
updated_amount_claim_rm = (amount_claim + sum) / 100
updated_amount_claim_sen = (amount_claim + sum) % 100
updated_amount_life_rm = (amount_life - sum) / 100
updated_amount_life_sen = (amount_life - sum) % 100
updated_amount_year_rm = (amount_year - sum) / 100
updated_amount_year_sen = (amount_year - sum) % 100
chosen_claim="Lifetime Claim Limit" ?
True
False
add customer_id into f_claim
add year into f_claim
add updated_amount_claim_rm into f_claim
add updated_amount_claim_sen into f_claim
add updated_amount_life_rm into f_claim
add updated_amount_life_sen into f_claim
add customer_id into f_claim
add year into f_claim
add updated_amount_claim_rm into f_claim
add updated_amount_claim_sen into f_claim
add updated_amount_year_rm into f_claim
add updated_amount_year_sen into f_claim
CLOSE f_claim
main_repeat=1

Figure 2.2.12: *Claim (Part 1)*

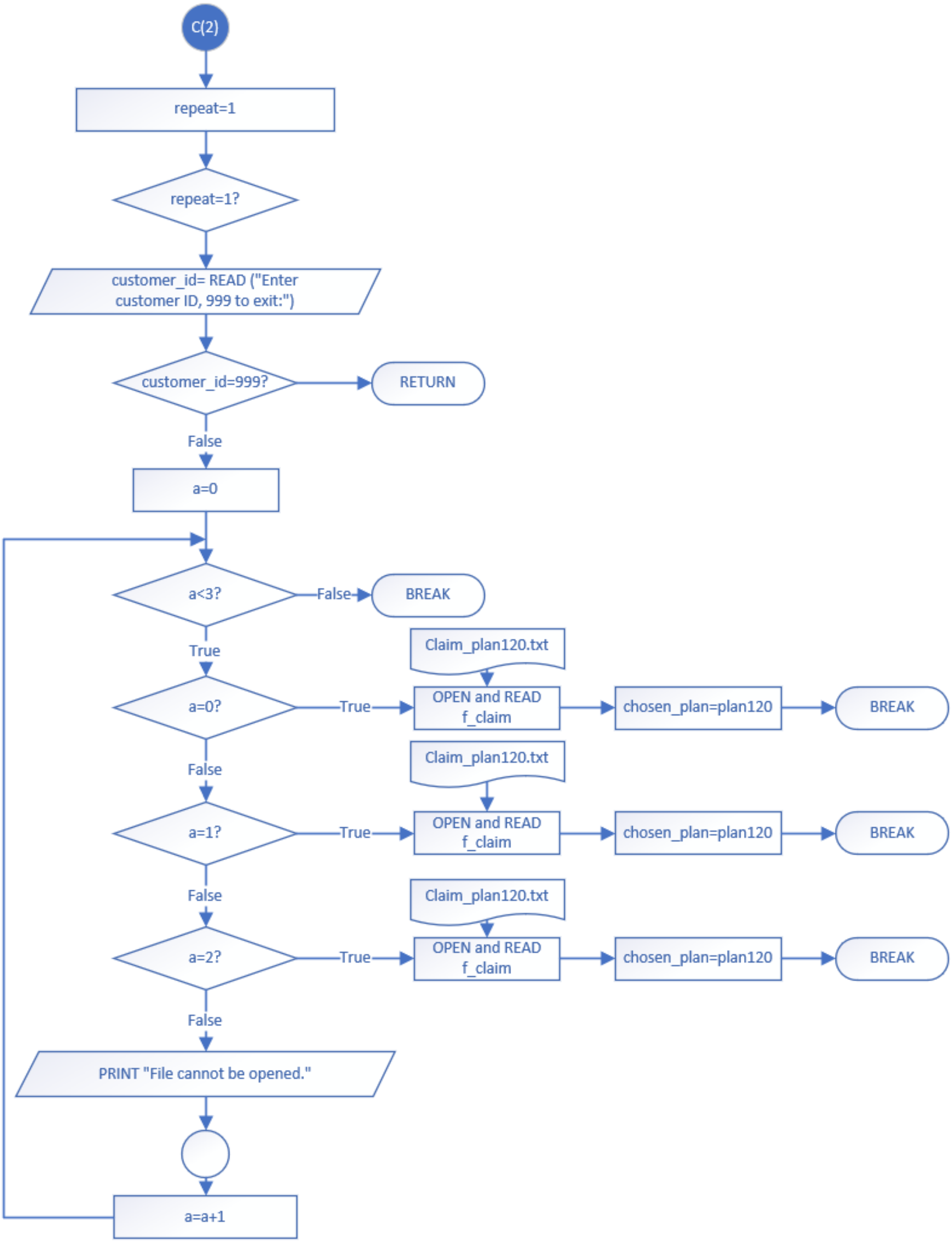


Figure 2.2.13: *Claim (Part 2)*

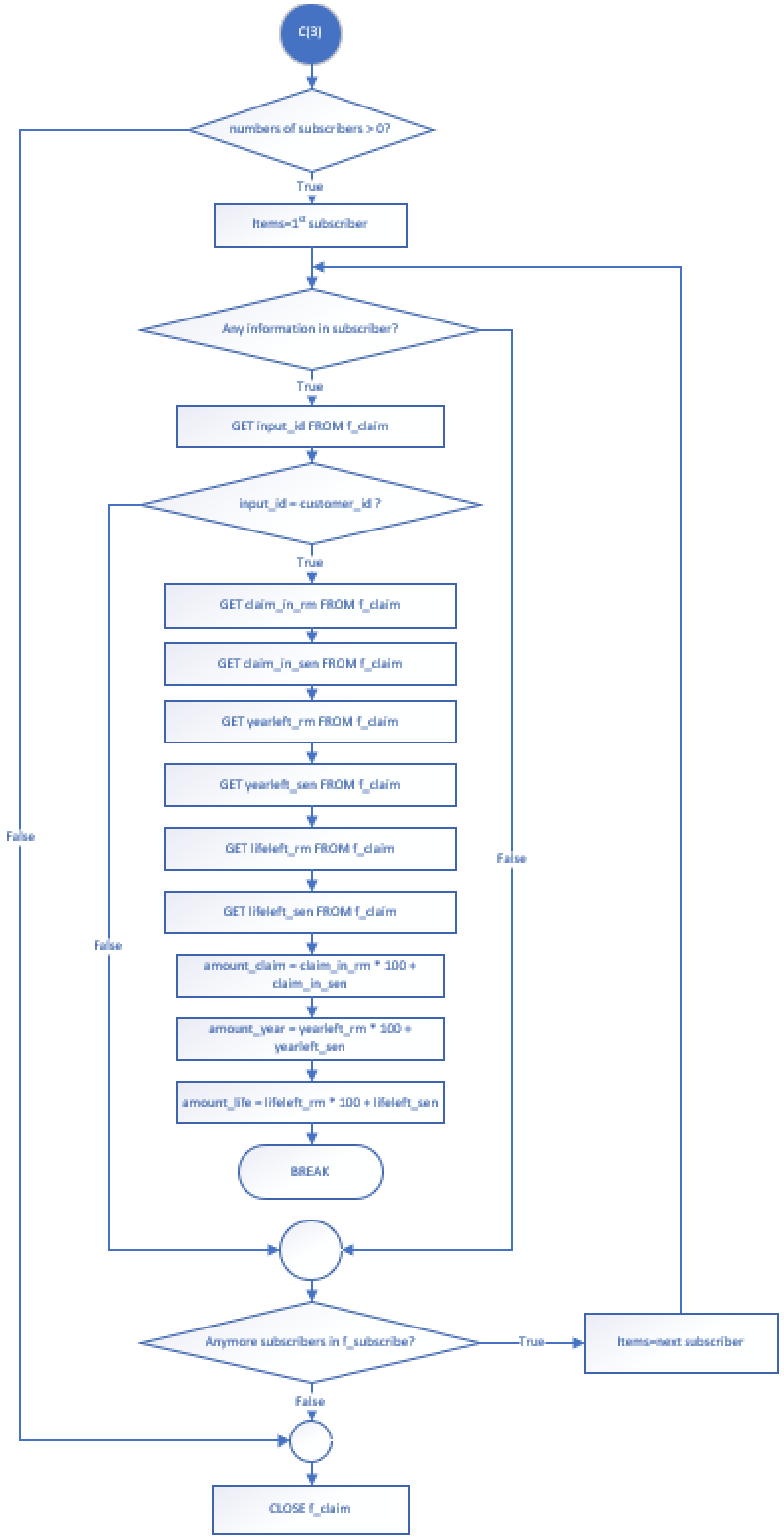

C[3]
numbers of subscribers > 0?
True
Items=1st subscriber
Any information in subscriber?
True
GET input_id FROM f_claim
input_id = customer_id ?
True
GET claim_in_rm FROM f_claim
GET claim_in_sen FROM f_claim
GET yearleft_rm FROM f_claim
GET yearleft_sen FROM f_claim
GET lifeleft_rm FROM f_claim
GET lifeleft_sen FROM f_claim
amount_claim = claim_in_rm * 100 + claim_in_sen
amount_year = yearleft_rm * 100 + yearleft_sen
amount_life = lifeleft_rm * 100 + lifeleft_sen
BREAK
False
False
False
Anymore subscribers in f_subscribe?
True
Items=next subscriber
False
CLOSE f_claim

Figure 2.2.14: *Claim (Part 3)*

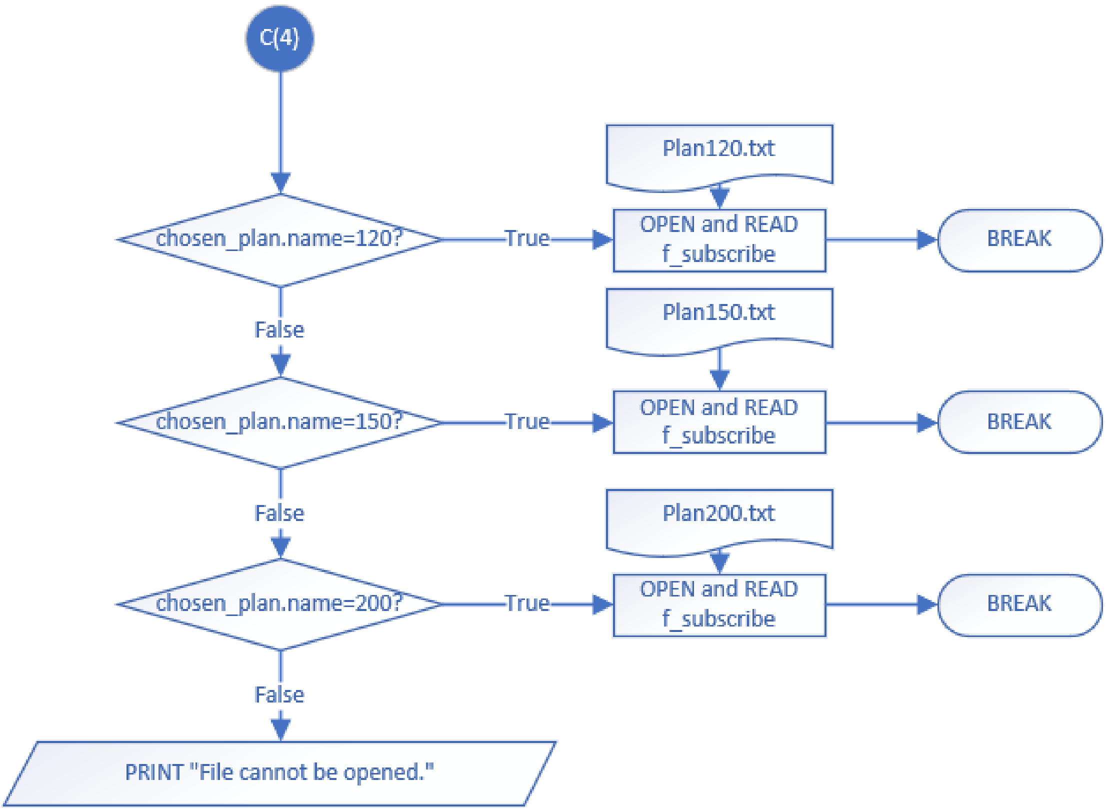


Figure 2.2.15: *Claim (Part 4)*

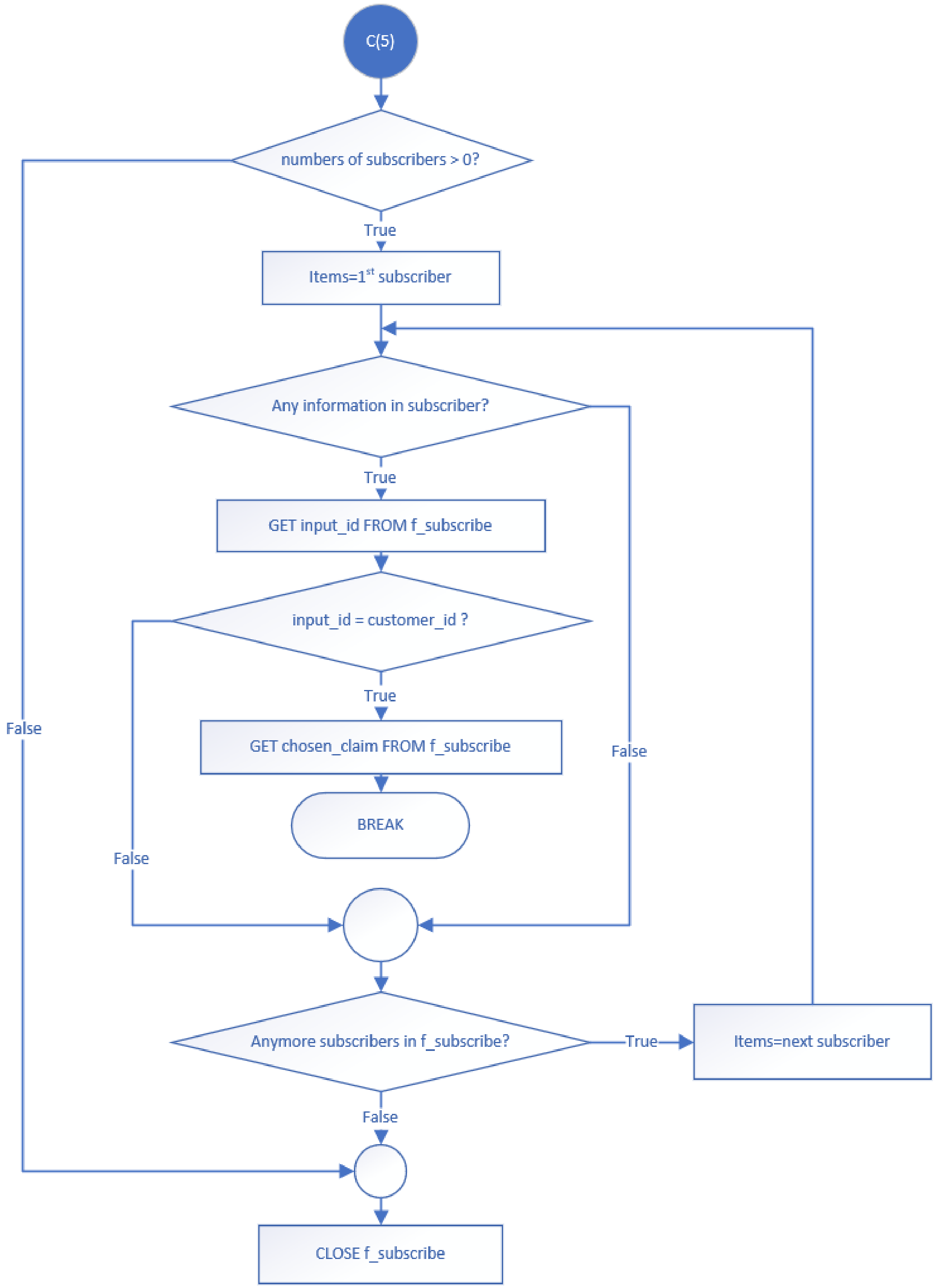

C(5)
numbers of subscribers > 0?
True
Items=1st subscriber
Any information in subscriber?
True
GET input_id FROM f_subscribe
input_id = customer_id ?
True
GET chosen_claim FROM f_subscribe
BREAK
False
False
False
Anymore subscribers in f_subscribe?
True
Items=next subscriber
False
CLOSE f_subscribe

Figure 2.2.16: *Claim (Part 5)*

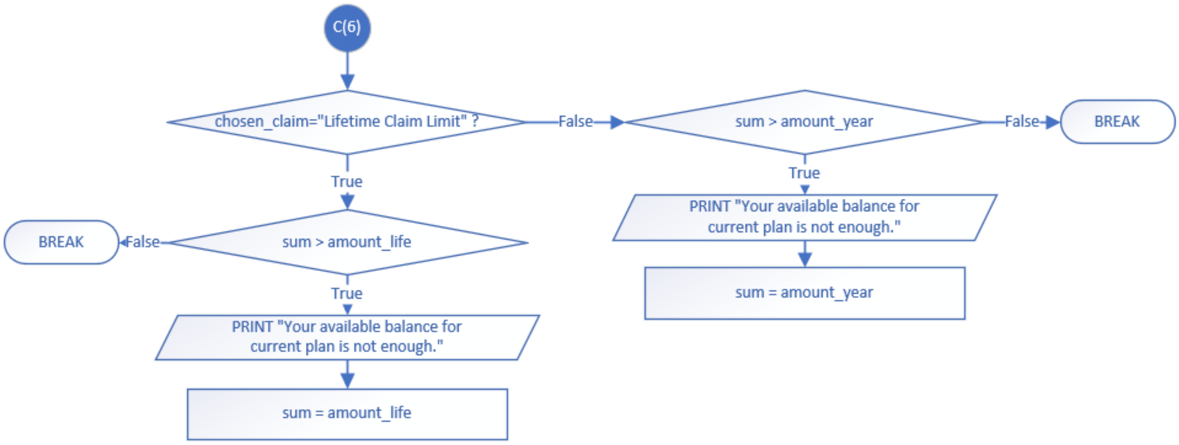

C(6)
chosen_claim="Lifetime Claim Limit" ?
False
sum > amount_year
False
BREAK
True
PRINT "Your available balance for current plan is not enough."
sum = amount_year
True
BREAK
False
sum > amount_life
True
PRINT "Your available balance for current plan is not enough."
sum = amount_life

Figure 2.2.17: *Claim (Part 6)*

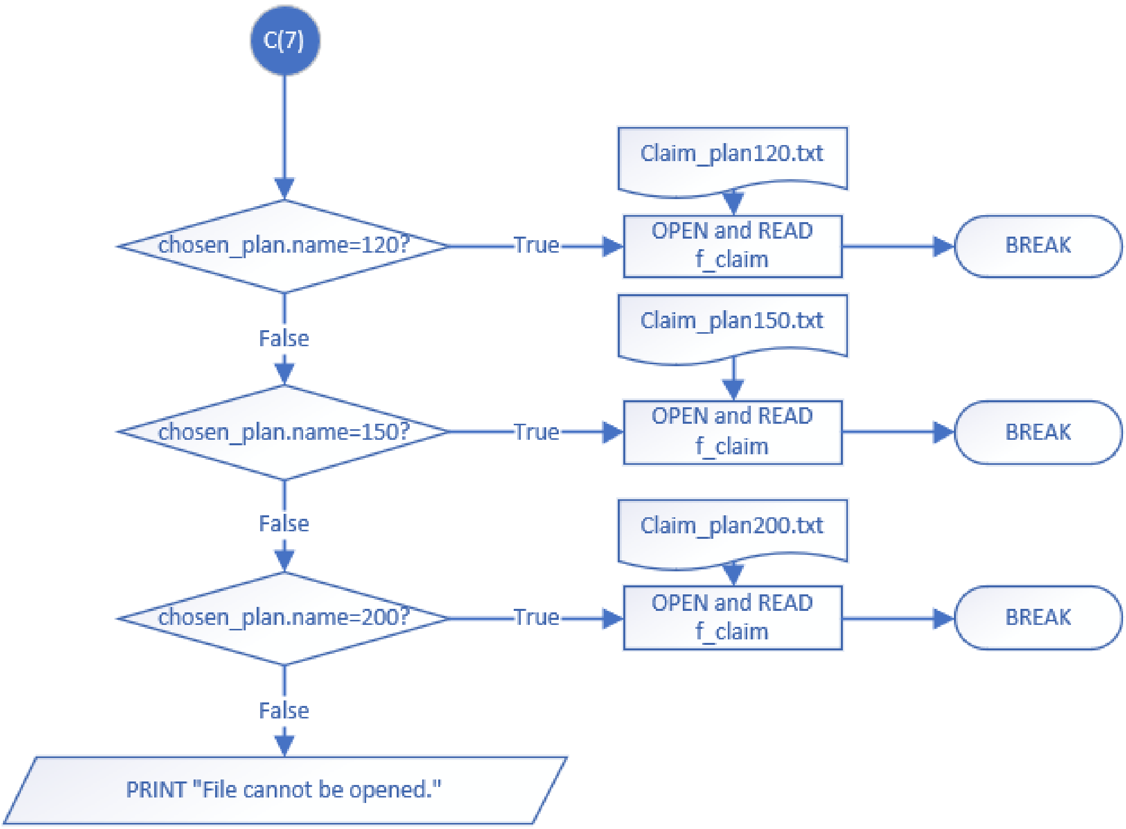


Figure 2.2.18: *Claim (Part 7)*

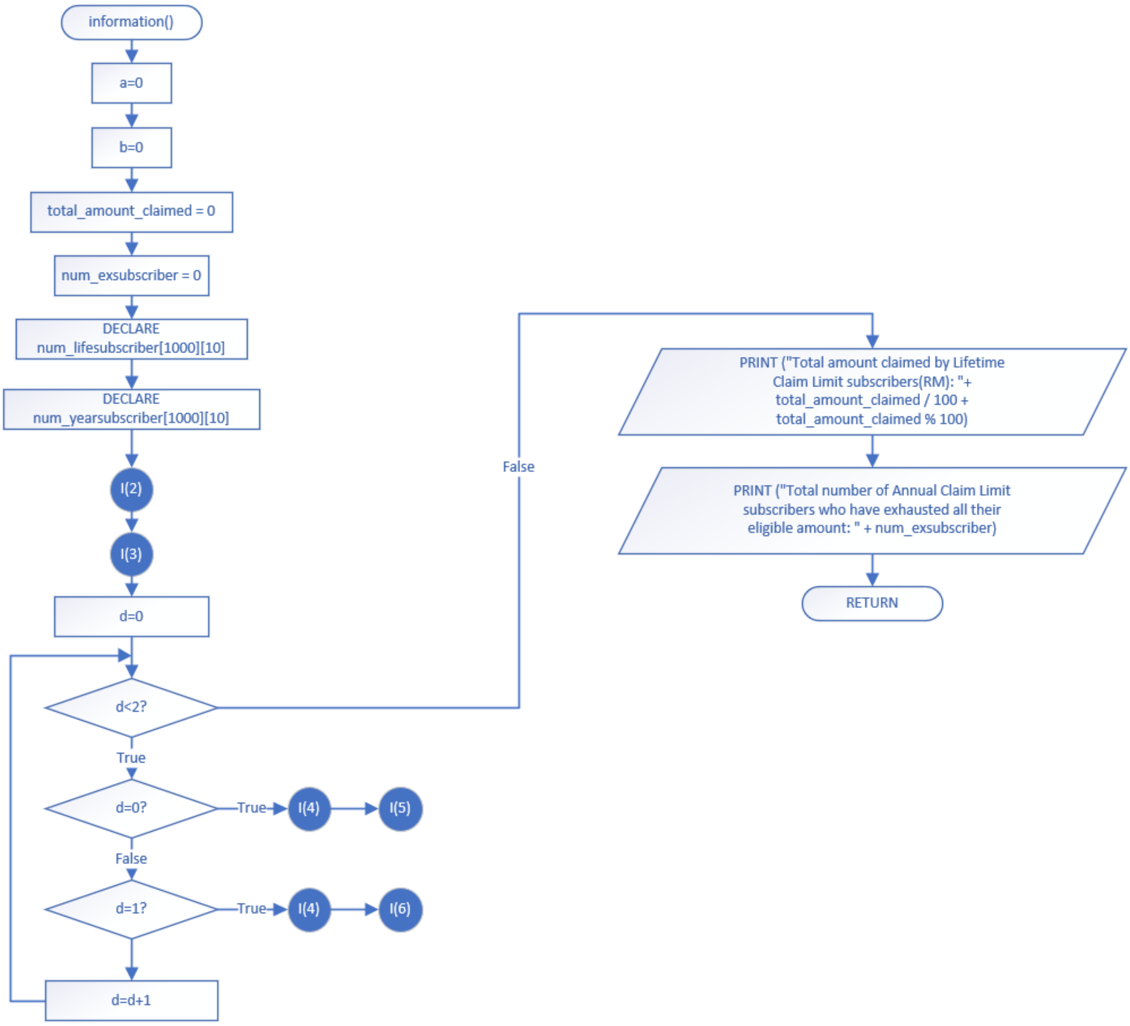

information()
a=0
b=0
total_amount_claimed = 0
num_exsubscriber = 0
DECLARE num_lifesubscriber[1000][10]
DECLARE num_yearsubscriber[1000][10]
I(2)
I(3)
d=0
d<2?
True
d=0?
True
I(4)
I(5)
False
d=1?
True
I(4)
I(6)
d=d+1
False
PRINT ("Total amount claimed by Lifetime Claim Limit subscribers(RM): "+ total_amount_claimed / 100 + total_amount_claimed % 100)
PRINT ("Total number of Annual Claim Limit subscribers who have exhausted all their eligible amount: " + num_exsubscriber)
RETURN

Figure 2.2.19: *Account Information (Part 1)*

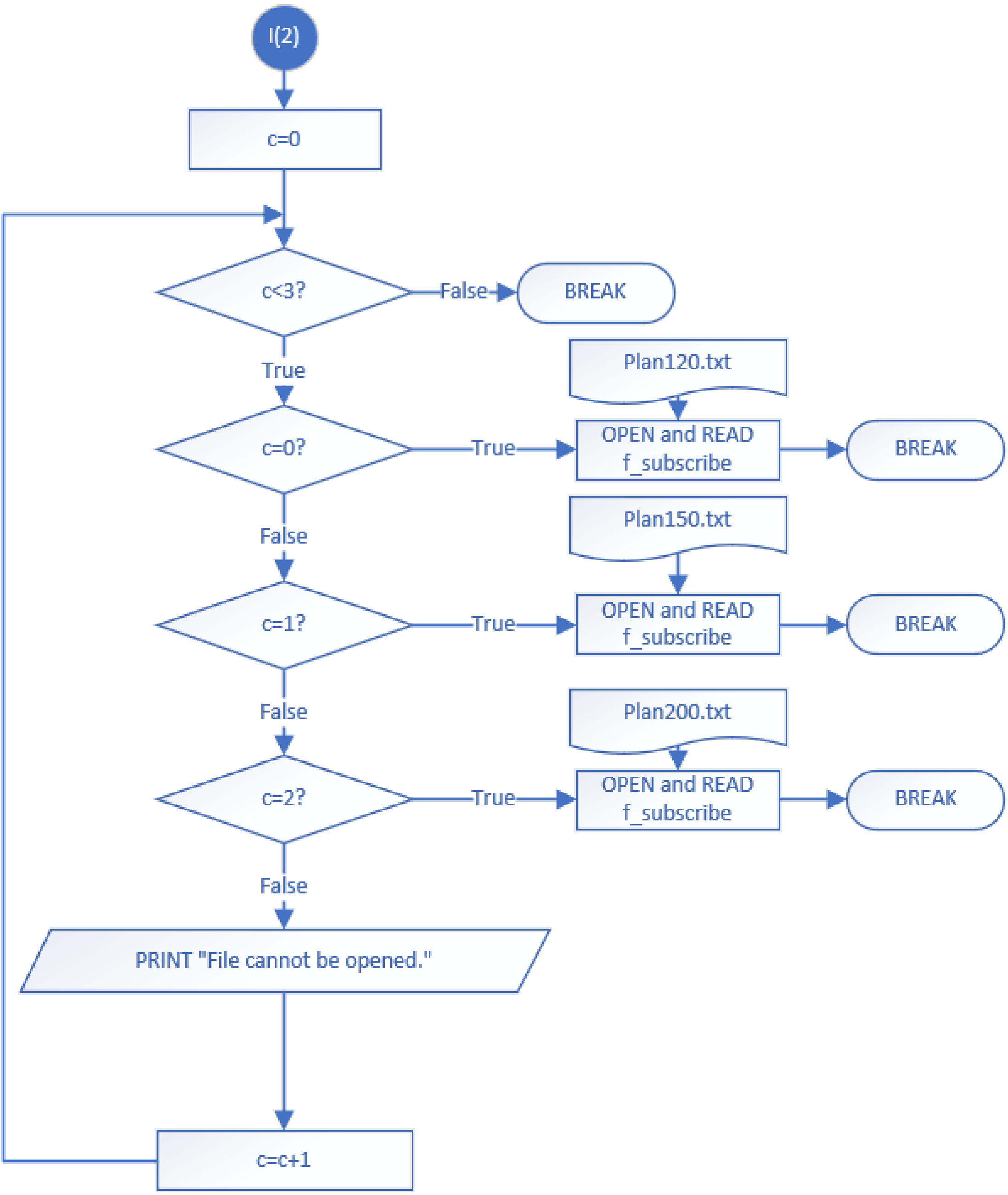


Figure 2.2.20: *Account Information (Part 2)*

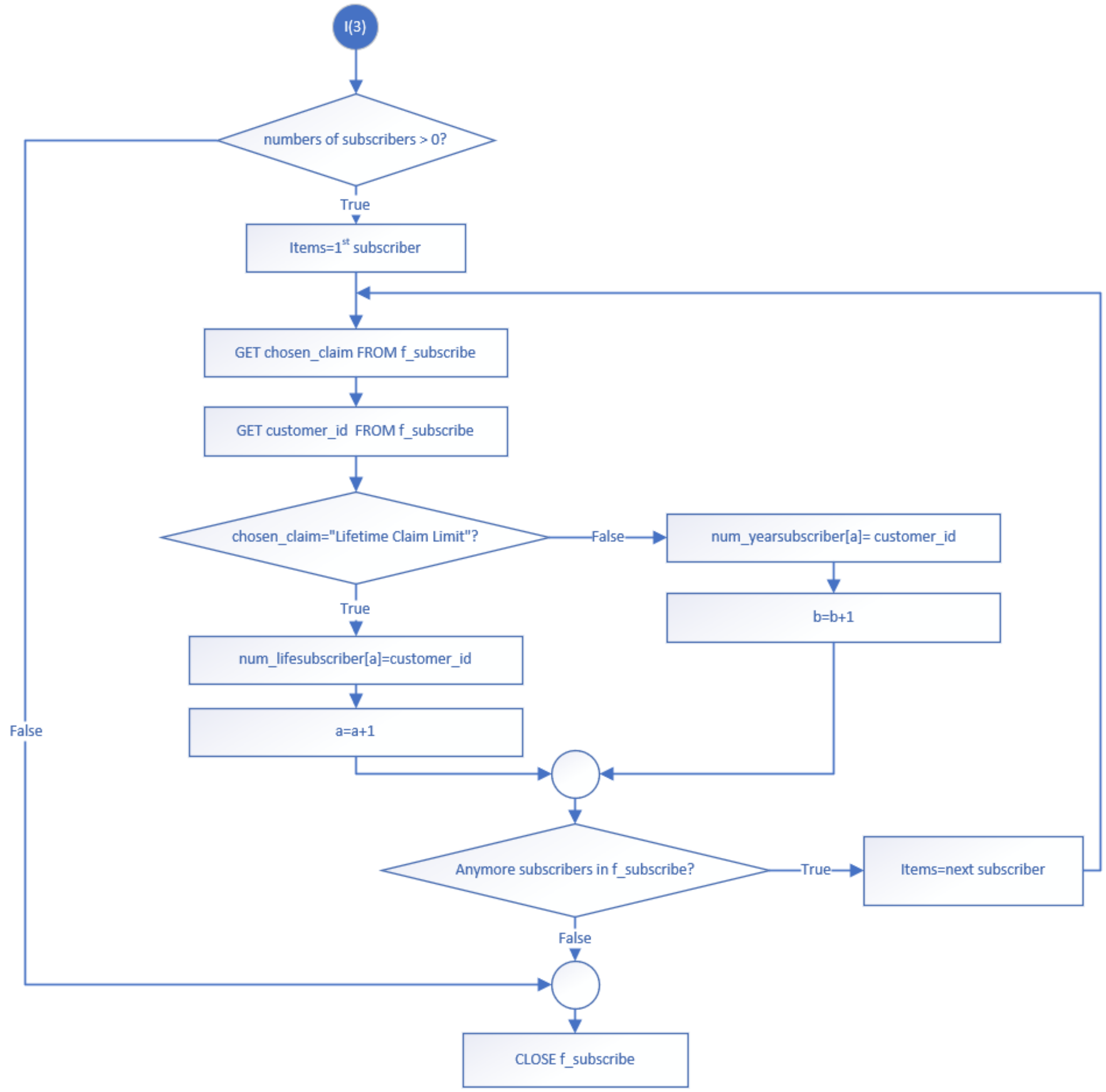

I(3)
numbers of subscribers > 0?
True
Items=1st subscriber
GET chosen_claim FROM f_subscribe
GET customer_id FROM f_subscribe
chosen_claim="Lifetime Claim Limit"?
False
num_yearsubscriber[a]= customer_id
b=b+1
True
num_lifesubscriber[a]=customer_id
a=a+1
False
Anymore subscribers in f_subscribe?
True
Items=next subscriber
False
CLOSE f_subscribe

Figure 2.2.21: *Account Information (Part 3)*

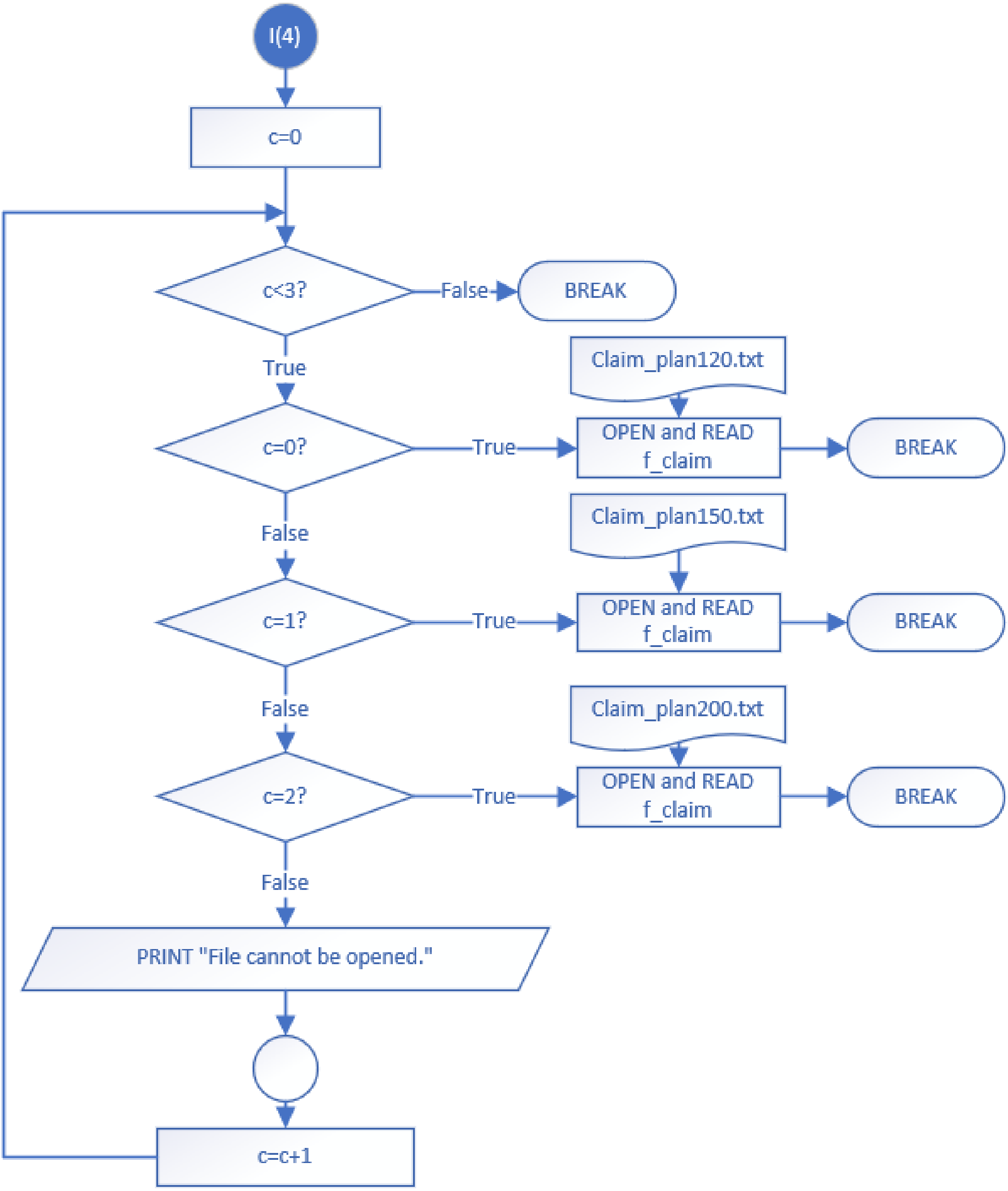


Figure 2.2.22: *Account Information (Part 4)*

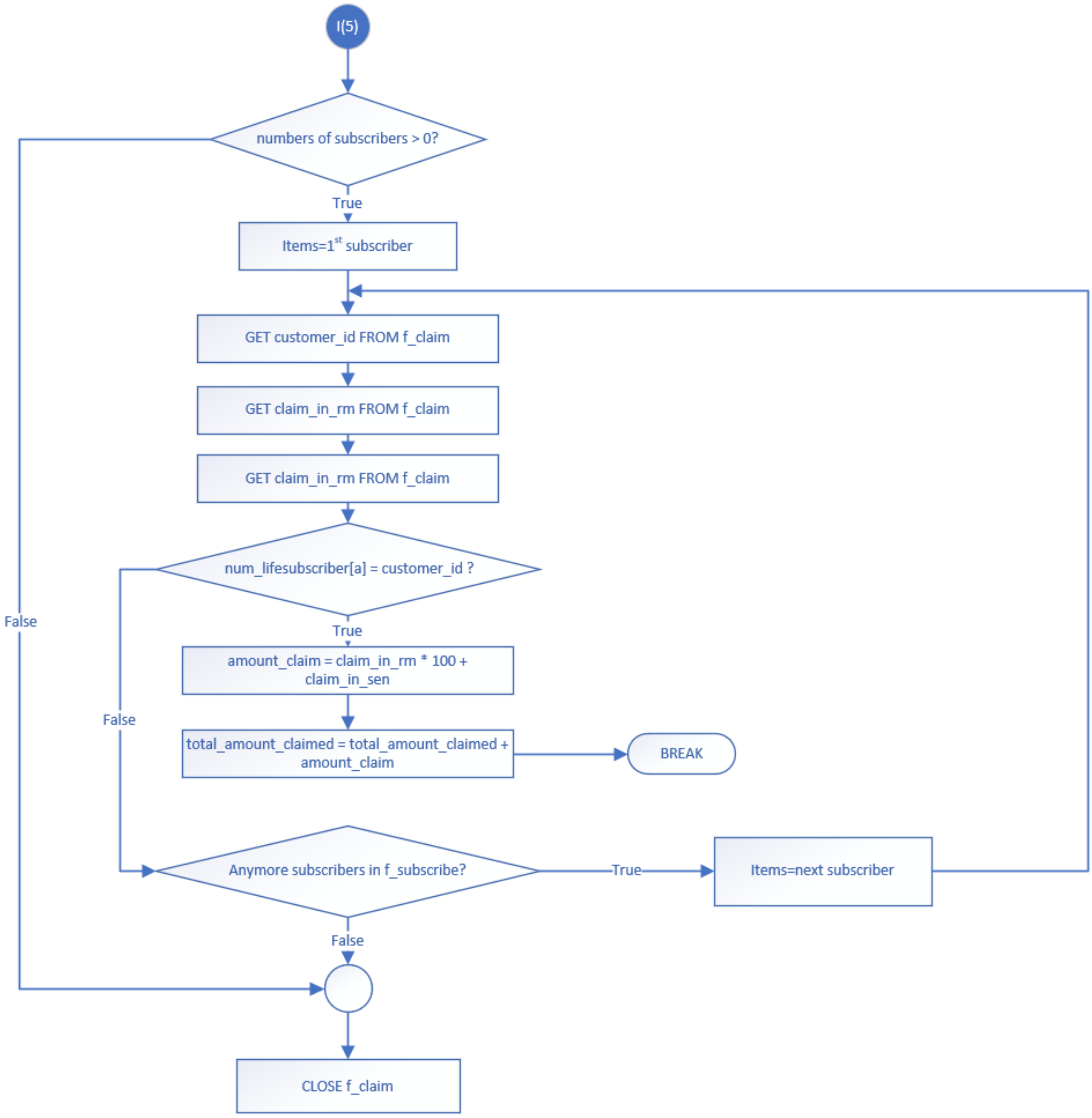

I(5)
numbers of subscribers > 0?
True
Items=1st subscriber
GET customer_id FROM f_claim
GET claim_in_rm FROM f_claim
GET claim_in_rm FROM f_claim
num_lifesubscriber[a] = customer_id ?
True
amount_claim = claim_in_rm * 100 + claim_in_sen
total_amount_claimed = total_amount_claimed + amount_claim
BREAK
False
False
Anymore subscribers in f_subscribe?
True
Items=next subscriber
False
CLOSE f_claim

Figure 2.2.23: *Account Information (Part 5)*

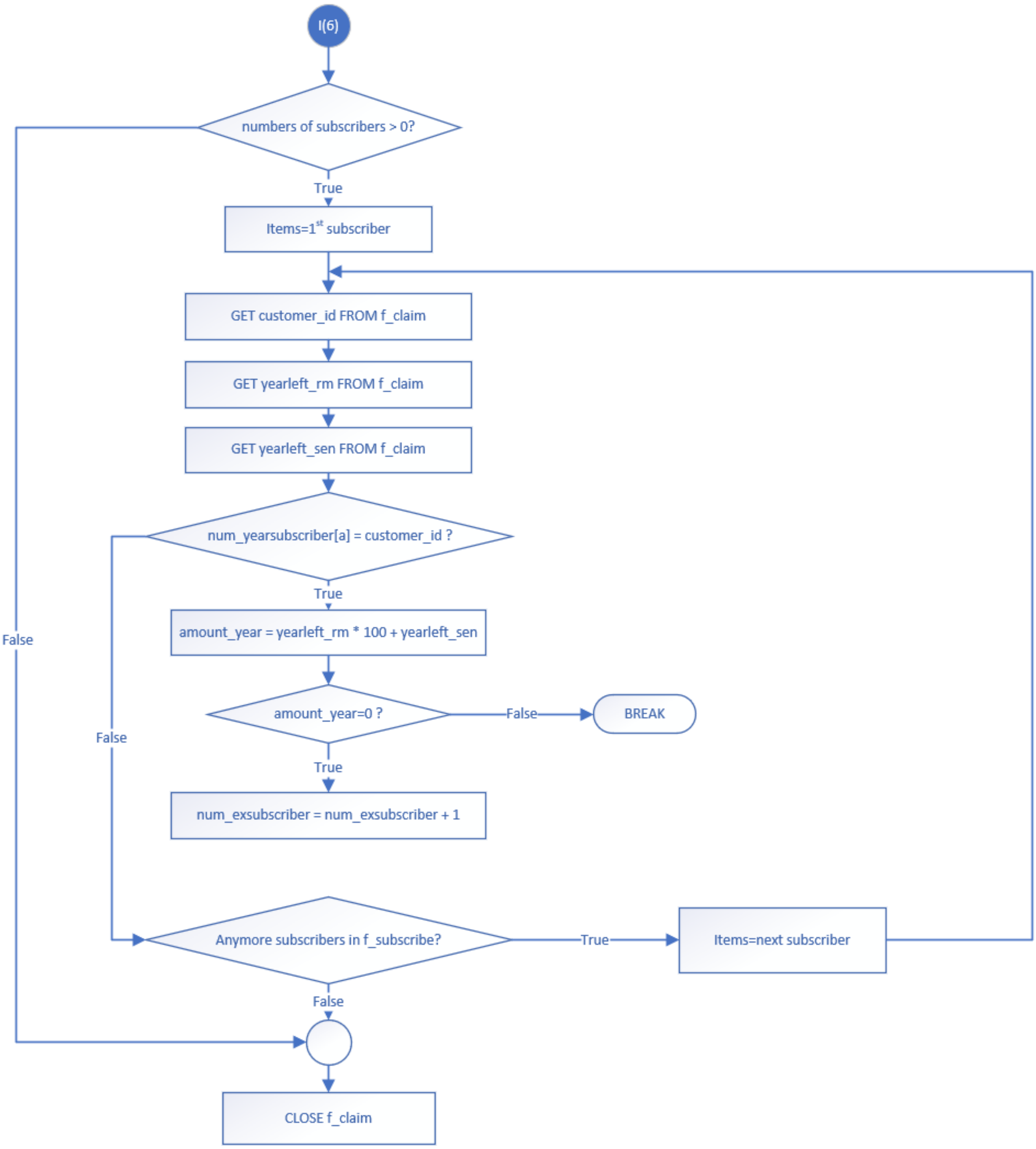

I(6)
numbers of subscribers > 0?
True
Items=1st subscriber
GET customer_id FROM f_claim
GET yearleft_rm FROM f_claim
GET yearleft_sen FROM f_claim
num_yearsubscriber[a] = customer_id ?
True
amount_year = yearleft_rm * 100 + yearleft_sen
amount_year=0 ?
False
BREAK
True
num_exsubscriber = num_exsubscriber + 1
False
False
Anymore subscribers in f_subscribe?
True
Items=next subscriber
False
CLOSE f_claim

Figure 2.2.24: *Account Information (Part 6)*

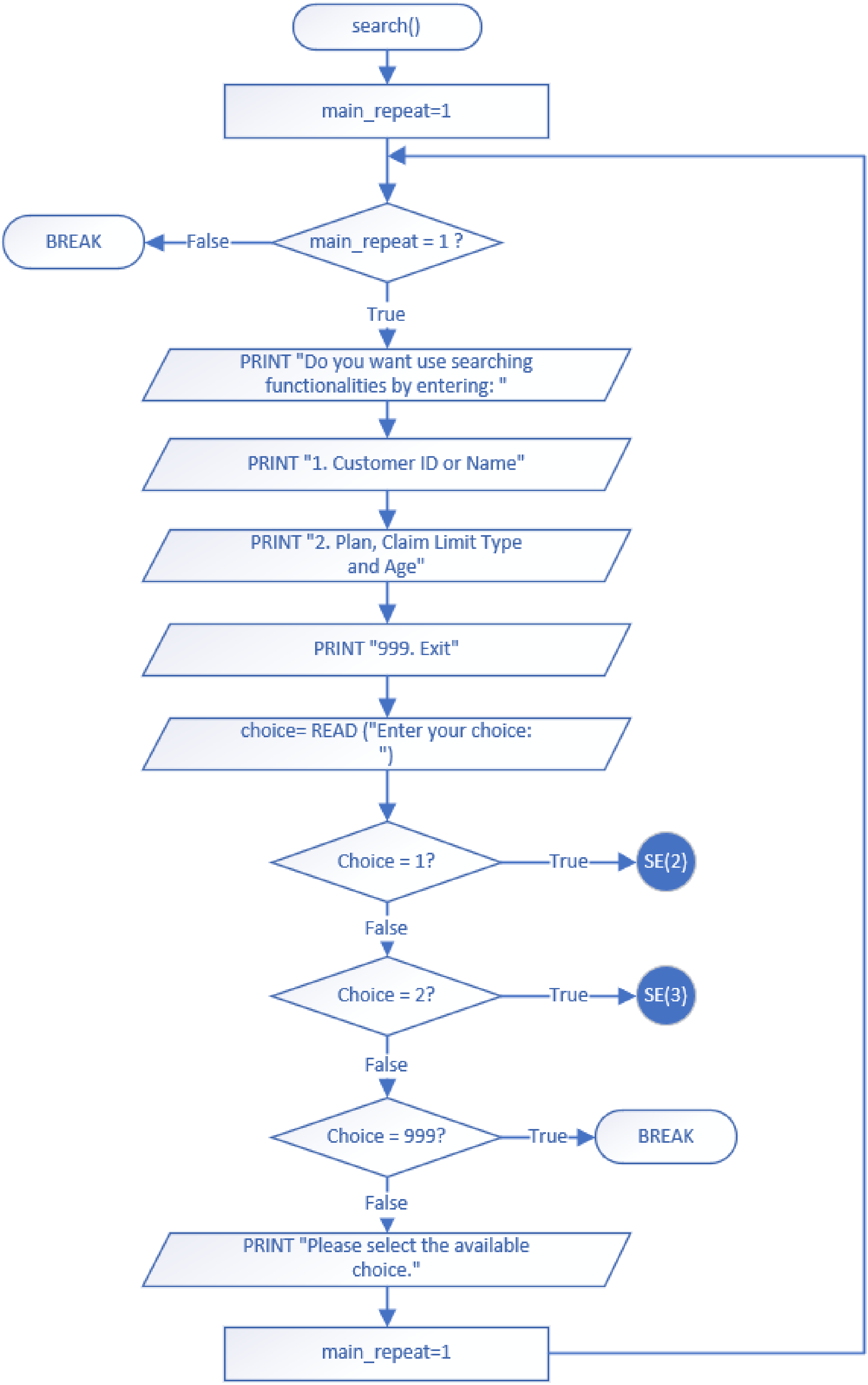

search()
main_repeat=1
BREAK
False
main_repeat = 1 ?
True
PRINT "Do you want use searching functionalities by entering: "
PRINT "1. Customer ID or Name"
PRINT "2. Plan, Claim Limit Type and Age"
PRINT "999. Exit"
choice= READ ("Enter your choice: ")
Choice = 1?
True
SE(2)
False
Choice = 2?
True
SE(3)
False
Choice = 999?
True
BREAK
False
PRINT "Please select the available choice."
main_repeat=1

Figure 2.2.25: *Searching Functionalities (Part 1)*

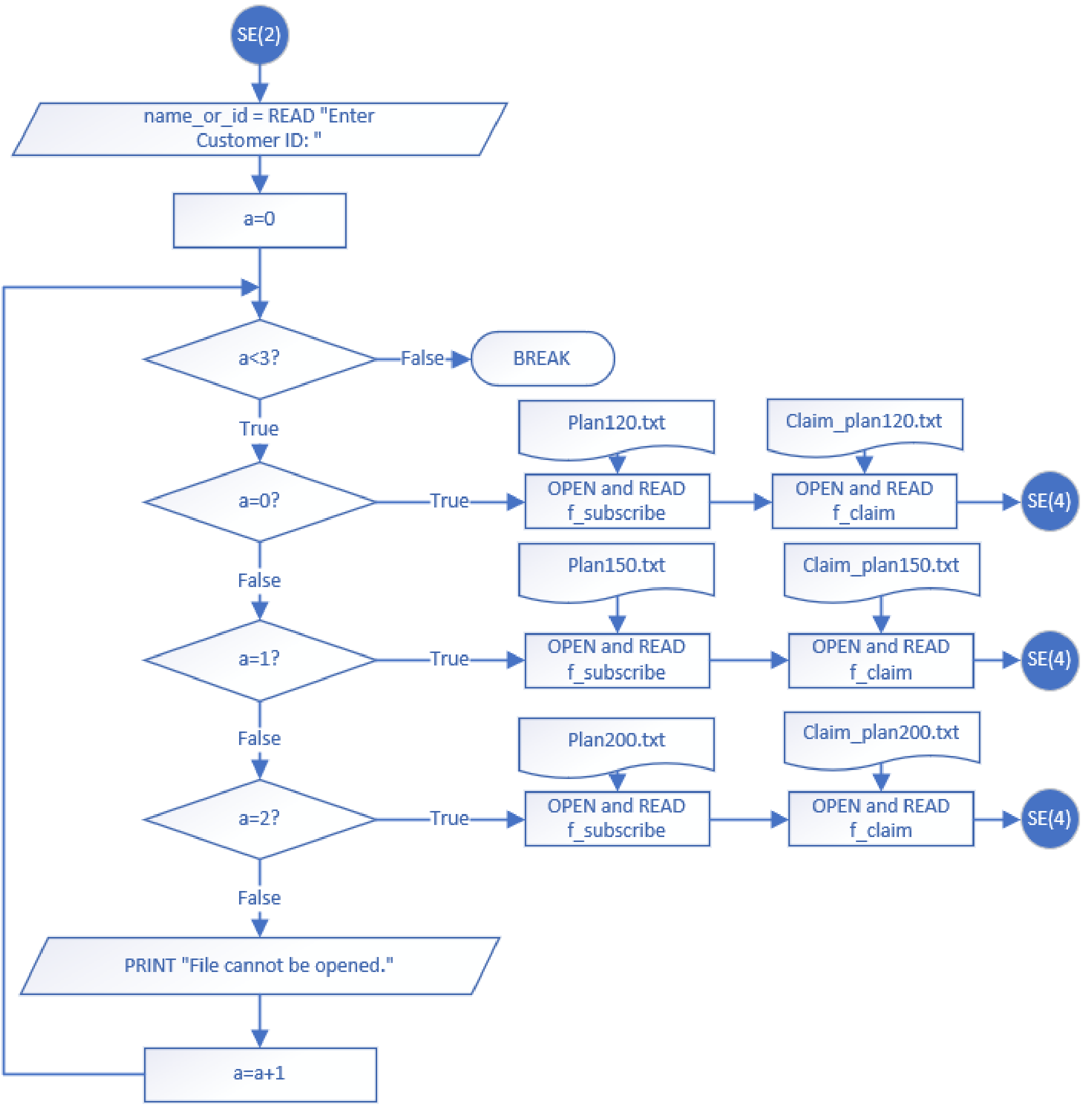


Figure 2.2.26: *Searching Functionalities (Part 2)*

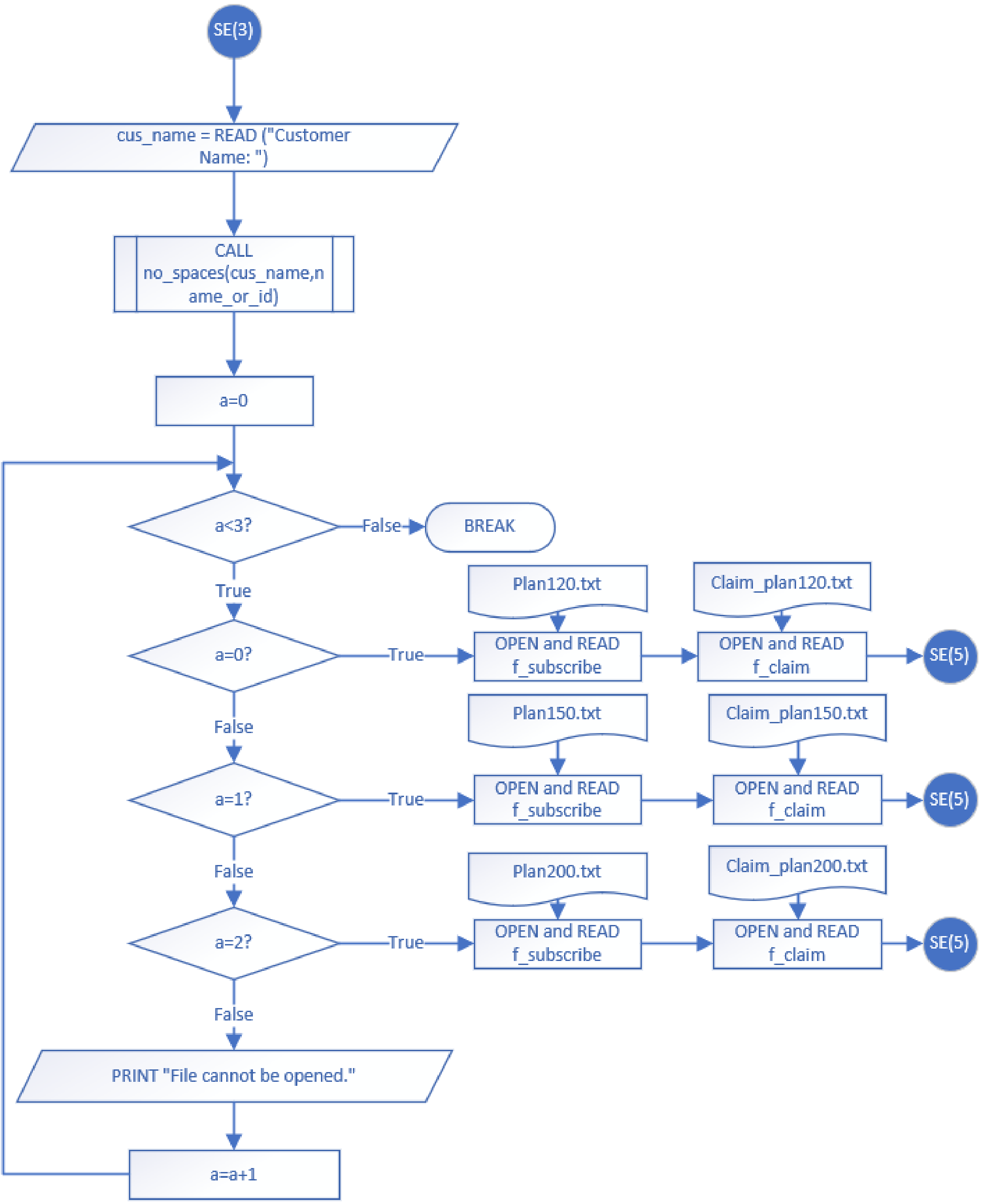


Figure 2.2.27: *Searching Functionalities (Part 3)*

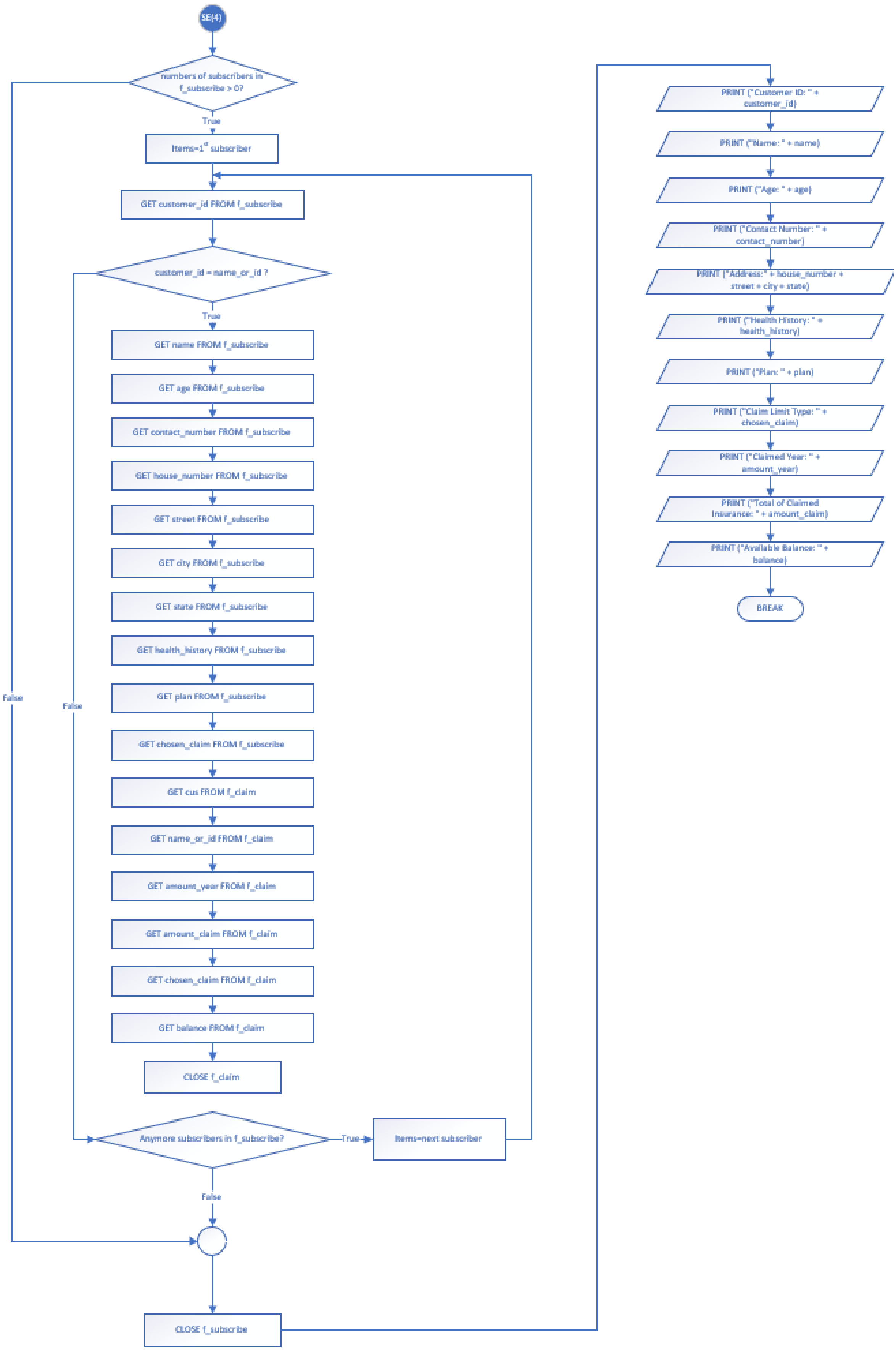

SE(4)
numbers of subscribers in f_subscribe > 0?
True
False
Items=1st subscriber
GET customer_id FROM f_subscribe
customer_id = name_or_id ?
True
False
GET name FROM f_subscribe
GET age FROM f_subscribe
GET contact_number FROM f_subscribe
GET house_number FROM f_subscribe
GET street FROM f_subscribe
GET city FROM f_subscribe
GET state FROM f_subscribe
GET health_history FROM f_subscribe
GET plan FROM f_subscribe
GET chosen_claim FROM f_subscribe
GET cus FROM f_claim
GET name_or_id FROM f_claim
GET amount_year FROM f_claim
GET amount_claim FROM f_claim
GET chosen_claim FROM f_claim
GET balance FROM f_claim
CLOSE f_claim
Anymore subscribers in f_subscribe?
True
Items=next subscriber
False
CLOSE f_subscribe
PRINT ("Customer ID: " + customer_id)
PRINT ("Name: " + name)
PRINT ("Age: " + age)
PRINT ("Contact Number: " + contact_number)
PRINT ("Address:" + house_number + street + city + state)
PRINT ("Health History: " + health_history)
PRINT ("Plan: " + plan)
PRINT ("Claim Limit Type: " + chosen_claim)
PRINT ("Claimed Year: " + amount_year)
PRINT ("Total of Claimed Insurance: " + amount_claim)
PRINT ("Available Balance: " + balance)
BREAK

Figure 2.2.28: *Searching Functionalities (Part 4)*

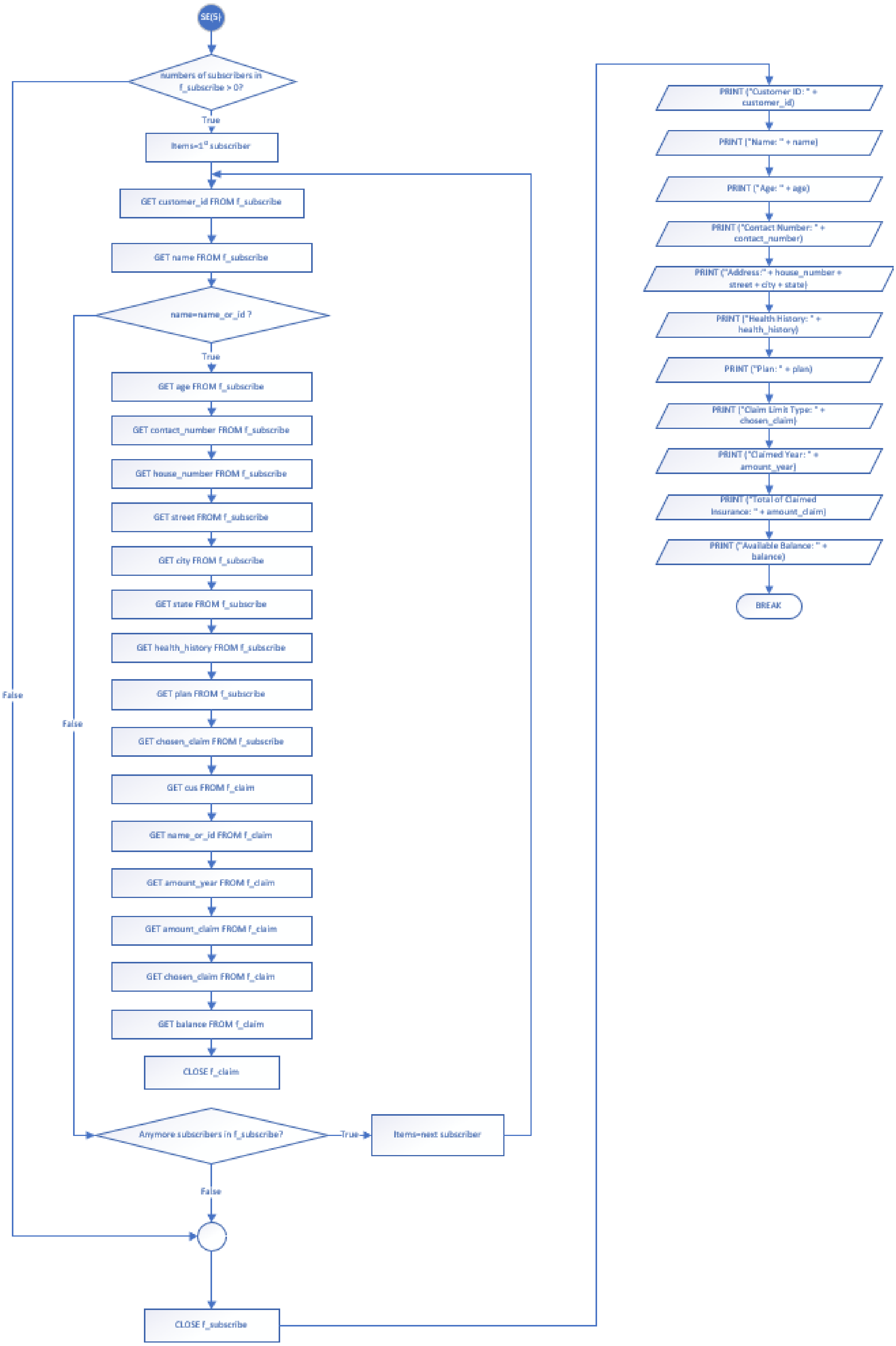

SE(S)
numbers of subscribers in f_subscribe > 0?
True
Items=1st subscriber
GET customer_id FROM f_subscribe
GET name FROM f_subscribe
name=name_or_id ?
True
GET age FROM f_subscribe
GET contact_number FROM f_subscribe
GET house_number FROM f_subscribe
GET street FROM f_subscribe
GET city FROM f_subscribe
GET state FROM f_subscribe
GET health_history FROM f_subscribe
GET plan FROM f_subscribe
GET chosen_claim FROM f_subscribe
GET cus FROM f_claim
GET name_or_id FROM f_claim
GET amount_year FROM f_claim
GET amount_claim FROM f_claim
GET chosen_claim FROM f_claim
GET balance FROM f_claim
CLOSE f_claim
False
False
Anymore subscribers in f_subscribe?
True
Items=next subscriber
False
CLOSE f_subscribe
PRINT ("Customer ID: " + customer_id)
PRINT ("Name: " + name)
PRINT ("Age: " + age)
PRINT ("Contact Number: " + contact_number)
PRINT ("Address:" + house_number + street + city + state)
PRINT ("Health History: " + health_history)
PRINT ("Plan: " + plan)
PRINT ("Claim Limit Type: " + chosen_claim)
PRINT ("Claimed Year: " + amount_year)
PRINT ("Total of Claimed Insurance: " + amount_claim)
PRINT ("Available Balance: " + balance)
BREAK

Figure 2.2.29: *Searching Functionalities (Part 5)*

3.0 Program Source Code with Comment

3.1 header.h

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>

void create_file();
void menu();
void subscription();
void claim();
void information();
void search();
void no_spaces(char store[], char filter[]);
```

3.2 main.c

```
#include "header.h"

//main function//
int main() {
        create_file();
        menu();
        return 0;
}
//Creating file at the beginning of the program//
void create_file() {
        FILE* f;
        //Read the file if the file is already existed//
        //Create a new file if the file is not existed//
        //For storing information of customers who subscribe to Plan120//
        if (f = fopen("Plan120.txt", "r")) {
                fclose(f);
        }
        else {
                f = fopen("Plan120.txt", "w");
                fprintf(f,
"ID\t\tNAME\t\tAGE\tCONTACT_NUMBER\tHOUSE_NUMBER\tSTREET\t\tCITY\t\tSTATE\t\tHEALTH_HI
STORY\t\tPLAN_TYPE\tCLAIM_LIMIT_TYPE\n");
                fclose(f);
        }
        //For storing information of customers who subscribe to Plan150//
        if (f = fopen("Plan150.txt", "r")) {
                fclose(f);
        }
        else {
                f = fopen("Plan150.txt", "w");
                fprintf(f,
"ID\t\tNAME\t\tAGE\tCONTACT_NUMBER\tHOUSE_NUMBER\tSTREET\t\tCITY\t\tSTATE\t\tHEALTH_HI
STORY\t\tPLAN_TYPE\tCLAIM_LIMIT_TYPE\n");
                fclose(f);
        }
        //For storing information of customers who subscribe to Plan200//
        if (f = fopen("Plan200.txt", "r")) {
                fclose(f);
        }
        else {
                f = fopen("Plan200.txt", "w");
                fprintf(f,
"ID\t\tNAME\t\tAGE\tCONTACT_NUMBER\tHOUSE_NUMBER\tSTREET\t\tCITY\t\tSTATE\t\tHEALTH_HI
STORY\t\tPLAN_TYPE\tCLAIM_LIMIT_TYPE\n");
                fclose(f);
        }
        //For storing details of claiming process for Plan120//
        if (f = fopen("Claim_plan120.txt", "r")) {
                fclose(f);
        }
        else {
                f = fopen("Claim_plan120.txt", "w");
                fprintf(f,
"ID\t\tCLAIMED_YEAR\t\tAMOUNT_CLAIMED\t\tBALANCE(YEAR)(RM)\tBALANCE(LIFETIME)(RM)\n");
                fclose(f);
        }
        //For storing details of claiming process for Plan150//
        if (f = fopen("Claim_plan150.txt", "r")) {
                fclose(f);
        }
```

```
       else {
              f = fopen("Claim_plan150.txt", "w");
              fprintf(f,
"ID\t\tCLAIMED_YEAR\t\tAMOUNT_CLAIMED\t\tBALANCE(YEAR)(RM)\tBALANCE(LIFETIME)(RM)\n");
              fclose(f);
       }
       //For storing details of claiming process for Plan200//
       if (f = fopen("Claim_plan200.txt", "r")) {
              fclose(f);
       }
       else {
              f = fopen("Claim_plan200.txt", "w");
              fprintf(f,
"ID\t\tCLAIMED_YEAR\t\tAMOUNT_CLAIMED\t\tBALANCE(YEAR)(RM)\tBALANCE(LIFETIME)(RM)\n");
              fclose(f);
       }
}

//Menu-driven//
void menu() {
       int choice, statement;
       statement = 1;
       //For looping the menu again and again//
       while (statement == 1) {
              printf("--------------------------------------------------------------
");
              printf("\nHEALTH INSURANCE MANAGEMENT SYSTEM");
              printf("\nPlease select one of the functions:");
              printf("\n1. Insurance Plan Subscription");
              printf("\n2. Claim Processing");
              printf("\n3. Accounts Information");
              printf("\n4. Searching Functionalities");
              printf("\n5. Exit");
              printf("\nYour choice:");
              scanf_s("%d", &choice);
              // Avoid useless input characters//
              while (getchar() != '\n');
              switch (choice) {
              case 1:
                     subscription();
                     break;
              case  2:
                     claim();
                     break;
              case 3:
                     information();
                     break;
              case 4:
                     search();
                     break;
              case 5:
                     //End the program//
                     return;
                     break;
                     //If the input not equal to 1-5, it will run the menu again//
              default:
                     printf("Invalid input\n");
                     statement = 1;
              }
       }
}
```

```
// Delete the spaces between the data while reading the file//
void no_spaces(char store[], char filter[]) {
       int a, b = 0;
       //Check whether there is a space within the fetched data//
       for (a = 0; store[a] != '\0'; a++) {
              //If there is not a space between two data, copy the whole data to an
array for the use of program//
              if (store[a] != ' ') {
                     filter[b] = store[a];
                     b++;
              }
       }
       filter[b] = '\0';
}
```

3.3 subscription.c

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>


//Create structure for storing the plan details//
typedef struct  {
	int name, monthly_premium, annual_limit, lifetime_limit, room_charges, 
icu_charges;
}plan_type;
//Three structures for three different plan types//
plan_type plan120 = { 120,120,12000000,60000000,120,250 };
plan_type plan150 = { 150,150,15000000,75000000,150,400 };
plan_type plan200 = { 200,200,20000000,100000000,200,700 };
plan_type* p1 = &plan120, * p2 = &plan150, * p3 = &plan200;

void plan_name(plan_type* p);

//This function is used for storing the details of plans//
void plan_name(plan_type *p) {
	printf("\nPlan Name: Plan%d\nMonthly Premium: %d\nAnnual Claim Limit : 
%d\nLifetime Claim Limit : %d", (*p).name,(*p).monthly_premium, (*p).annual_limit/100, 
(*p).lifetime_limit/100);
	printf("\nRoom Charges : %d / day\nIntensive Care Unit(ICU) Charge: %d/day", 
(*p).room_charges, (*p).icu_charges);
	printf("\nHospital Supplies and Services: As charged. Subject to approval by 
ZeeMediLife\nSurgical Fees: As charged. Subject to approval by ZeeMediLife.\nOther 
Fees: As charged. Subject to approval by ZeeMediLife.");
}

void subscription() {
	//Main_repeat is a while loop for the entire subscription program, if user want 
to enter another customer's data, it will repeat again//
	//Repeat is a while loop for the seperate part of the subscription program, if 
user enters a wrong input, it will require user to enter again//
	int main_repeat = 1, repeat, age, days, plan, claim, age_group, count;
	char cus[200];
	char id[5], customer_id[9];
	char cus_name[100], cus_contact_number[20], cus_house_number[20], 
cus_street[100], cus_city[50], cus_state[20], cus_health_history[100];
	char name[100], contact_number[20], house_number[20], street[100], city[50], 
state[20], health_history[100];
	//This structure is used to store the details of users' chosen plan and display 
them out after they entered their information//
	plan_type chosen_plan;
	char chosen_claim[30];

	//For main_repeat and repeat, '1' for entering the loop, '0' for ending the 
loop//
	while (main_repeat == 1) {
		repeat = 1;
		//1. Ask age//
		while (repeat == 1) {
			repeat = 0;
			//If users didn't want to enter another subscription, '999' for 
them to exit this function and enter to the menu//
			printf("\nEnter your age, (999) to exit:");
			scanf_s("%d", &age);
			// Avoid useless input characters//
			while (getchar() != '\n');
```

```
            if (age == 0) {
                //Ask day of the baby//
                printf("How many days old the baby is:");
                scanf_s("%d", &days);
                // Avoid useless input characters//
                while (getchar() != '\n');
                //Baby under 15 days is not eligible for applying
insurance//
                if (days >= 15) {
                    age_group = 1;
                }
                else {
                    printf("Your baby is not eligible for applying any
plans.\n");
                    repeat = 1;
                }
            }
            else if (age >= 1 && age <= 20) {
                age_group = 1;
            }
            else if (age >= 21 && age <= 40) {
                age_group = 2;
            }
            else if (age >= 41 && age <= 54) {
                age_group = 3;
            }
            else if (age == 999) {
                printf("\n");
                return;
            }
            else {
                printf("Your age is not eligible for applying any
plans.");
                repeat = 1;
            }
        }
        //Repeat change to 1 again to enter the next loop//
        repeat = 1;
        //2. Choose plan//
        if (age_group == 1) {
            printf("\nYou are eligible for applying the following plans:");
            //Plan120 descibrition//
            printf("\n1. Plan120");
            plan_name(p1);
            //Plan150 descibrition//
            printf("\n\n2. Plan150");
            plan_name(p2);
            //Plan200 descibrition//
            printf("\n\n3. Plan200");
            plan_name(p3);
            while (repeat == 1) {
                repeat = 0;
                printf("\n\nEnter your choice:");
                scanf_s("%d", &plan);
                // Avoid useless input characters//
                while (getchar() != '\n');
                if (plan == 1) {
                    chosen_plan = plan120;
                }
                else if (plan == 2) {
                    chosen_plan = plan150;
                }
                else if (plan == 3) {
                    chosen_plan = plan200;
                }
                else {
                    printf("\nPlease choose a listed plan.");
                    repeat = 1;
                }
            }
        }
        else if (age_group == 2) {
            printf("\nYou are eligible for applying the following plans:");
            //Plan150 descibrition//
            printf("\n1. Plan150");
            plan_name(p2);
            //Plan200 descibrition//
            printf("\n\n2. Plan200");
            plan_name(p3);
            while (repeat == 1) {
                repeat = 0;
                printf("\n\nEnter your choice:");
                scanf_s("%d", &plan);
                // Avoid useless input characters//
                while (getchar() != '\n');
                if (plan == 1) {
                    chosen_plan = plan150;
                }
                else if (plan == 2) {
                    chosen_plan = plan200;
                }
                else {
                    printf("\nPlease choose a listed plan.");
                    repeat = 1;
                }
            }
        }
        else if (age_group == 3) {
            printf("\nYou are eligible for applying the following plans:");
            //Plan20\n0 descibrition//
            printf("\n1. Plan200");
            plan_name(p3);
            while (repeat == 1) {
                repeat = 0;
                printf("\n\nEnter your choice:");
                scanf_s("%d", &plan);
                // Avoid useless input characters//
                while (getchar() != '\n');
                if (plan == 1) {
                    chosen_plan = plan200;
                }
                else {
                    printf("\nPlease choose a listed plan.");
                    repeat = 1;
                }
            }
        }
        //Repeat change to 1 again to enter the next loop//
        repeat = 1;
        //3. Choose claim limit type//
        printf("\n\nPlease choose a claim limit type:");
        printf("\n1. Annual Claim Limit");
        printf("\n2. Lifetime Claim Limit");
        while (repeat == 1) {
            repeat = 0;
            printf("\nEnter your choice:");
            scanf_s("%d", &claim);
            // Avoid useless input characters//
            while (getchar() != '\n');
            if (claim == 1) {
                strcpy_s(chosen_claim, 30, "Annual Claim Limit");
                chosen_plan.lifetime_limit = NULL;
            }
            else if (claim == 2) {
                strcpy_s(chosen_claim, 30, "Lifetime Claim Limit");
                chosen_plan.annual_limit = NULL;
            }
            else {
                printf("\nPlease choose a listed claim limit.");
                repeat = 1;
            }
        }
        //4. Customer information//
        // Delete the spaces between the data//
        printf("\n\nName: ");
        gets(cus_name);
        no_spaces(cus_name,name);
        printf("Contact Number: ");
        gets(cus_contact_number);
        no_spaces(cus_contact_number, contact_number);
        printf("House Number: ");
        gets(cus_house_number);
        no_spaces(cus_house_number, house_number);
        printf("Street: ");
        gets(cus_street);
        no_spaces(cus_street, street);
        printf("City: ");
        gets(cus_city);
        no_spaces(cus_city, city);
        printf("State: ");
        gets(cus_state);
        no_spaces(cus_state, state);
        printf("Health History: ");
        gets(cus_health_history);
        no_spaces(cus_health_history, health_history);
        count = 0;
        //Open customer subscription files//
        FILE* f_subscribe = NULL;
        //Open customer claim files//
        FILE* f_claim = NULL;
        switch (chosen_plan.name) {
        case 120:
            f_subscribe = fopen("Plan120.txt", "a+");
            f_claim = fopen("Claim_plan120.txt", "a");
            break;
        case 150:
            f_subscribe = fopen("Plan150.txt", "a+");
            f_claim = fopen("Claim_plan150.txt", "a");
            break;
        case 200:
            f_subscribe = fopen("Plan200.txt", "a+");
            f_claim = fopen("Claim_plan200.txt", "a");
            break;
        default:
            printf("File cannot be opened.");
        }
```

```
        //Skip the first line in the file before writing the customer 
information//
        fgets(cus, 200, f_subscribe);
        // Counting the number of customers//
        while (!feof(f_subscribe)) {
            if (fgetc(f_subscribe) - '0' == age_group) {
                count++;
            }
            fgets(cus, 200, f_subscribe);
        }
        // Generate a unique sequence number for the customers as there will be 
different level of number when excess 10,100,1000//
        if (count < 10) {
            sprintf(id, "00%d", count);
        }
        else if (count - 2 < 100) {
            sprintf(id, "0%d", count);
        }
        else {
            sprintf(id, "%d", count);
        }
        //Generate a customer id//
        //Customer ID format: ABBBXXX A:AGE_GROUP(1,2,3) 
BBB:CHOSEN_PLAN(120,150,200) XXX:SEQUENCE NUMBER(001,002,...)//
        sprintf(customer_id, "%d%d%s", age_group, chosen_plan.name, id);
        //Store customer information to subscribe file//
        fprintf(f_subscribe, "%-16s%-17s%-7d%-16s%-16s%-16s%-16s%-16s%-24s%-
16d%s\n", customer_id, name, age, contact_number, house_number, street, city, state, 
health_history, chosen_plan.name, chosen_claim);
        //Store plan information to claim file//
        fprintf(f_claim, "%-16sN/A\t\t\t\t0.00\t\t%11d.%-12.2d%11d.%-
12.2d\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\n", customer_id, 
chosen_plan.annual_limit / 100, chosen_plan.annual_limit % 100, 
chosen_plan.lifetime_limit / 100, chosen_plan.lifetime_limit % 100);
        fclose(f_subscribe);
        fclose(f_claim);
        //5. Show customer details and plan information to customer//
        printf("\nCustomer ID: %s\nName: %s\nAge: %d\nPhone Number: 
%s\nAddress:%s, %s, %s, %s\nHealth History: %s\nChosen Plan: Plan%d\nClaim Limit Type: 
%s\n", customer_id, cus_name, age, cus_contact_number, cus_house_number, cus_street, 
cus_city, cus_state, cus_health_history, chosen_plan.name, chosen_claim);
        //Repeat the subscription function again//
        main_repeat = 1;
    }
}
```

3.4 claim.c

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <math.h>

void claim() {
        //Create structure for storing the plan details//
        typedef struct {
                int name, monthly_premium, annual_limit, lifetime_limit, room_charges,
icu_charges;
        }plan_type;
        //Three structures for three different plan types//
        plan_type plan120 = { 120,120,12000000,60000000,120,250 };
        plan_type plan150 = { 150,150,15000000,75000000,150,400 };
        plan_type plan200 = { 200,200,20000000,100000000,200,700 };
        plan_type* p1 = &plan120, * p2 = &plan150, * p3 = &plan200;

        int choice, main_repeat = 1, repeat, a, count, amount_claim, claim_in_rm,
claim_in_sen, amount_year, yearleft_rm, yearleft_sen, amount_life, lifeleft_rm,
lifeleft_sen, sum;
        int year = 2020;
        char customer_id[10], cus[200], input_id[50], chosen_claim[21];
        //This structure is used to store the details of users' chosen plan//
        plan_type chosen_plan;
        int day_ward = 0, day_icu = 0, services = 0, surgery = 0, others = 0;

        while (main_repeat == 1) {
                repeat = 1;
                while (repeat == 1) {
                        printf("Enter customer ID, 999 to exit:");
                        gets(customer_id);
                        //999 to exit claim function and back to menu function//
                        if (strcmp(customer_id, "999") == 0) {
                                return;
                        }
                        //Declare file pointer//
                        FILE* f = NULL;
                        //This for loop is used to check whether customer ID is exist in
each Claim_plan text file//
                        for (a = 0; a < 3; a++) {
                                switch (a) {
                                case 0:
                                        f = fopen("Claim_plan120.txt", "r");
                                        chosen_plan = plan120;
                                        break;
                                case 1:
                                        f = fopen("Claim_plan150.txt", "r");
                                        chosen_plan = plan150;
                                        break;
                                case 2:
                                        f = fopen("Claim_plan200.txt", "r");
                                        chosen_plan = plan200;
                                        break;
                                default:
                                        printf("File cannot be opened.");
                                }
                                //Skip the first line in the file before input_iding the
customer information//
                                fgets(cus, 200, f);
                                //As the first line is skipped, hence add 1 into count//
                count = 1;
                //This for loop is used to fetch the available balance of
the customer//
                for (fscanf(f, "%s", input_id); !feof(f); fscanf(f, "%s",
input_id)) {
                    count++;
                    //To check whether the user's input ID is same with
the customer ID in each file//
                    if (strcmp(input_id, customer_id) == 0) {
                        fscanf(f, "%*s %d%*c%d %d%*c%d %d%*c%d",
&claim_in_rm, &claim_in_sen, &yearleft_rm, &yearleft_sen, &lifeleft_rm,
&lifeleft_sen);
                        //To calculate the available balance of
customer//
                        amount_claim = claim_in_rm * 100 +
claim_in_sen;
                        amount_year = yearleft_rm * 100 +
yearleft_sen;
                        amount_life = lifeleft_rm * 100 +
lifeleft_sen;
                        fclose(f);
                        switch (chosen_plan.name) {
                        case 120:
                            f = fopen("Plan120.txt", "r");
                            break;
                        case 150:
                            f = fopen("Plan150.txt", "r");
                            break;
                        case 200:
                            f = fopen("Plan200.txt", "r");
                        }
                        //Skip the first line in the file before
reading the customer information//
                        fgets(cus, 200, f);
                        for (fscanf(f, "%s", input_id); !feof(f);
fscanf(f, "%s", input_id)) {
                            //To check whether the user's input ID
is same with the customer ID in each file//
                            if (strcmp(input_id, customer_id) ==
0) {
                                fscanf(f, "%*s %*s %*s %*s %*s
%*s %*s %*s %*s %s %*s %*s", chosen_claim);
                                fclose(f);
                                repeat = 0;
                                break;
                            }
                            else {
                                //Go to the next line to check
whether the user's input ID is same with the customer ID//
                                fgets(cus, 200, f);
                            }
                        }
                        break;
                    }
                    else {
                        //Go to the next line to check whether the
user's input ID is same with the customer ID//
                        fgets(cus, 200, f);
                    }
                }
                //After the customer ID has been found, the for loop will
be break//
```

```
                if (repeat == 0) {
                    break;
                }
                //After checking all the file, if the input ID still
haven't found, the following message will be printed//
                if (a == 2) {
                    printf("Please enter a valid input.\n");
                }
            }
        }
        repeat = 1;
        while (repeat == 1) {
            repeat = 0;
            printf("How many days do you stay in normal wards: ");
            scanf_s("%d", &day_ward);
            // Avoid useless input characters//
            while (getchar() != '\n');
            printf("How many days do you stay in ICU: ");
            scanf_s("%d", &day_icu);
            // Avoid useless input characters//
            while (getchar() != '\n');
            printf("Enter the amount of hospital supplies and services in RM:
");

            scanf_s("%d", &services);
            // Avoid useless input characters//
            while (getchar() != '\n');
            printf("Enter the amount of surgical fees in RM: ");
            scanf_s("%d", &surgery);
            // Avoid useless input characters//
            while (getchar() != '\n');
            printf("Enter the amount of other fees in RM: ");
            scanf_s("%d", &others);
            // Avoid useless input characters//
            while (getchar() != '\n');

            //Calculate the sum of the claimed amount//
            sum = (day_ward * chosen_plan.room_charges + day_icu *
chosen_plan.icu_charges + services + surgery + others) * 100;

            //To check the amount of the available balance in customer's
chosen claim limit type//
            if (strcmp(chosen_claim, "Lifetime") == 0) {
                if (sum > amount_life) {
                    //If the sum of claimed amount excess the available
balance, then customers can only claim the left balance//
                    printf("Your available balance for current plan is
not enough.");

                    sum = amount_life;
                }
            }
            else {
                if (sum > amount_year) {
                    //If the sum of claimed amount excess the available
balance, then customers can only claim the left balance//
                    printf("Your available balance for current plan is
not enough.");

                    sum = amount_year;
                }
            }
            //Declare file pointer//
            FILE* f = NULL;
            switch (chosen_plan.name) {
```

```
            case 120:
                f = fopen("Claim_plan120.txt", "r+");
                break;
            case 150:
                f = fopen("Claim_plan150.txt", "r+");
                break;
            case 200:
                f = fopen("Claim_plan200.txt", "r+");
            }

            //To find of the line of data that needed to be modified//
            for (cus[0] = fgetc(f); cus[0] != EOF; cus[0] = fgetc(f)) {
                if (cus[0] == '\n') {
                    count--;
                    if (count == 1) {
                        break;
                    }
                }
            }
            fseek(f, 0, SEEK_CUR);

            //To check the customer's chosen claim type limit//
            if (strcmp(chosen_claim, "Lifetime") == 0) {
                //Update the amount_claimed and the balance of their
plan//
                fprintf(f, "%-16s%-16s%11d.%-12.2d\t  0.00\t\t%11d.%-
12.2d", customer_id, year, (amount_claim + sum) / 100, (amount_claim + sum) % 100,
(amount_life - sum) / 100, (amount_life - sum) % 100);
            }
            else {
                //Update the amount_claimed and the balance of their
plan//
                fprintf(f, "%-16s%-24d%11d.%-12.2d%11d.%-12.2d\t
0.00\t\t", customer_id, year, (amount_claim + sum) / 100, (amount_claim + sum) % 100,
(amount_year - sum) / 100, (amount_year - sum) % 100);
            }
            fclose(f);
            main_repeat = 1;
        }
    }
}
```

3.5 information.c

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>

void information() {
	int a = 0, b = 0, c, d;
	char num_lifesubscriber[1000][10], num_yearsubscriber[1000][10];
	int total_amount_claimed = 0, num_exsubscriber = 0;
	//Declare file pointer//
	FILE* f = NULL;
	char cus[200], customer_id[10], chosen_claim[21];
	for (c = 0; c < 3; c++) {
		switch (c) {
		case 0:
			f = fopen("Plan120.txt", "r");
			break;
		case 1:
			f = fopen("Plan150.txt", "r");
			break;
		case 2:
			f = fopen("Plan200.txt", "r");
		}
		//Skip the first line in the file before reading the customer
information//
		fgets(cus, 200, f);

		//Only read the needed information//
		for (fscanf(f, "%s %*s %*s %*s %*s %*s %*s %*s %*s %*s %s %*s %*s",
customer_id, chosen_claim); !feof(f); fscanf(f, "%s %*s %*s %*s %*s %*s %*s %*s %*s
%*s %s %*s %*s", customer_id, chosen_claim)) {
			if (strcmp(chosen_claim, "Lifetime") == 0) {
				strcpy(num_lifesubscriber[a], customer_id);
				a++;
			}
			else {
				strcpy(num_yearsubscriber[b], customer_id);
				b++;
			}
		}
		fclose(f);
	}
	int amount_claim, claim_in_rm, claim_in_sen, amount_year, yearleft_rm,
yearleft_sen;
	for (d = 0; d < 2; d++) {
		switch (d) {
		case 0:
			for (c = 0; c < 3; c++) {
				switch (c) {
				case 0:
					f = fopen("Claim_plan120.txt", "r");
					break;
				case 1:
					f = fopen("Claim_plan150.txt", "r");
					break;
				case 2:
					f = fopen("Claim_plan200.txt", "r");
				}
				//Skip the first line in the file before reading the
customer information//
				fgets(cus, 200, f);
```

```
                for (fscanf(f, "%s %*s %d%*c%d", customer_id,
&claim_in_rm, &claim_in_sen); !feof(f); fscanf(f, "%s %*s %d%*c%d", customer_id,
&claim_in_rm, &claim_in_sen)) {
                    for (a = 0; a < 1000; a++) {
                        if (strcmp(num_lifesubscriber[a],
customer_id) == 0) {
                            amount_claim = claim_in_rm * 100 +
claim_in_sen;
                            total_amount_claimed =
total_amount_claimed + amount_claim;
                        }
                    }
                    //Next new line//
                    fgets(cus, 200, f);
                }
            }
            break;
        case 1:
            for (c = 0; c < 3; c++) {
                switch (c) {
                case 0:
                    f = fopen("Claim_plan120.txt", "r");
                    break;
                case 1:
                    f = fopen("Claim_plan150.txt", "r");
                    break;
                case 2:
                    f = fopen("Claim_plan200.txt", "r");
                }
                //Skip the first line in the file before reading the
customer information//
                fgets(cus, 200, f);
                for (fscanf(f, "%s %*s %*d%*c%*d %d%*c%d %*s",
customer_id, &yearleft_rm, &yearleft_sen); !feof(f); fscanf(f, "%s %*s %*d%*c%*d
%d%*c%d %*s", customer_id, &yearleft_rm, &yearleft_sen)) {
                    for (a = 0; a < 1000; a++) {
                        if (strcmp(num_yearsubscriber[a],
customer_id) == 0) {
                            amount_year = yearleft_rm * 100 +
yearleft_sen;
                            if (amount_year == 0) {
                                num_exsubscriber =
num_exsubscriber + 1;
                            }
                        }
                    }
                }
            }
        }
    }
    fclose(f);
    printf("\nTotal amount claimed by Lifetime Claim Limit subscribers(RM):
%d.%.2d\nTotal number of Annual Claim Limit subscribers who have exhausted all their
eligible amount: %d\n\n", total_amount_claimed / 100, total_amount_claimed % 100,
num_exsubscriber);
}
```

3.6 search.c

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>

void fetch_data(char string1[], char string2[], FILE* f_claim, char* amount_year, char* amount_claim, char chosen_claim[], char* balance);

//This function is used for fetching the data from claim_plan file//
void fetch_data(char string1[], char string2[], FILE* f_claim, char* amount_year, char* amount_claim, char chosen_claim[], char* balance) {
    if (strcmp(string1, string2) == 0) {
        fscanf(f_claim, "%s %s", amount_year, amount_claim);
        if (strcmp(chosen_claim, "Annual") == 0) {
            fscanf(f_claim, "%s", balance);
            fgets(string1, 200, f_claim);
        }
        else {
            fscanf(f_claim, "%*s %s", balance);
        }
    }
    else {
        fgets(string1, 200, f_claim);
    }
}

void search() {
    //Main_repeat is a while loop for the entire search program, if user want to search another data, it will repeat again//
    //Repeat is a while loop for the seperate part of the search program, if user enters a wrong input, it will require user to enter again//
    int choice, a, repeat, main_repeat = 1, plan, age;
    char name_or_id[100], cus_name[100], customer_id[10];
    int cus_plan, cus_age;
    char cus_chosen_plan[21];
    char name[100], contact_number[100], house_number[100], street[100], city[100], state[100], health_history[100], chosen_claim[21];
    char amount_year[5], amount_claim[100], balance[100];
    char cus[200];
    FILE* f_subscribe = NULL;
    FILE* f_claim = NULL;

    //For main_repeat and repeat, '1' for entering the loop, '0' for ending the loop//
    while (main_repeat == 1) {
        main_repeat = 0;
        printf("\nDo you want use searching functionalities by entering: ");
        printf("\n1. Customer ID or Name");
        printf("\n2. Plan, Claim Limit Type and Age");
        printf("\n999. Exit");
        printf("\nEnter your choice: ");
        //Choice variable only accepts the number(1,2), others will be assume as invalid
input//
        scanf_s("%d", &choice);
        // Avoid useless input characters//
        while (getchar() != '\n');
        repeat = 1;
        switch (choice) {
        //For searching by ID or customer name//
        case 1:
            while (repeat == 1) {
                repeat = 0;
                printf("\nDo you want use searching functionalities by entering:
");
                printf("\n1. Customer ID");
                printf("\n2. Customer Name");
                printf("\nEnter your choice: ");
                scanf_s("%d", &choice);
                // Avoid useless input characters//
                while (getchar() != '\n');
                switch (choice) {
                //For searching by customer ID//
                case 1:
                    repeat = 1;
                    while (repeat == 1) {
                        printf("Enter Customer ID: ");
                        gets(name_or_id);
                        //This for loop is used to open all the text files//
                        for (a = 0; a < 3; a++) {
                            switch (a) {
                            case 0:
                                f_subscribe = fopen("Plan120.txt",
"r");
                                f_claim                    =
fopen("Claim_plan120.txt", "r");
                                break;
                            case 1:
                                f_subscribe = fopen("Plan150.txt",
"r");
                                f_claim                    =
fopen("Claim_plan150.txt", "r");
                                break;
                            case 2:
                                f_subscribe = fopen("Plan200.txt",
"r");
                                f_claim                    =
fopen("Claim_plan200.txt", "r");
                            }
                            //Skip the first line in the file before
reading the customer information//
                            fgets(cus, 200, f_subscribe);
                            //Skip the first line in the file before
reading the customer information//
                            fgets(cus, 200, f_claim);
                            //This for loop is used to check customer
ID in all the text files//
                            for (fscanf(f_subscribe, "%s",
customer_id); !feof(f_subscribe); fscanf(f_subscribe, "%s", customer_id)) {
                                //This is for matching the input
and the data in each file//
                                if (strcmp(customer_id,
name_or_id) == 0) {
                                    //If matched, the program
will fetch the data from the specific file and store them in different variables//
                                    fscanf(f_subscribe,
"%s %d %s %s %s %s %s %s %d %s", name, &age, contact_number, house_number, street,
city, state, health_history, &plan, chosen_claim);
                                    for (fscanf(f_claim, "%s",
cus); !feof(f_claim); fscanf(f_claim, "%s", cus)) {
                                        fetch_data(cus,
name_or_id, f_claim, amount_year, amount_claim, chosen_claim, balance);
                                    }
                                    fclose(f_subscribe);
                                    //What about the f_claim
file????
                                    //Display the customer's
information//
                                    printf("\nCustomer
ID: %s\nName: %s\nAge: %d\nContact
Number: %s\nAddress:\t%s, %s,\n\t\t %s,\n\t\t%s\nHealth History: %s\nPlan: %d\nClaim
Limit Type: %s\nClaimed Year: %s\nTotal of Claimed Insurance: %s\nAvailable
Balance: %s\n", customer_id, name, age, contact_number, house_number, street, city, state,
health_history, plan, chosen_claim, amount_year, amount_claim, balance);
                                    repeat = 0;
                                    break;
                                }
                                fgets(cus, 200, f_subscribe);
                            }
                            if (repeat == 0) {
                                break;
                            }
                        }
                        if (repeat == 1) {
                            printf("Please enter a valid customer
ID.\n");
                        }
                    }
                    break;
                case 2:
                    repeat = 1;
                    while (repeat == 1) {
                        printf("Customer Name: ");
                        gets(cus_name);
                        no_spaces(cus_name, name_or_id);
                        //This for loop is used to open all the text files//
                        for (a = 0; a < 3; a++) {
                            switch (a) {
                            case 0:
                                f_subscribe = fopen("Plan120.txt",
"r");
                                f_claim =
fopen("Claim_plan120.txt", "r");
                                break;
                            case 1:
                                f_subscribe = fopen("Plan150.txt",
"r");
                                f_claim =
fopen("Claim_plan150.txt", "r");
                                break;
                            case 2:
                                f_subscribe = fopen("Plan200.txt",
"r");
                                f_claim =
fopen("Claim_plan200.txt", "r");
                            }
                            //Skip the first line in the file before
reading the customer information//
                            fgets(cus, 200, f_subscribe);
                            //Skip the first line in the file before
reading the customer information//
                            fgets(cus, 200, f_claim);
                            //This for loop is used to check customer
Name in all the text files//
                            for (fscanf(f_subscribe, "%s %s",
customer_id, name); !feof(f_subscribe); fscanf(f_subscribe, "%s %s", customer_id, name)) {
                                //This is for matching the input
and the data in each file//
                                if (strcmp(name, name_or_id) ==
0) {
                                    //If matched, the program
will fetch the data from the specific file and store them in different variables//
                                    fscanf(f_subscribe,
"%s %d %s %s %s %s %s %s %d %s", name, &age, contact_number, house_number, street,
city, state, health_history, &plan, chosen_claim);
                                    for (fscanf(f_claim, "%s",
cus); !feof(f_claim); fscanf(f_claim, "%s", cus)) {
                                        fetch_data(cus,
customer_id, f_claim, amount_year, amount_claim, chosen_claim, balance);
                                    }
                                    repeat = 0;
                                    //Display the customer's
information//
                                    printf("\nCustomer
ID: %s\nName: %s\nAge: %d\nContact
Number: %s\nAddress:\t%s, %s,\n\t\t %s,\n\t\t%s\nHealth History: %s\nPlan: %d\nClaim
Limit Type: %s\nClaimed Year: %s\nTotal of Claimed Insurance: %s\nAvailable
Balance: %s\n", customer_id, name, age, contact_number, house_number, street, city, state,
health_history, plan, chosen_claim, amount_year, amount_claim, balance);
                                }
                                fgets(cus, 200, f_subscribe);
                                if (repeat == 1) {
                                    printf("File cannot be
opened.");
                                }
                            }
                        }
                    }
                    break;
                default:
                    printf("Please enter a valid input.\n");
                    repeat = 1;
                }
            }
            //Break for ending a case in the switch//
            break;
        //For searching by plan, claim limit type and age//
        case 2:
            //Repeat change to 1 again to enter the next loop//
            repeat = 1;
            //1. Select a plan//
            printf("\nPlease select a plan:");
            printf("\n1. Plan120");
            printf("\n2. Plan150");
            printf("\n3. Plan200");
            while (repeat == 1) {
                repeat = 0;
                printf("\nEnter Choice:");
                scanf_s("%d", &choice);
                // Avoid useless input characters//
                while (getchar() != '\n');
                switch (choice) {
                case 1:
                    cus_plan = 120;
                    break;
                case 2:
                    cus_plan = 150;
                    break;
                case 3:
                    cus_plan = 200;
                    break;
```

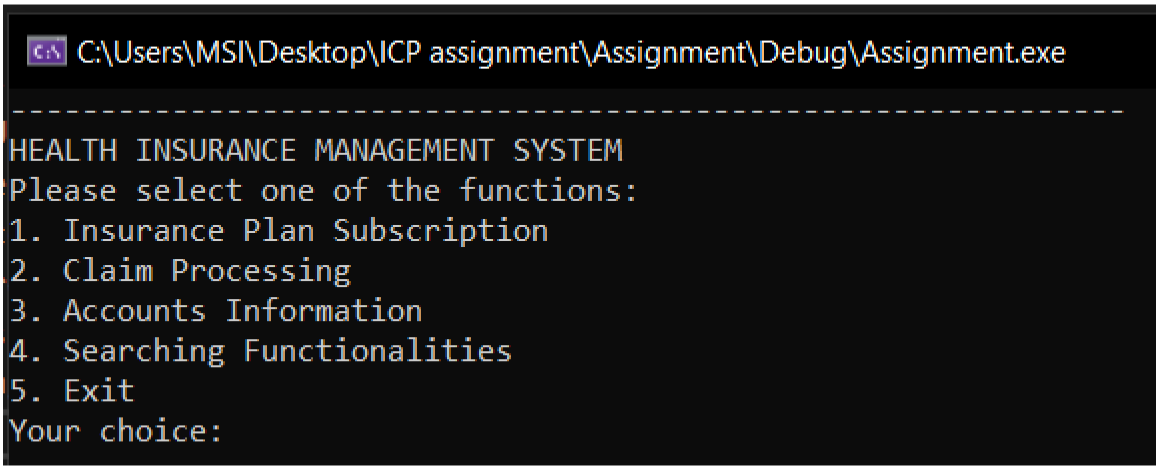
```
    default:
        printf("Please select a valid plan.");
        repeat = 1;
    }
}
//Repeat change to 1 again to enter the next loop//
repeat = 1;
//2. Select a claim limit type//
printf("\nPlease choose a claim limit type:");
printf("\n1.Annual Claim Limit");
printf("\n2.Lifetime Claim Limit");
while (repeat == 1) {
    repeat = 0;
    printf("\nEnter your choice: ");
    scanf_s("%d", &choice);
    // Avoid useless input characters//
    while (getchar() != '\n');
    switch (choice) {
    case 1:
        strcpy(cus_chosen_plan, "Annual");
        break;
    case 2:
        strcpy(cus_chosen_plan, "Lifetime");
        break;
    default:
        printf("Please choose a listed claim limit type. ");
        repeat = 1;
    }
}
//3. Select age//
printf("Enter age: ");
scanf_s("%d", &cus_age);
// Avoid useless input characters//
while (getchar() != '\n');
repeat = 1;
//This for loop is used to open all the text files//
for (a = 0; a < 3; a++) {
    switch (a) {
    case 0:
        f_subscribe = fopen("Plan120.txt", "r");
        f_claim = fopen("Claim_plan120.txt", "r");
        break;
    case 1:
        f_subscribe = fopen("Plan150.txt", "r");
        f_claim = fopen("Claim_plan150.txt", "r");
        break;
    case 2:
        f_subscribe = fopen("Plan200.txt", "r");
        f_claim = fopen("Claim_plan200.txt", "r");
    }
```

```
                //Skip the first line in the file before reading the customer
information//
                fgets(cus, 200, f_subscribe);
                //Skip the first line in the file before reading the customer
information//
                fgets(cus, 200, f_claim);
                //This for loop is used to check plan type, claim type limit and
age in all the text files//
                for (fscanf(f_subscribe,
"%s %s %d %s %s %s %s %s %s %d %s", customer_id, name, &age, contact_number,
house_number, street, city, state, health_history, &plan, chosen_claim); !feof(f_subscribe);
fscanf(f_subscribe, "%s %s %d %s %s %s %s %s %s %d %s", customer_id, name, &age,
contact_number, house_number, street, city, state, health_history, &plan, chosen_claim)) {
                    if (cus_plan == plan && strcmp(cus_chosen_plan,
chosen_claim) == 0 && cus_age == age) {
                        for (fscanf(f_claim, "%s", cus); !feof(f_claim);
fscanf(f_claim, "%s", cus)) {
                            //If matched, the program will fetch the
data from the specific file and store them in different variables//
                            fetch_data(cus, customer_id, f_claim,
amount_year, amount_claim, chosen_claim, balance);
                            //Display the customer's information//
                            printf("\nCustomer
ID: %s\nName: %s\nAge: %d\nContact Number: %s\nAddress:%s, %s, %s, %s\nHealth
History: %s\nPlan: %d\nClaim Limit Type: %s\nClaimed Year: %s\nTotal of Claimed
Insurance: %s\nAvailable Balance: %s\n", customer_id, name, age, contact_number,
house_number, street, city, state, health_history, plan, chosen_claim, amount_year,
amount_claim, balance);
                            repeat = 0;
                        }
                    }
                    //Skip the last two words in the file and read the next
line//
                    fgets(cus, 200, f_subscribe);
                }
            }
            if (repeat == 1) {
                printf("Can't find the specific age.\n");
            }
            break;
        //999 for exit search function and back to menu
        case 999:
            printf("\n");
            return;
            break;
        default:
            printf("Please select the available choice.\n");
            main_repeat = 1;
        }
    }
```

```
}
```

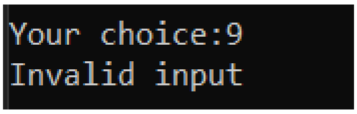
4.0 Input and Output of Program

4.1 Main Menu

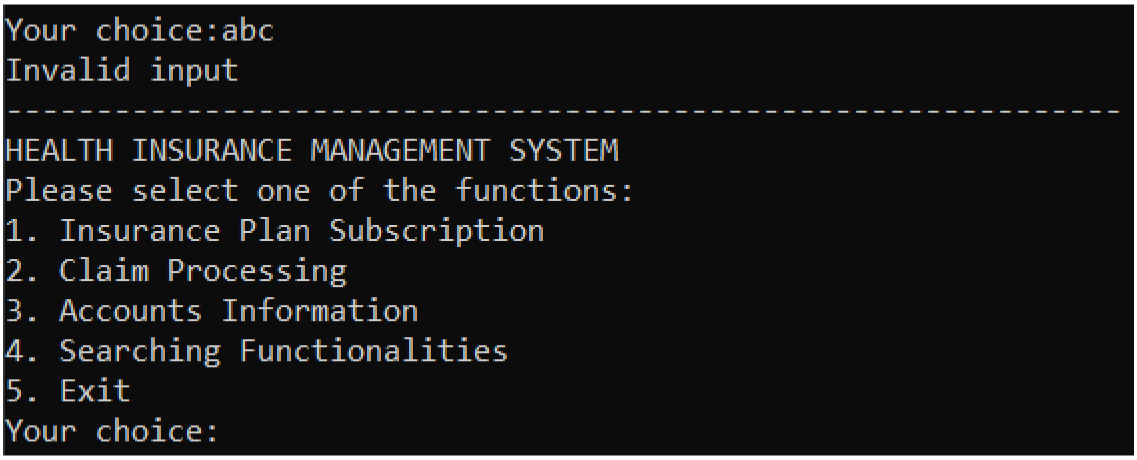


```
C:\Users\MSI\Desktop\ICP assignment\Assignment\Debug\Assignment.exe
-----------------------------------------------------------------
HEALTH INSURANCE MANAGEMENT SYSTEM
Please select one of the functions:
1. Insurance Plan Subscription
2. Claim Processing
3. Accounts Information
4. Searching Functionalities
5. Exit
Your choice:
```

After compiled and start the program without debugging, the very first output is the main menu of the system. It provides five functionalities including plan subscription, insurance claiming, account information and searching functionalities. Besides integers from 1 to 5, users are not allowed to enter any other integers or characters.

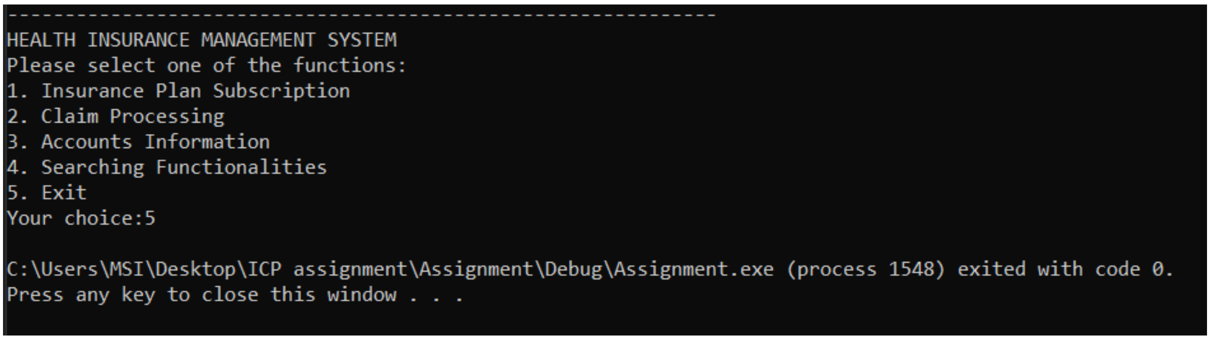


```
Your choice:9
Invalid input
```



```
Your choice:abc
Invalid input
-----------------------------------------------------------------
HEALTH INSURANCE MANAGEMENT SYSTEM
Please select one of the functions:
1. Insurance Plan Subscription
2. Claim Processing
3. Accounts Information
4. Searching Functionalities
5. Exit
Your choice:
```

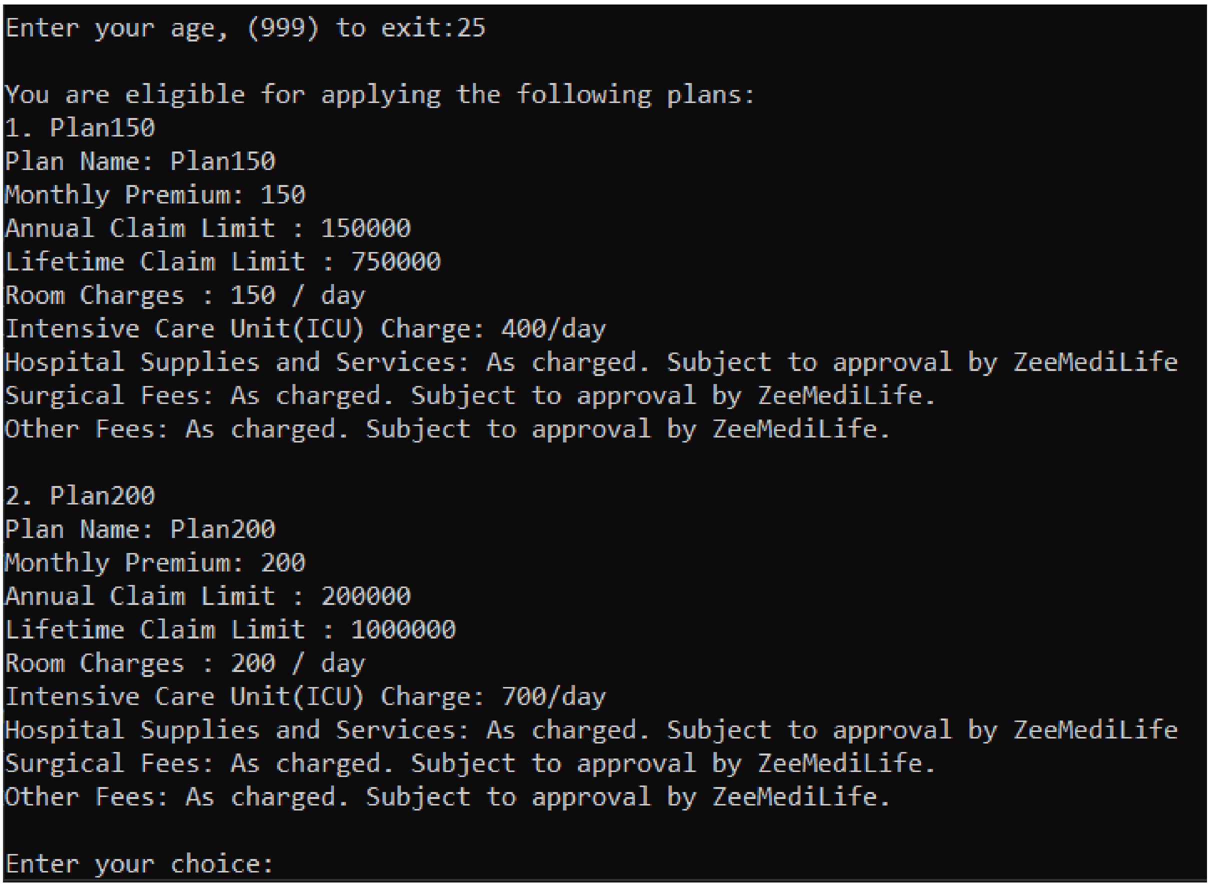
Otherwise, the “invalid input” will be shown and the program will run again the main menu.



```
--------------------------------------------------------------
HEALTH INSURANCE MANAGEMENT SYSTEM
Please select one of the functions:
1. Insurance Plan Subscription
2. Claim Processing
3. Accounts Information
4. Searching Functionalities
5. Exit
Your choice:5

C:\Users\MSI\Desktop\ICP assignment\Assignment\Debug\Assignment.exe (process 1548) exited with code 0.
Press any key to close this window . . .
```

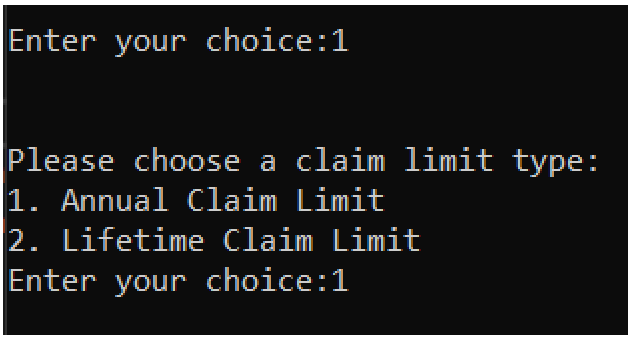
The input ‘5’ will end up the program automatically.

4.2 Insurance Plan Subscription



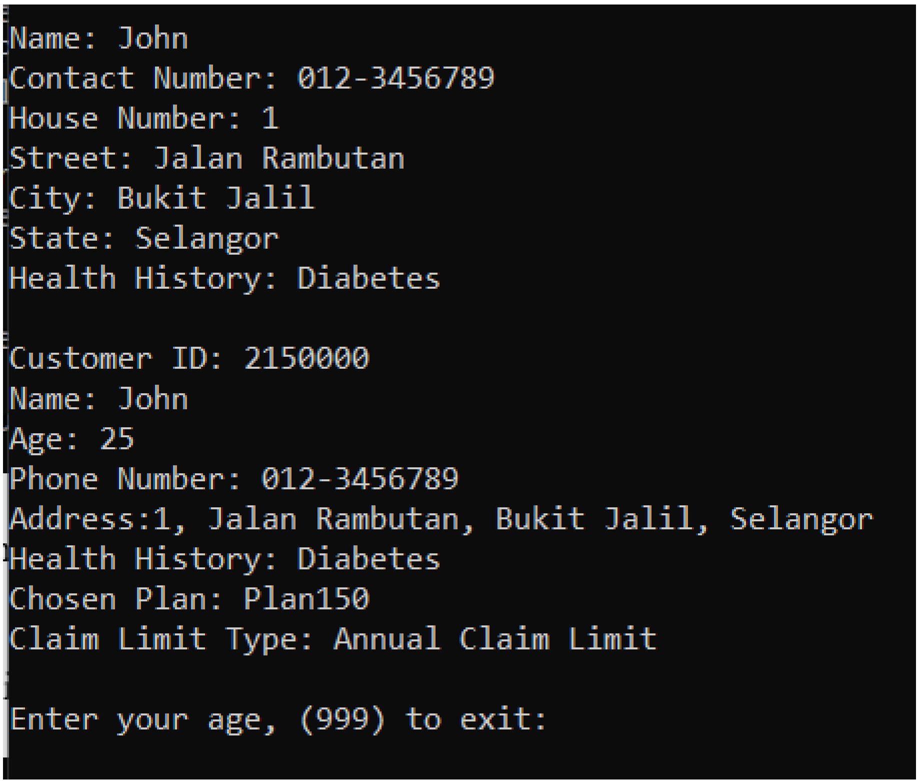
```
Enter your age, (999) to exit:25

You are eligible for applying the following plans:
1. Plan150
Plan Name: Plan150
Monthly Premium: 150
Annual Claim Limit : 150000
Lifetime Claim Limit : 750000
Room Charges : 150 / day
Intensive Care Unit(ICU) Charge: 400/day
Hospital Supplies and Services: As charged. Subject to approval by ZeeMediLife
Surgical Fees: As charged. Subject to approval by ZeeMediLife.
Other Fees: As charged. Subject to approval by ZeeMediLife.

2. Plan200
Plan Name: Plan200
Monthly Premium: 200
Annual Claim Limit : 200000
Lifetime Claim Limit : 1000000
Room Charges : 200 / day
Intensive Care Unit(ICU) Charge: 700/day
Hospital Supplies and Services: As charged. Subject to approval by ZeeMediLife
Surgical Fees: As charged. Subject to approval by ZeeMediLife.
Other Fees: As charged. Subject to approval by ZeeMediLife.

Enter your choice:
```

The first feature of the designed system is to accept new customer's subscription. By entering '1' in the main menu, the system will ask the customer's age to show the available plan for his or her age group. After entered the age, the plan details will be shown on the screen to let the users to choose one of the available plans.



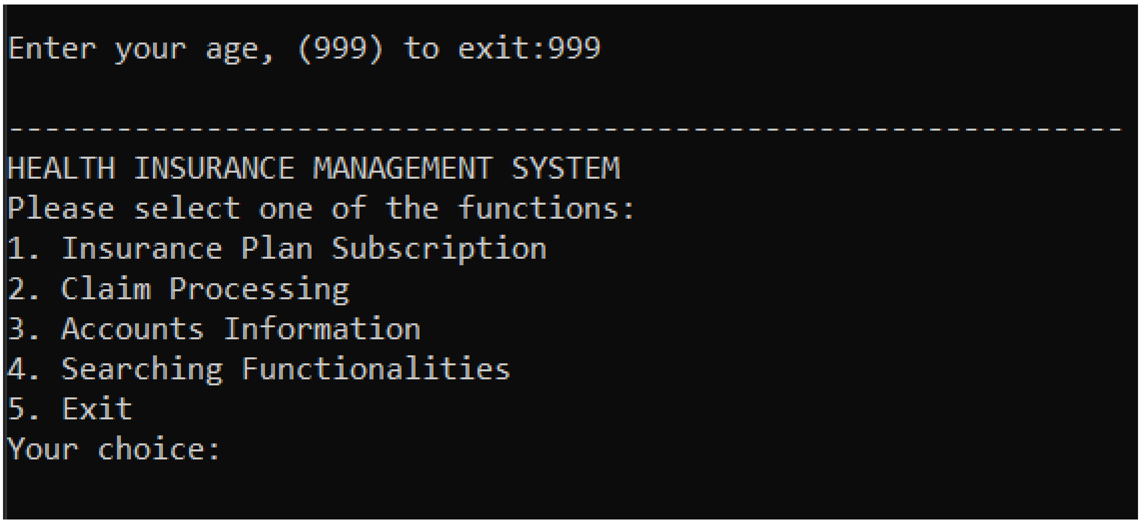
```
Enter your choice:1


Please choose a claim limit type:
1. Annual Claim Limit
2. Lifetime Claim Limit
Enter your choice:1
```

After choosing the plan, users are required to choose one of the claim limits.

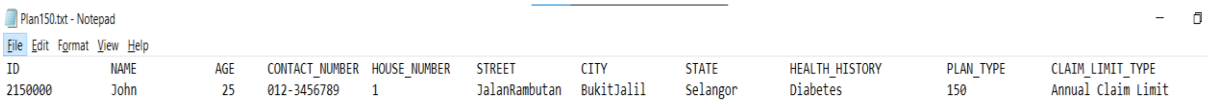


```
Name: John
Contact Number: 012-3456789
House Number: 1
Street: Jalan Rambutan
City: Bukit Jalil
State: Selangor
Health History: Diabetes

Customer ID: 2150000
Name: John
Age: 25
Phone Number: 012-3456789
Address:1, Jalan Rambutan, Bukit Jalil, Selangor
Health History: Diabetes
Chosen Plan: Plan150
Claim Limit Type: Annual Claim Limit

Enter your age, (999) to exit:
```

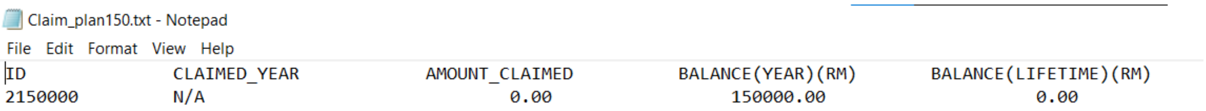
After that, users are required to enter the customer's details include name, contact number, address and health history. Customer ID will then be generated, and the system will display the details of all the related information for this subscription. Integer '999' will let the system return to the main menu.



```
Enter your age, (999) to exit:999

-------------------------------------------------------------
HEALTH INSURANCE MANAGEMENT SYSTEM
Please select one of the functions:
1. Insurance Plan Subscription
2. Claim Processing
3. Accounts Information
4. Searching Functionalities
5. Exit
Your choice:
```



```
ID       NAME     AGE  CONTACT_NUMBER  HOUSE_NUMBER  STREET         CITY        STATE     HEALTH_HISTORY  PLAN_TYPE  CLAIM_LIMIT_TYPE
2150000  John     25   012-3456789     1             JalanRambutan  BukitJalil  Selangor  Diabetes        150        Annual Claim Limit
```

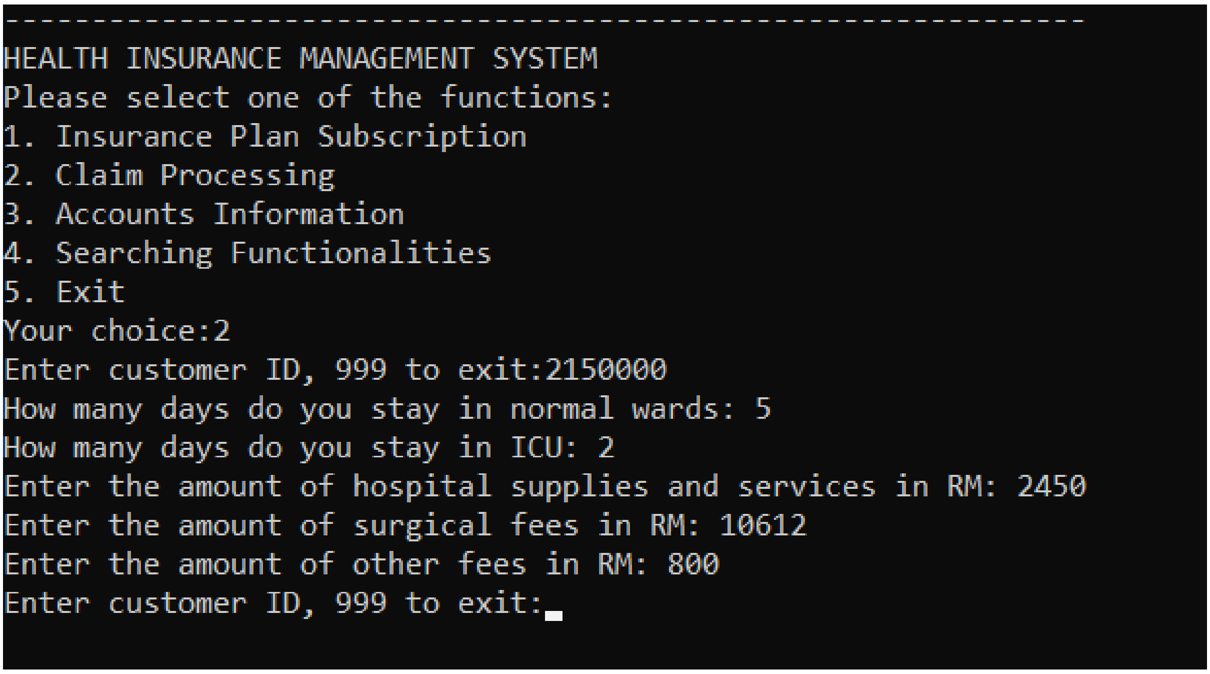


Claim_plan150.txt - Notepad

```
ID                CLAIMED_YEAR              AMOUNT_CLAIMED            BALANCE(YEAR)(RM)         BALANCE(LIFETIME)(RM)
2150000           N/A                                 0.00                  150000.00                       0.00
```

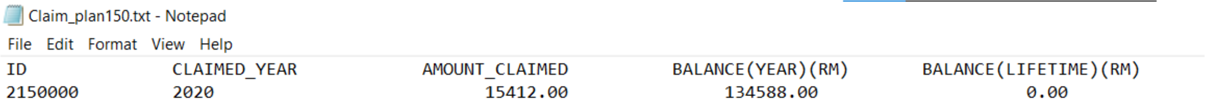
The related information will be stored to the relevant text file immediately.

4.3 Claiming Processing

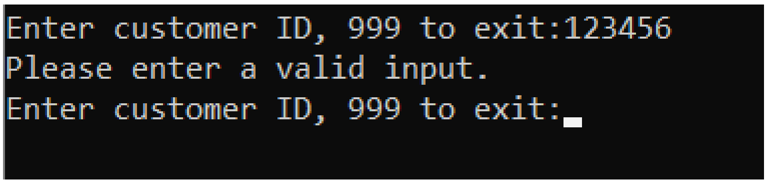


```
-------------------------------------------------------------
HEALTH INSURANCE MANAGEMENT SYSTEM
Please select one of the functions:
1. Insurance Plan Subscription
2. Claim Processing
3. Accounts Information
4. Searching Functionalities
5. Exit
Your choice:2
Enter customer ID, 999 to exit:2150000
How many days do you stay in normal wards: 5
How many days do you stay in ICU: 2
Enter the amount of hospital supplies and services in RM: 2450
Enter the amount of surgical fees in RM: 10612
Enter the amount of other fees in RM: 800
Enter customer ID, 999 to exit:
```

The integer “2” will lead the users to the claiming process. The system will ask the customers a list of questions and implement the calculation. After that, the result of calculation and the year of claiming the insurance will be stored in the text file immediately.

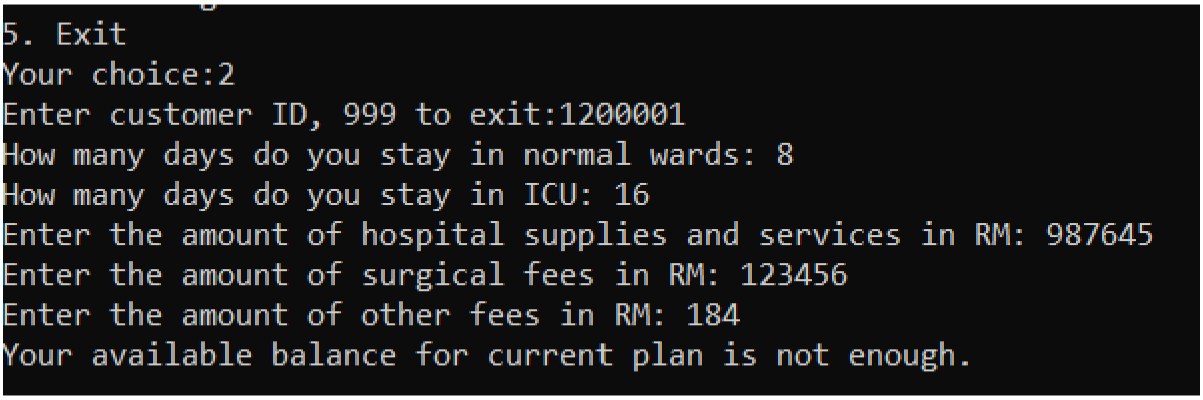


```
ID              CLAIMED_YEAR            AMOUNT_CLAIMED          BALANCE(YEAR)(RM)       BALANCE(LIFETIME)(RM)
2150000         2020                          15412.00                134588.00                    0.00
```



```
Enter customer ID, 999 to exit:123456
Please enter a valid input.
Enter customer ID, 999 to exit:
```

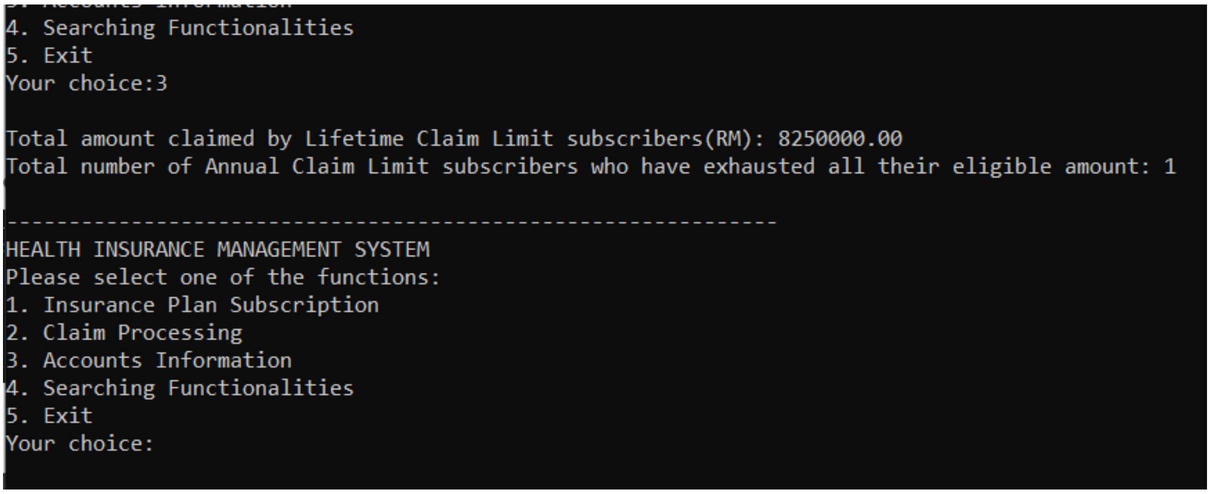
If the users didn’t enter the correct ID, the error message will then be generated. Integer ‘999’ will let the system return to the main menu.



```
5. Exit
Your choice:2
Enter customer ID, 999 to exit:1200001
How many days do you stay in normal wards: 8
How many days do you stay in ICU: 16
Enter the amount of hospital supplies and services in RM: 987645
Enter the amount of surgical fees in RM: 123456
Enter the amount of other fees in RM: 184
Your available balance for current plan is not enough.
```

If the claimed amount excess the balance, the message will be shown.

4.4 Accounts Information

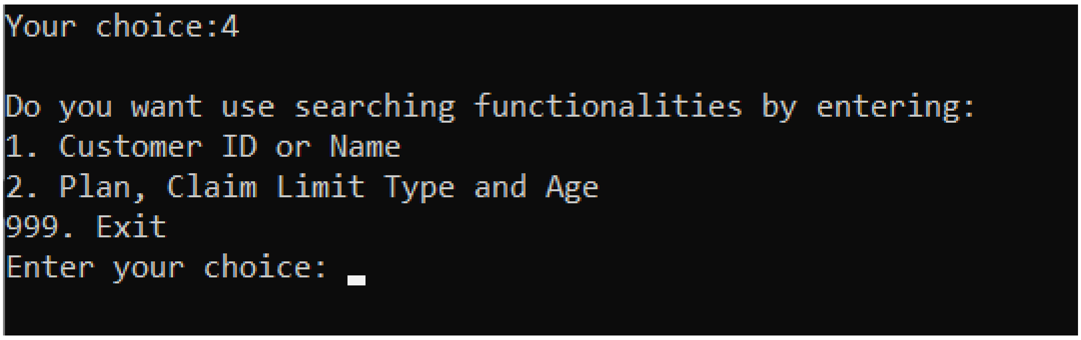


```
4. Searching Functionalities
5. Exit
Your choice:3

Total amount claimed by Lifetime Claim Limit subscribers(RM): 8250000.00
Total number of Annual Claim Limit subscribers who have exhausted all their eligible amount: 1

--------------------------------------------------------------
HEALTH INSURANCE MANAGEMENT SYSTEM
Please select one of the functions:
1. Insurance Plan Subscription
2. Claim Processing
3. Accounts Information
4. Searching Functionalities
5. Exit
Your choice:
```

The third function is providing the account information that shown as the figure above. It will provide the total amount claimed by Lifetime Claim Limit subscribers in RM and total number of Annual Claim Limit subscribers who have exhausted all their eligible amount.

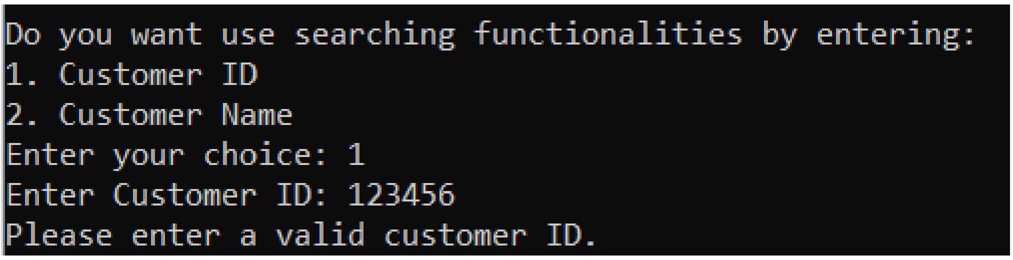
After that, the system will then back to the main menu.

4.5 Searching Functionalities



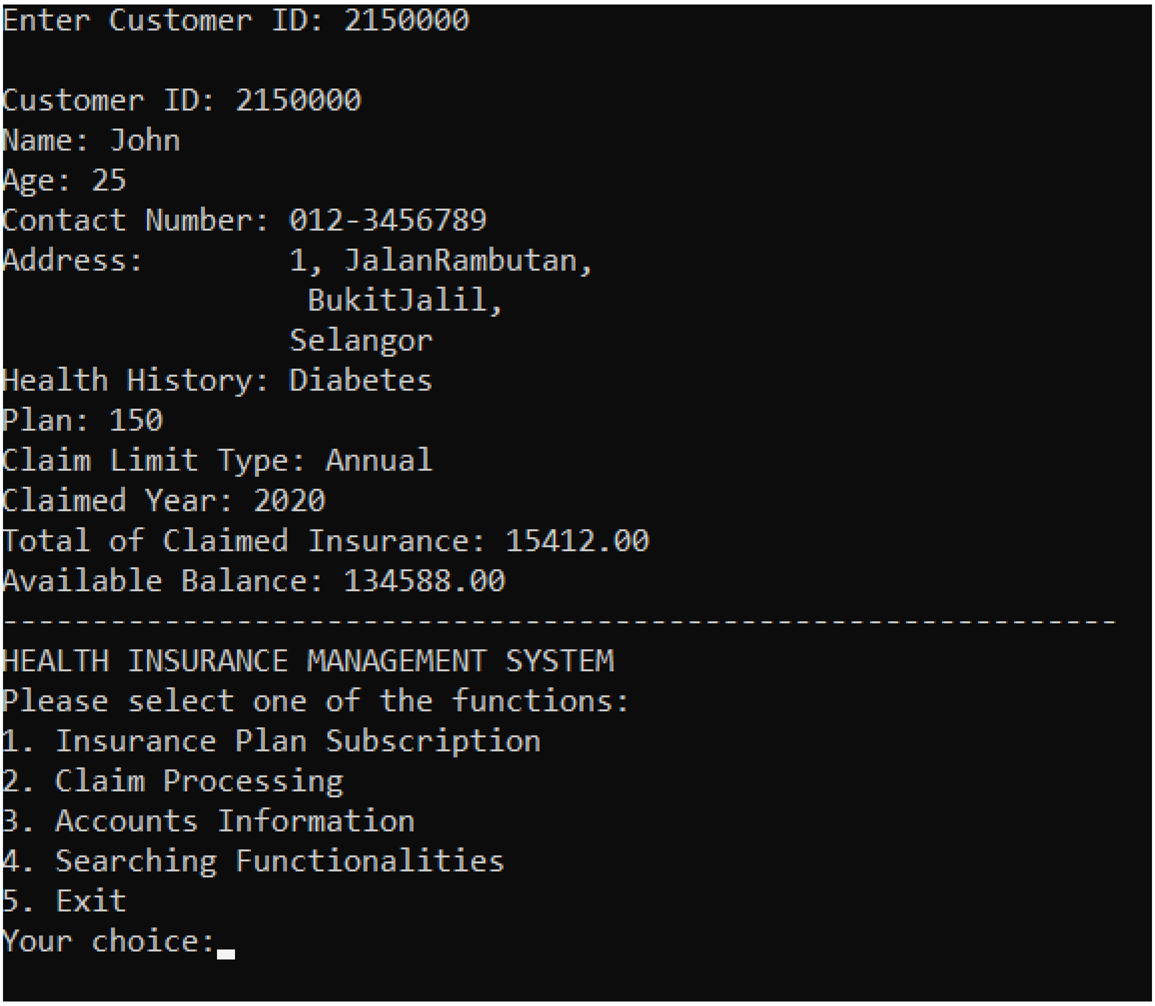
```
Your choice:4

Do you want use searching functionalities by entering:
1. Customer ID or Name
2. Plan, Claim Limit Type and Age
999. Exit
Enter your choice: _
```

The last function is the searching function. It allows users to search the information of customers and chosen plan according to their ID or name or plan type, claim limit type and age.



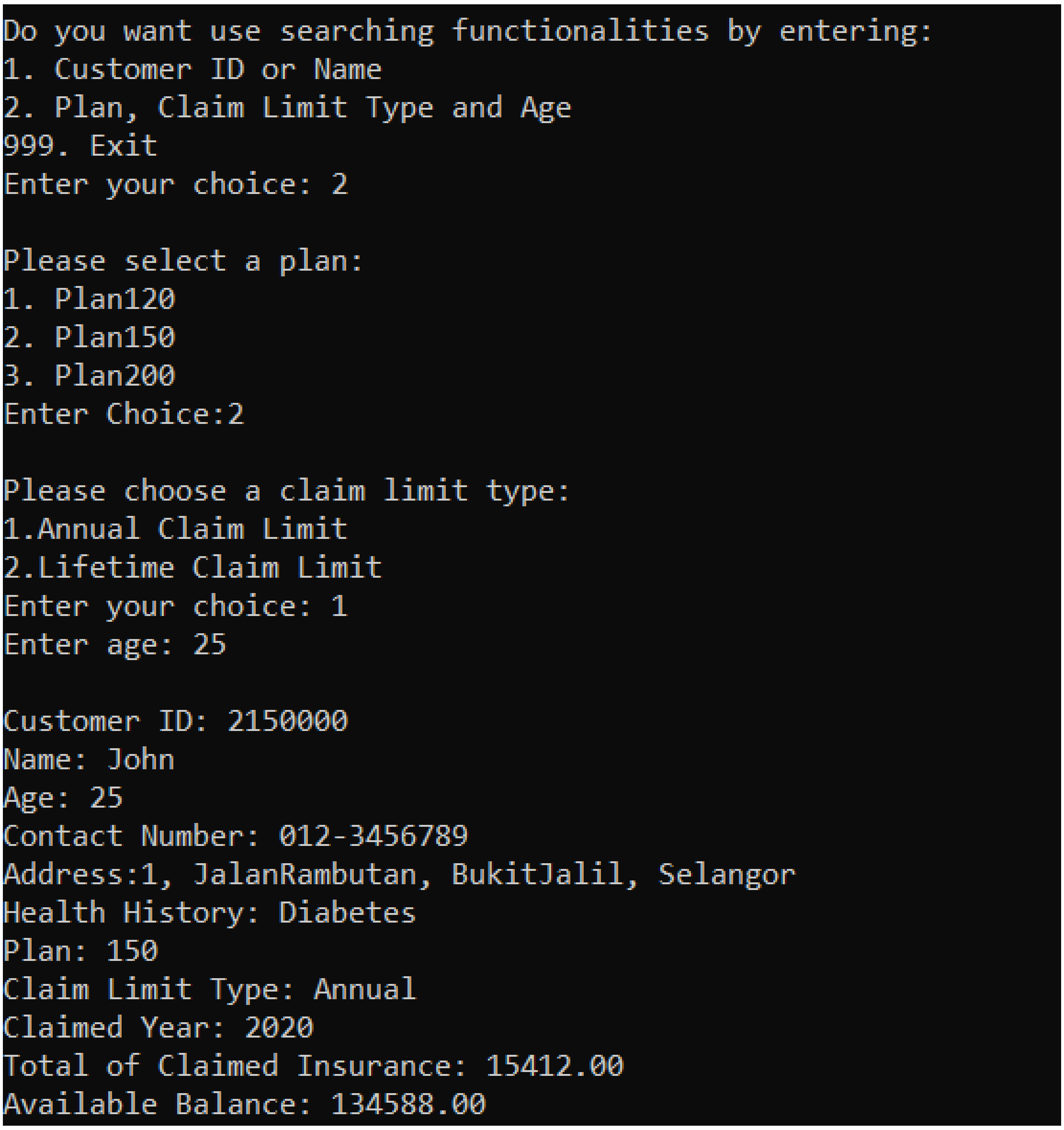
```
Do you want use searching functionalities by entering:
1. Customer ID
2. Customer Name
Enter your choice: 1
Enter Customer ID: 123456
Please enter a valid customer ID.
```

If the customer ID is not found in the text file, the error message will then be printed on the screen.



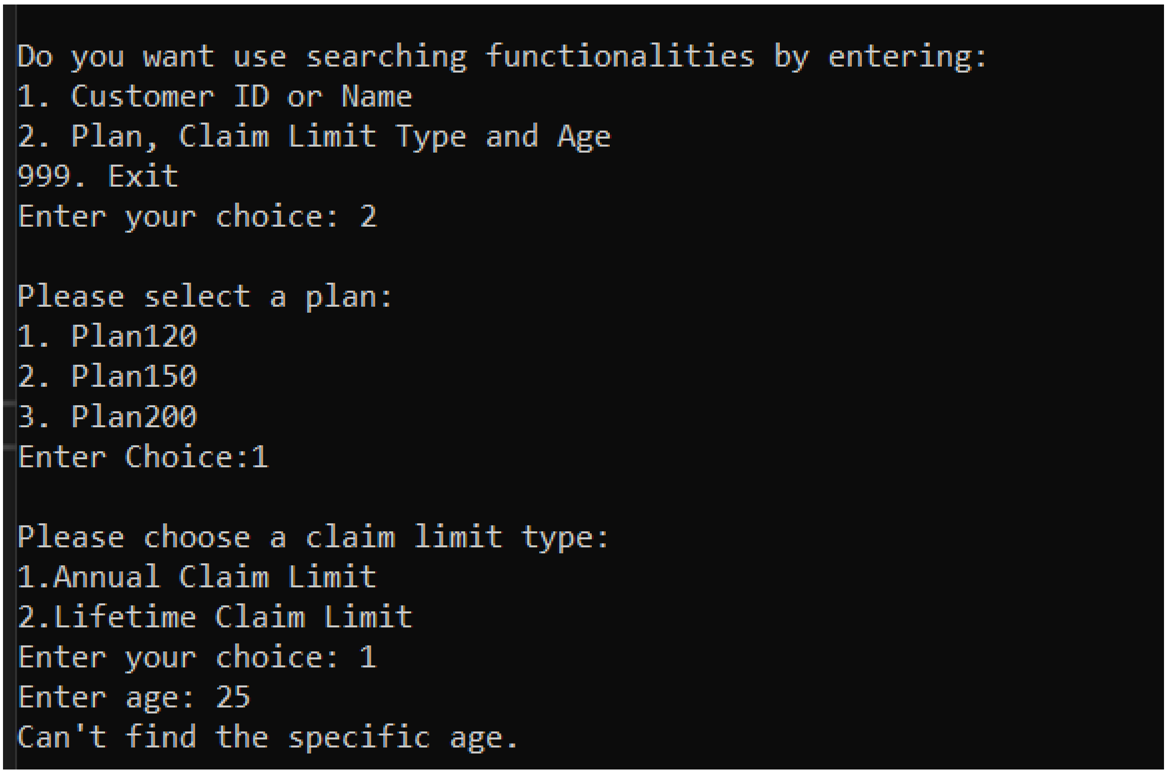
```
Enter Customer ID: 2150000

Customer ID: 2150000
Name: John
Age: 25
Contact Number: 012-3456789
Address:        1, JalanRambutan,
                 BukitJalil,
                Selangor
Health History: Diabetes
Plan: 150
Claim Limit Type: Annual
Claimed Year: 2020
Total of Claimed Insurance: 15412.00
Available Balance: 134588.00
--------------------------------------------------------------
HEALTH INSURANCE MANAGEMENT SYSTEM
Please select one of the functions:
1. Insurance Plan Subscription
2. Claim Processing
3. Accounts Information
4. Searching Functionalities
5. Exit
Your choice:_
```

If the entered input was found in the text file, the related information will then be printed out line by line. After that, the program will automatically back to the main menu for the access to other functionalities.



```
Do you want use searching functionalities by entering:
1. Customer ID or Name
2. Plan, Claim Limit Type and Age
999. Exit
Enter your choice: 2

Please select a plan:
1. Plan120
2. Plan150
3. Plan200
Enter Choice:2

Please choose a claim limit type:
1.Annual Claim Limit
2.Lifetime Claim Limit
Enter your choice: 1
Enter age: 25

Customer ID: 2150000
Name: John
Age: 25
Contact Number: 012-3456789
Address:1, JalanRambutan, BukitJalil, Selangor
Health History: Diabetes
Plan: 150
Claim Limit Type: Annual
Claimed Year: 2020
Total of Claimed Insurance: 15412.00
Available Balance: 134588.00
```

Besides, users can also search the information according to the plan type, claim limit type and age. However, if one of the inputs is wrong, the system will not find the related information.



```
Do you want use searching functionalities by entering:
1. Customer ID or Name
2. Plan, Claim Limit Type and Age
999. Exit
Enter your choice: 2

Please select a plan:
1. Plan120
2. Plan150
3. Plan200
Enter Choice:1

Please choose a claim limit type:
1.Annual Claim Limit
2.Lifetime Claim Limit
Enter your choice: 1
Enter age: 25
Can't find the specific age.
```

5.0 Conclusion

Health Insurance Management System helps ZeeMediLife insurance company to record the information of their customers' details. It is user-friendly and suitable for any users who know English. Compared to other applications, this well-designed system is more straightforward.

In the code, the program was written in C Programming Language using Visual Studio 2019. The whole program is menu-driven. After completing a function, users will back to the menu and choose the other function again to perform the following steps. The program is completed using basic C programming knowledge such as variables, pointers, for and while loops, structures, and arrays.

In the ICP module, I have learnt the powerful and efficient C programming language. It is considered as the most fundamental programming language that must be learnt by every programmer.

6.0 References

Stack Overflow. 2020. *Stack Overflow - Where Developers Learn, Share, & Build Careers*. [online] Available at: <https://stackoverflow.com/>
[Accessed 4 November 2020].