

1. Table of Content

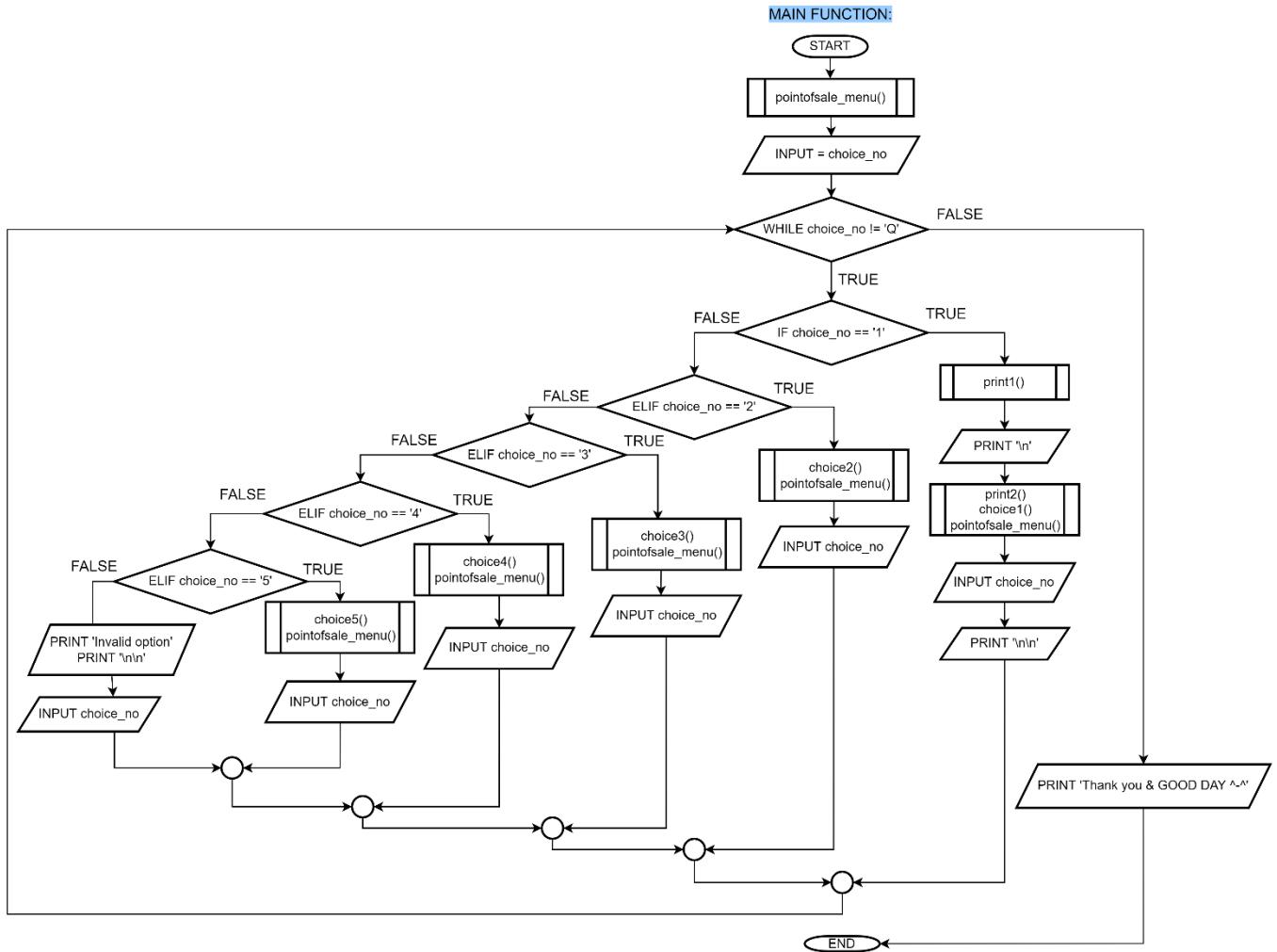
Content	Page Number
1. Task Distribution	3
2. Flowchart:	4 - 44
3. Screenshots: (A) Main Menu (B) Item Maintenance (i) Adding New Item (ii) New Item Added (iii) Deleting Item (iv) Item Deleted (v) Modifying Item (vi) Item Modified (C) Inventory Shortfall (D) Membership Maintenance (i) Introduction of Membership (ii) Membership Registration (iii) Checking Membership Status (E) Adding Item for Payment (F) Payment (G) Receipt (H) Report (I) Data Validation	45 - 54
4. Source-code (program code):	55 - 71

1. Task Distribution

Name	Task
Keh Yi Qian	Coding for <1>Item Maintenance, <4> Sales, and <5> Report and Flowchart for <5> Report. Check the errors in the code and modify the code.
Chin Zi Wei	Coding and Flowchart for <2> Stock/Inventory Management, <3> Membership Maintenance. Check the errors in the code
Tan Kok Fu	Coding <3> Membership Maintenance Check the errors in the code
Liang Kwan Yee	Flowchart for <1>Item Maintenance, <4> Sales. Word Document Report. Check the errors in the code

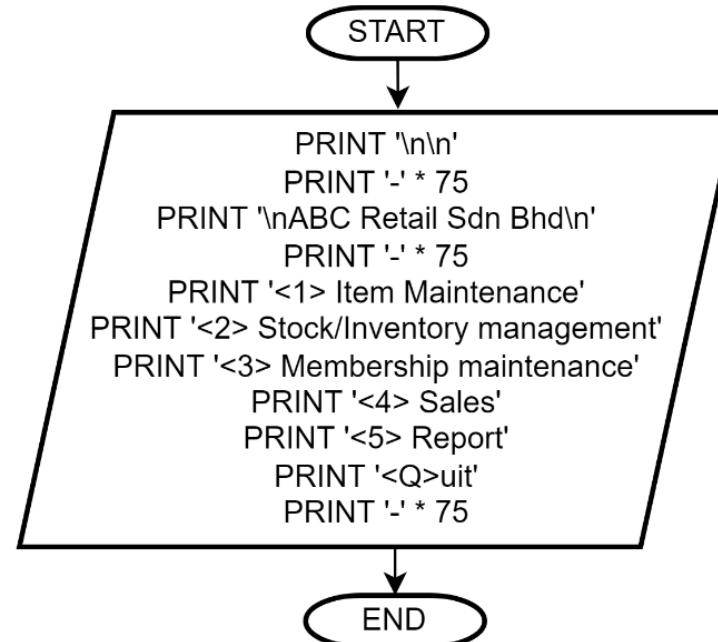
2. Flowcharts

(Draw the flowchart using draw.io or any other flowchart software, copy the pictures and paste it here. Do not draw by hand. Make sure the flowchart is clear. Use page connector to connect the flowchart if the flowchart cannot fit in one page.)

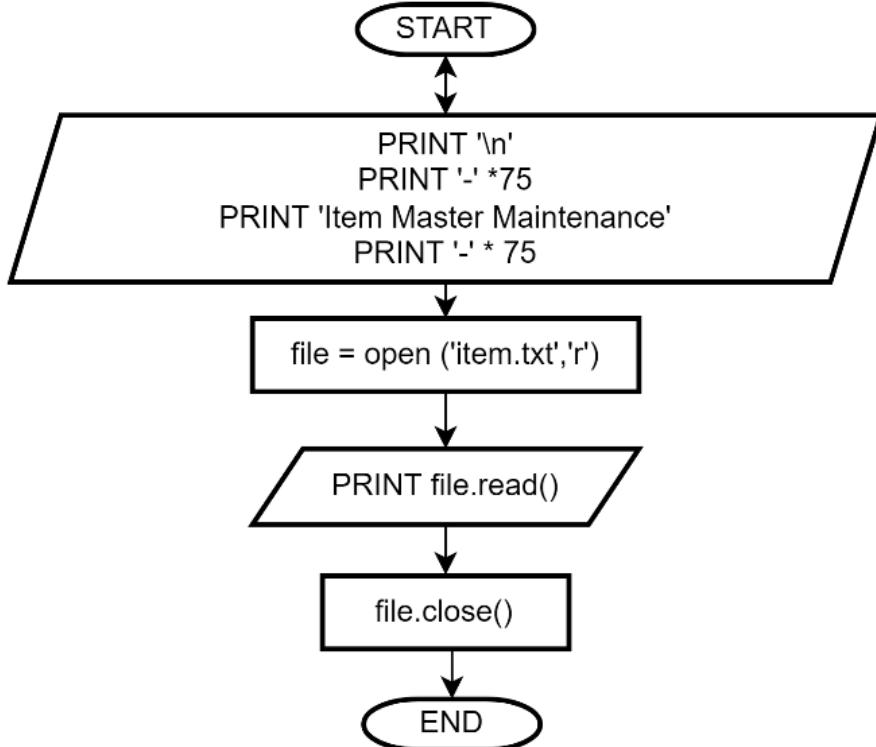


Z)

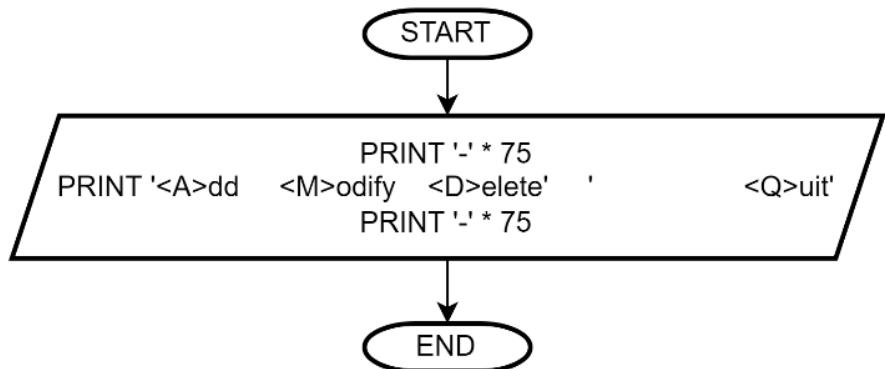
```
def pointofsale_menu():
```

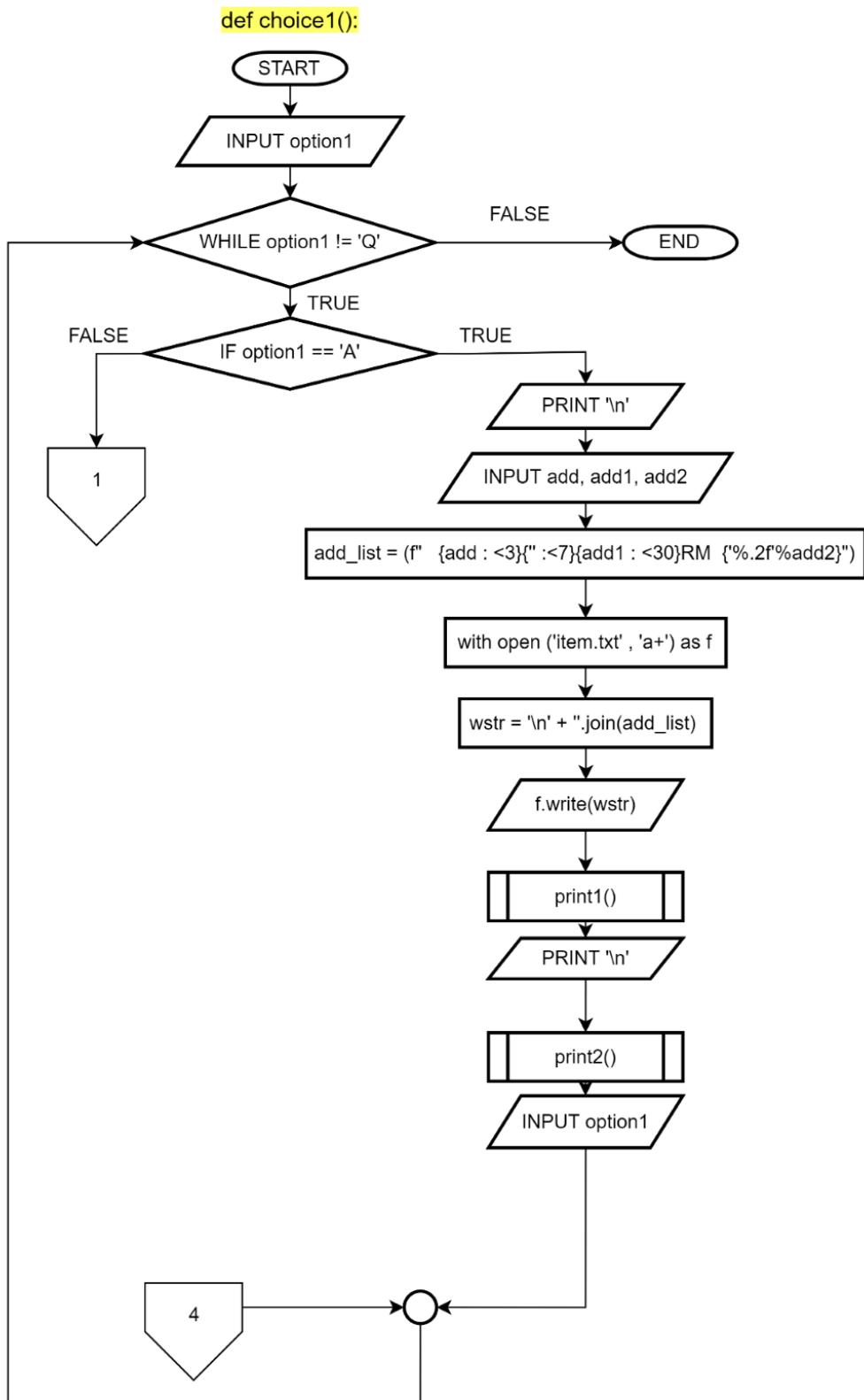


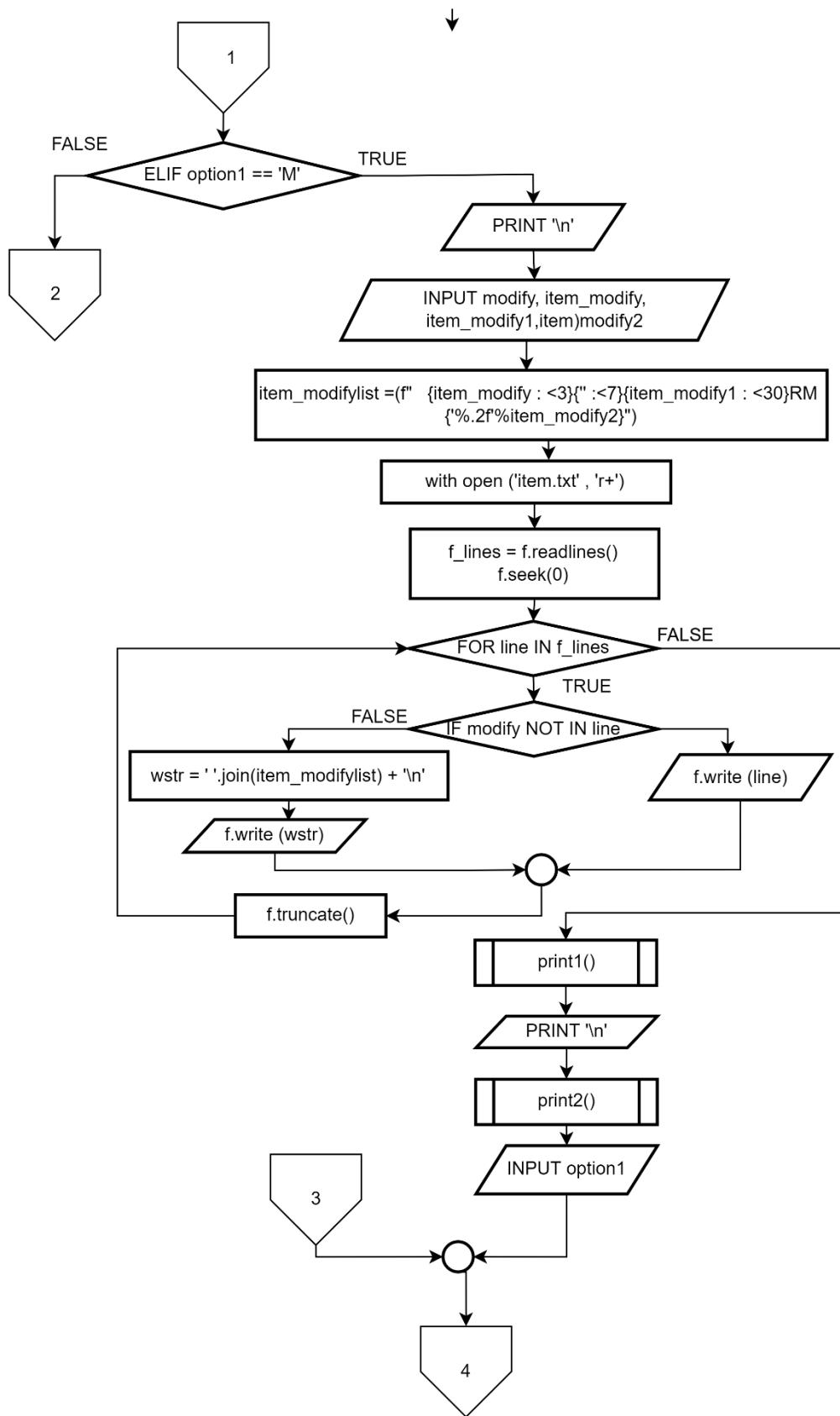
```
def print1():
```

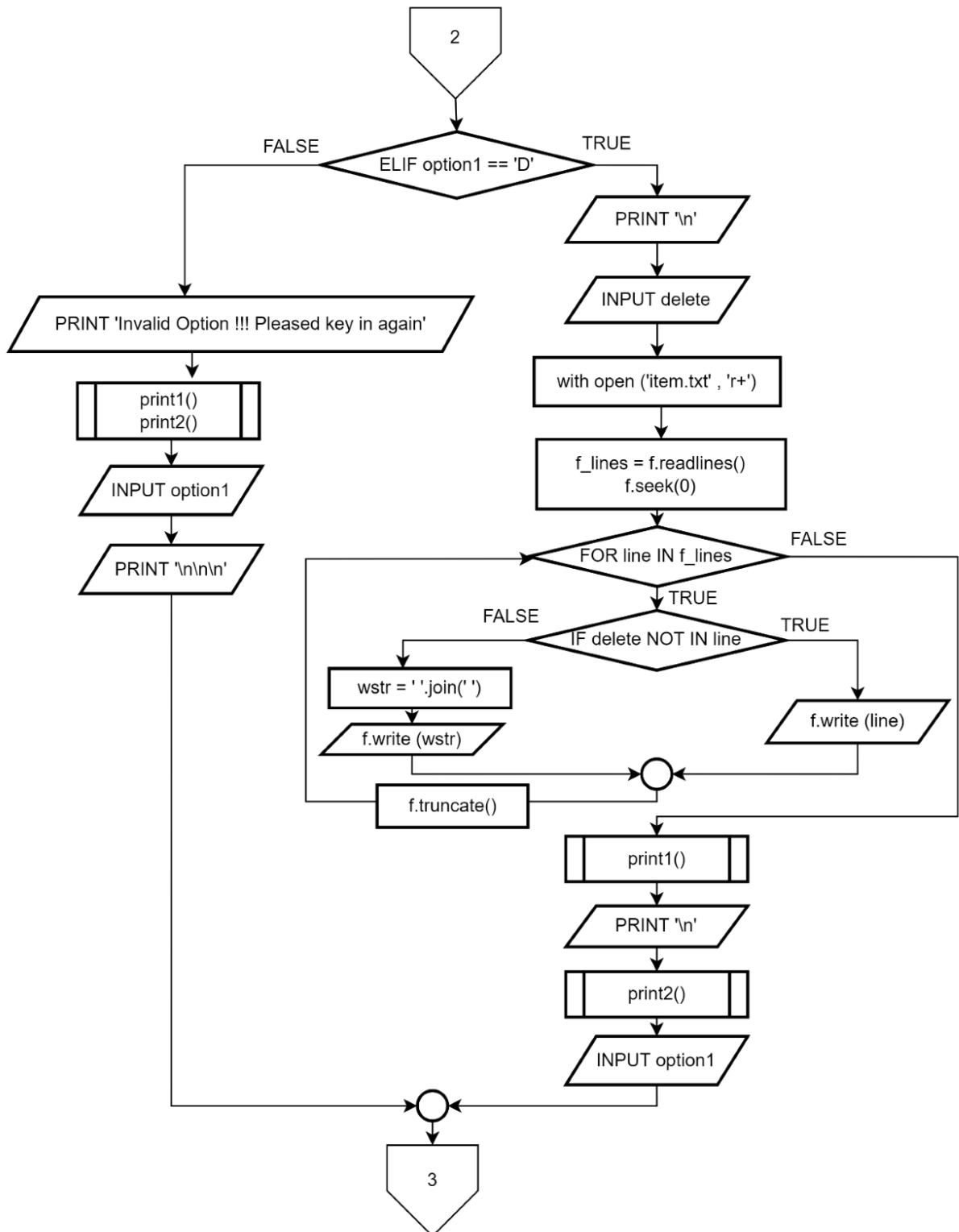


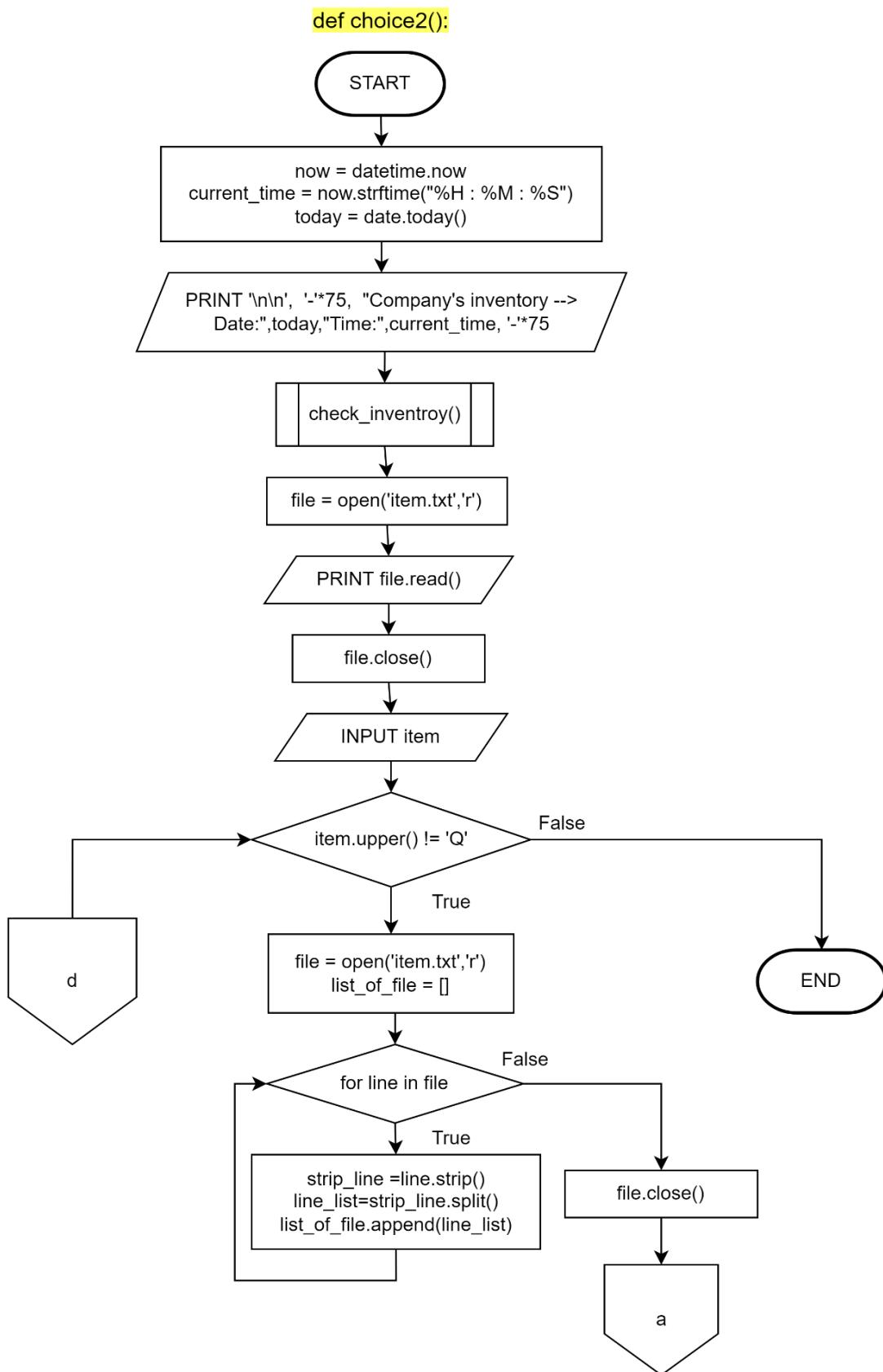
```
def print2():
```

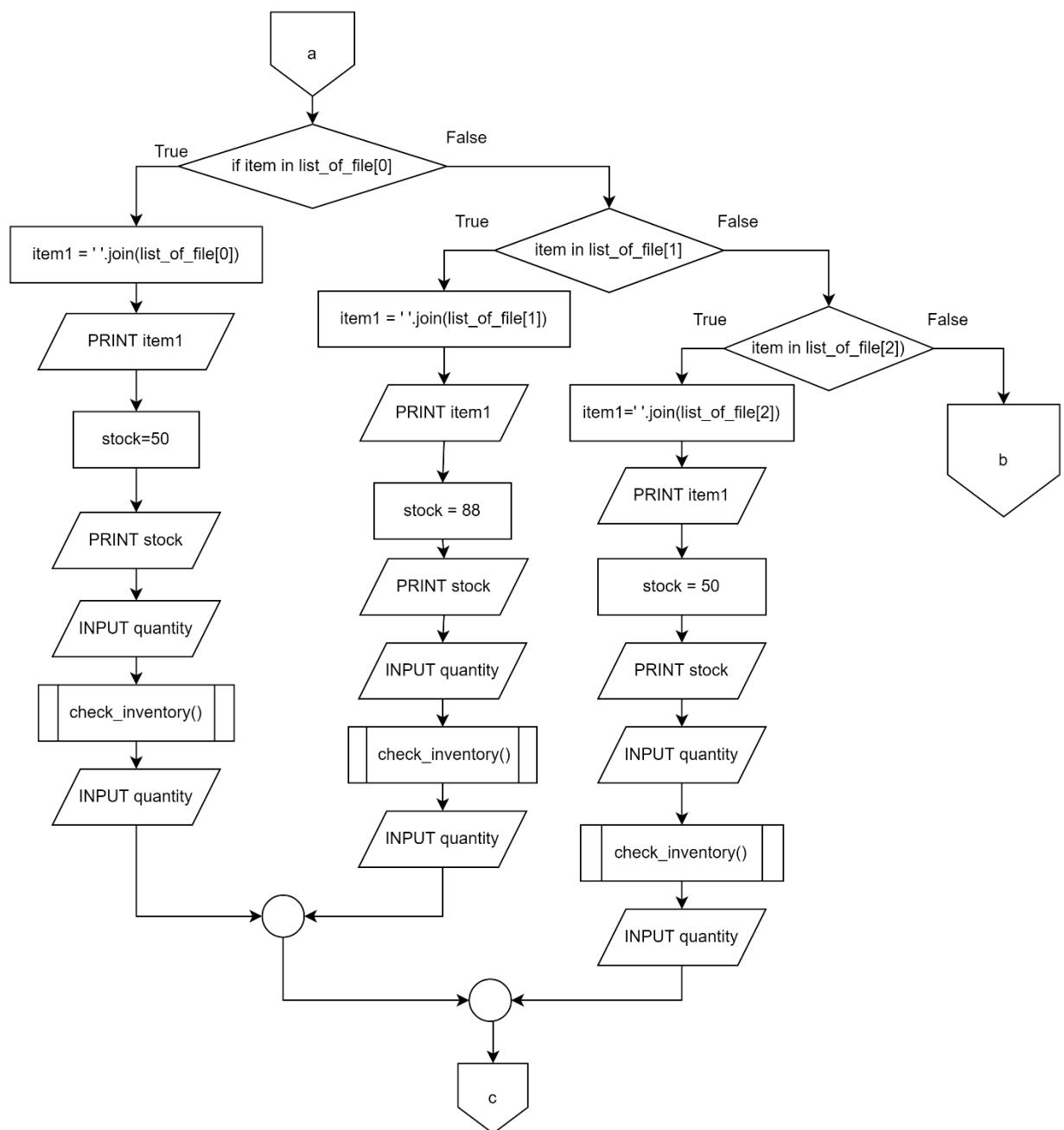


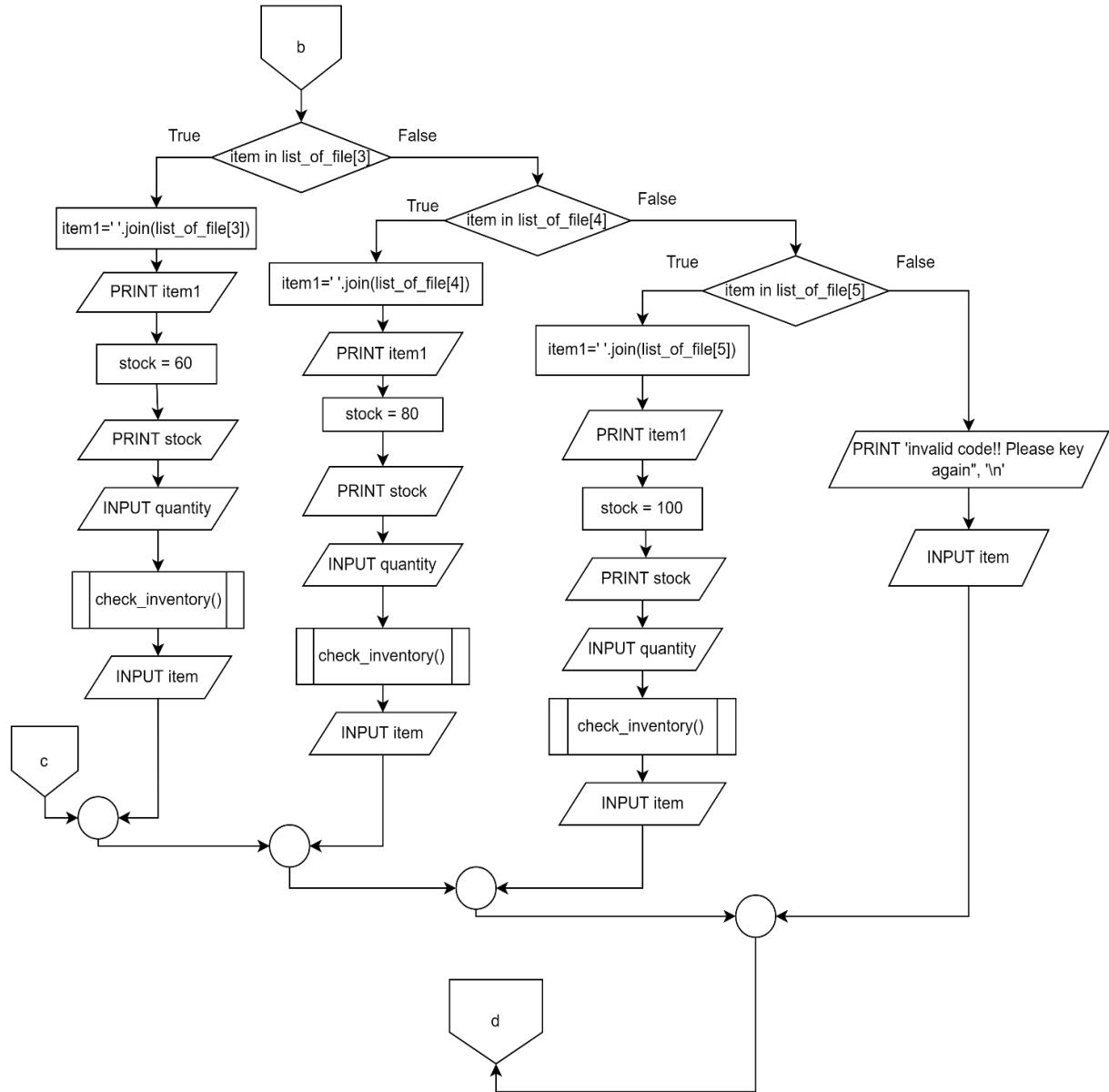




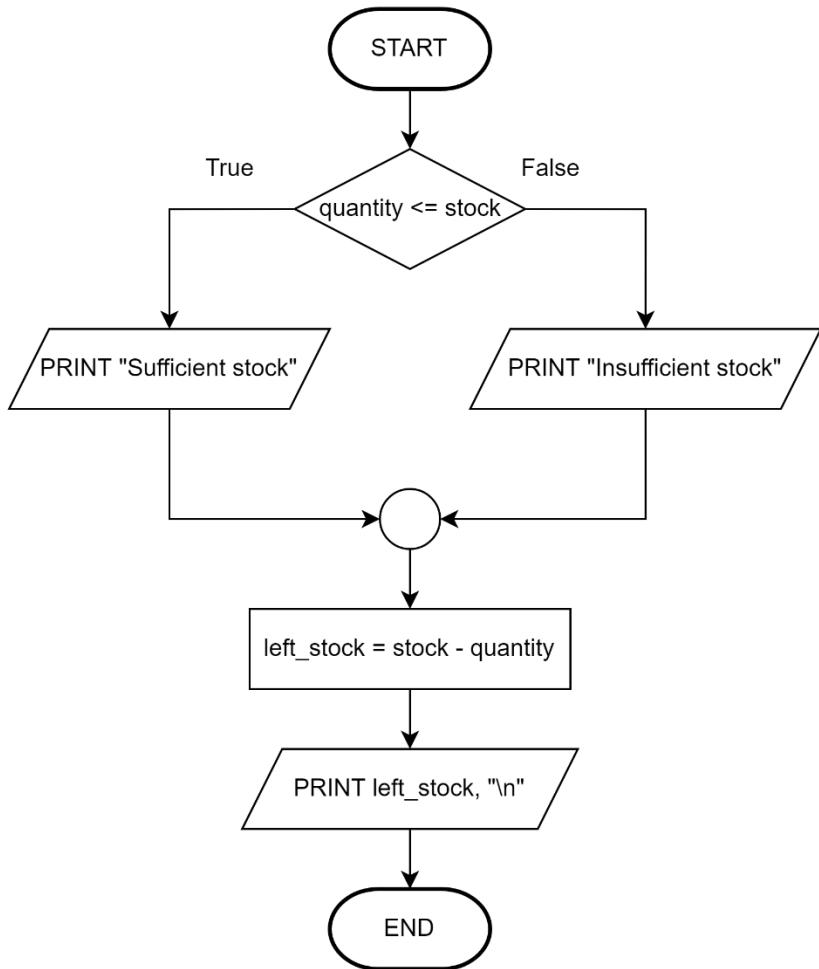




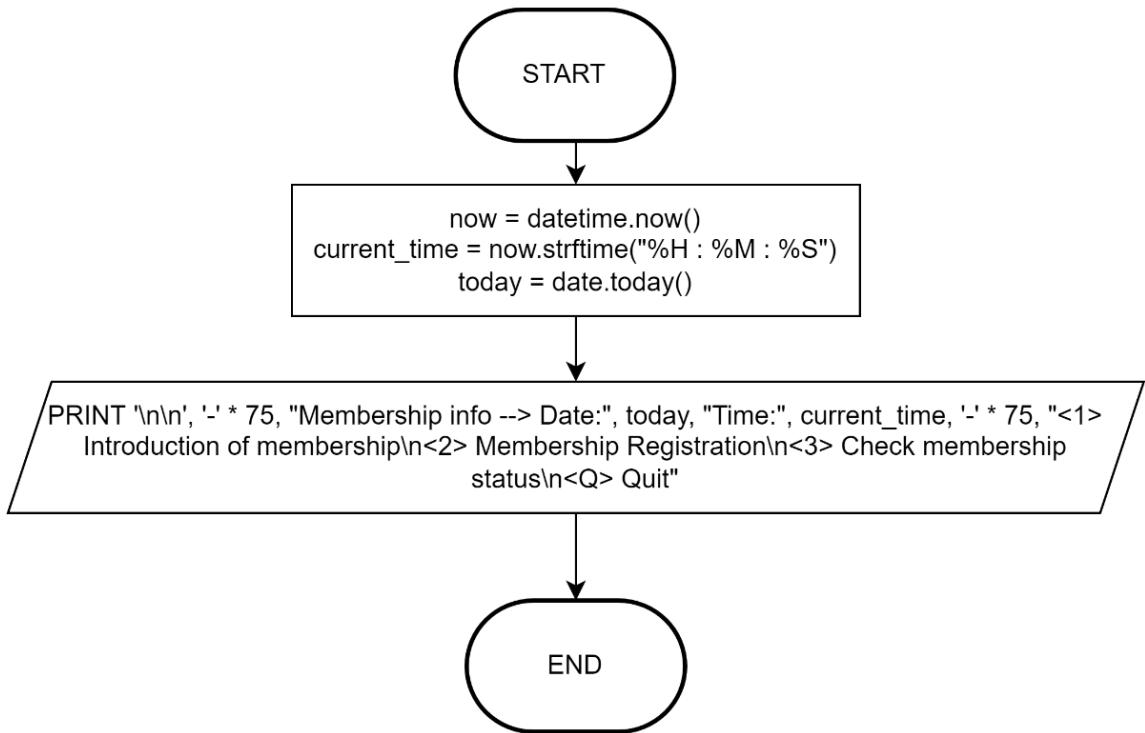


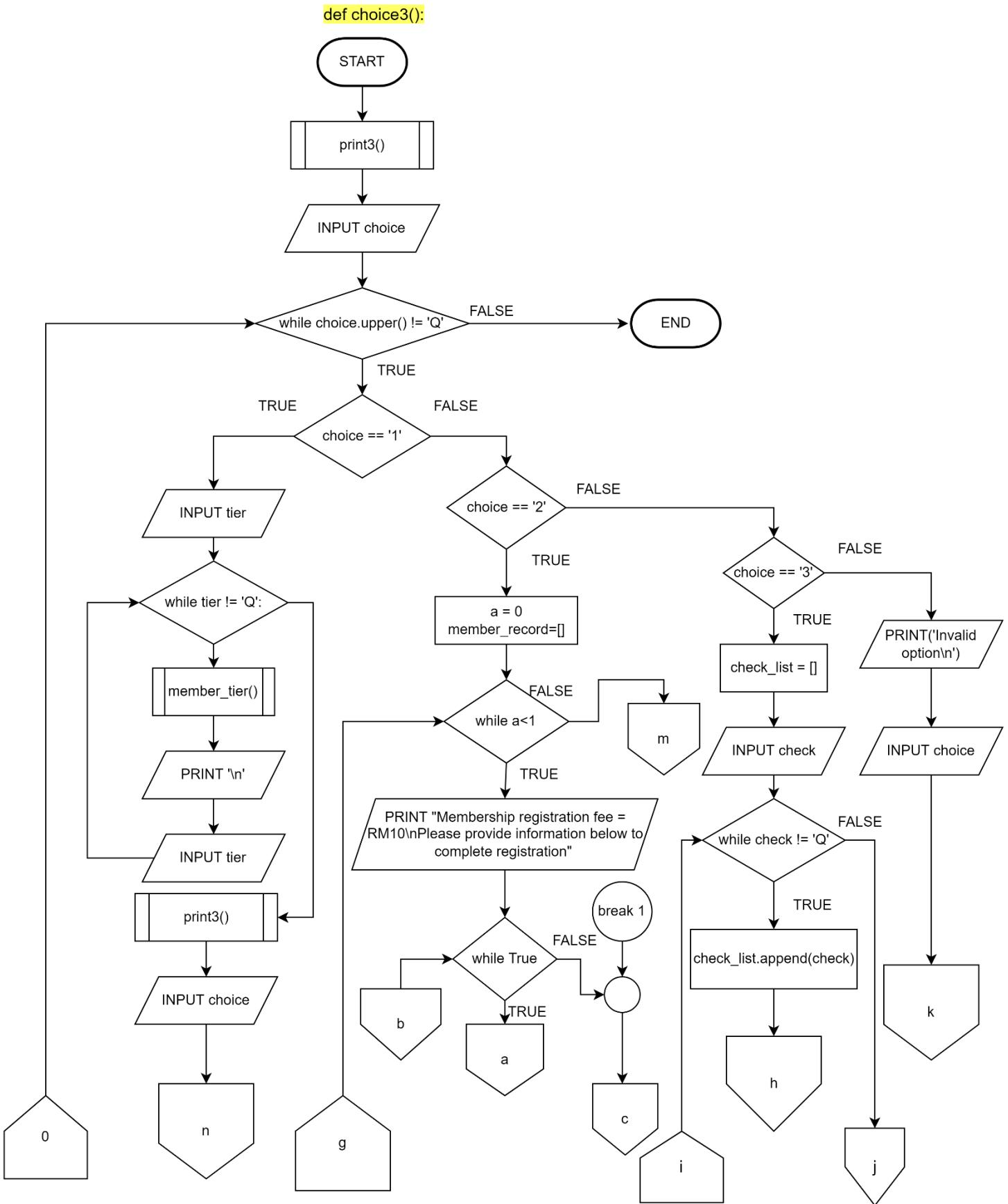


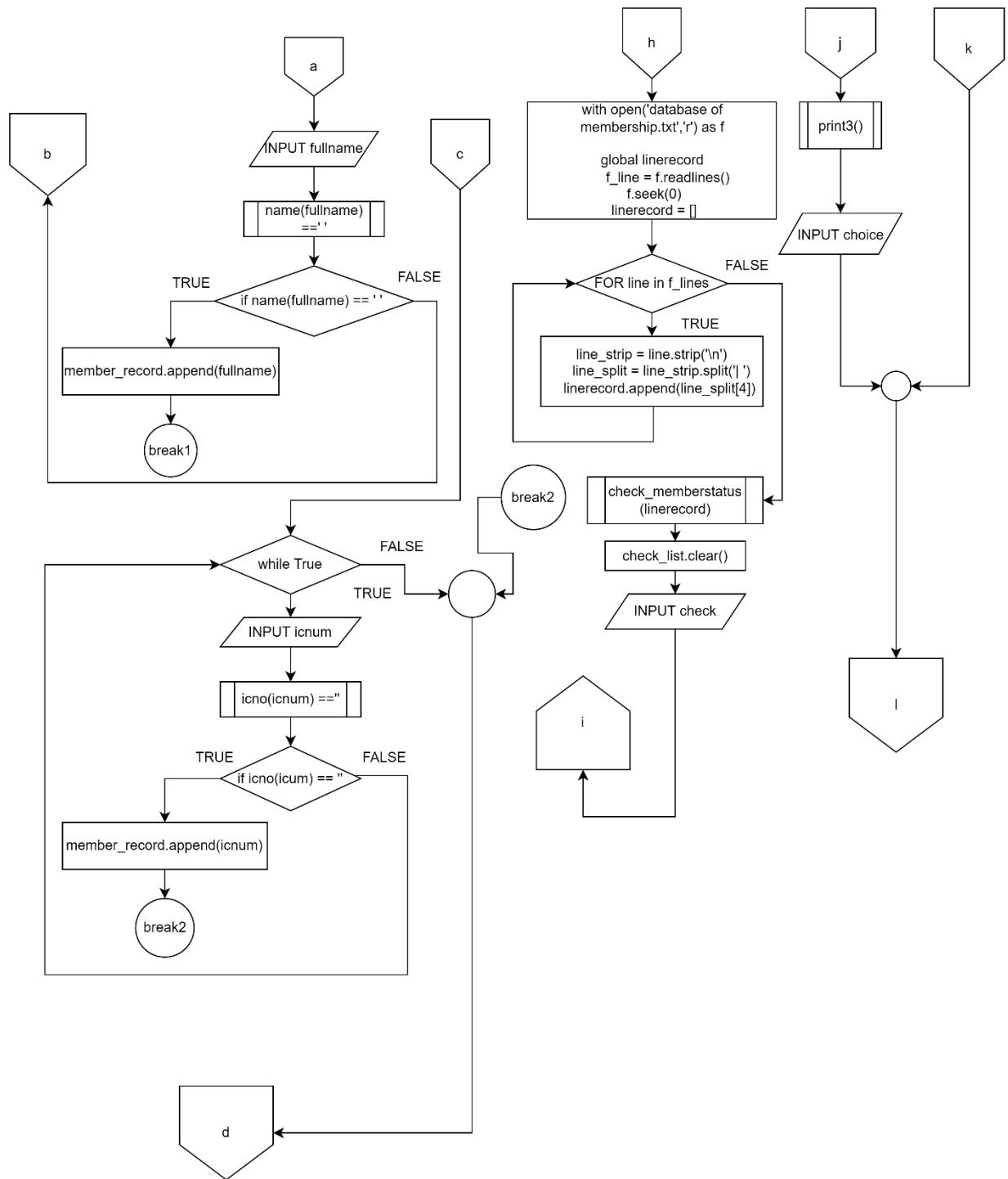
```
def check_inventory():
```

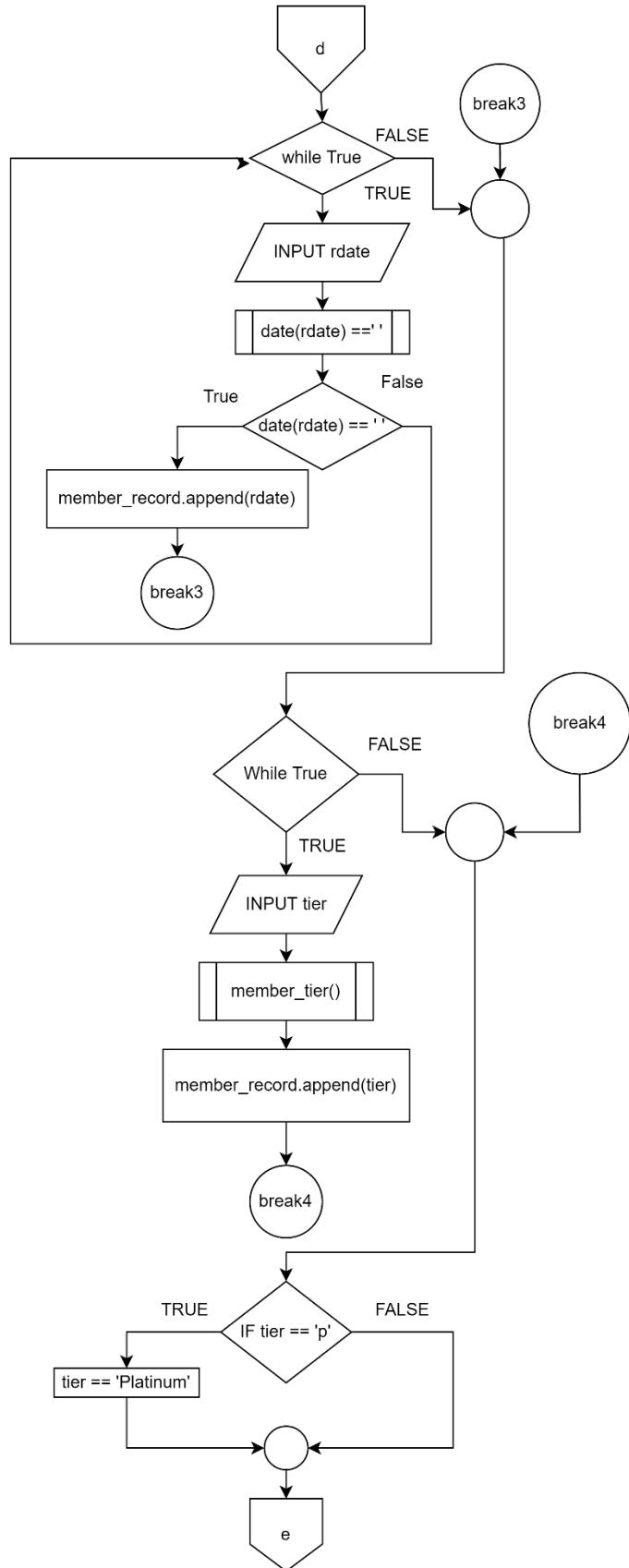


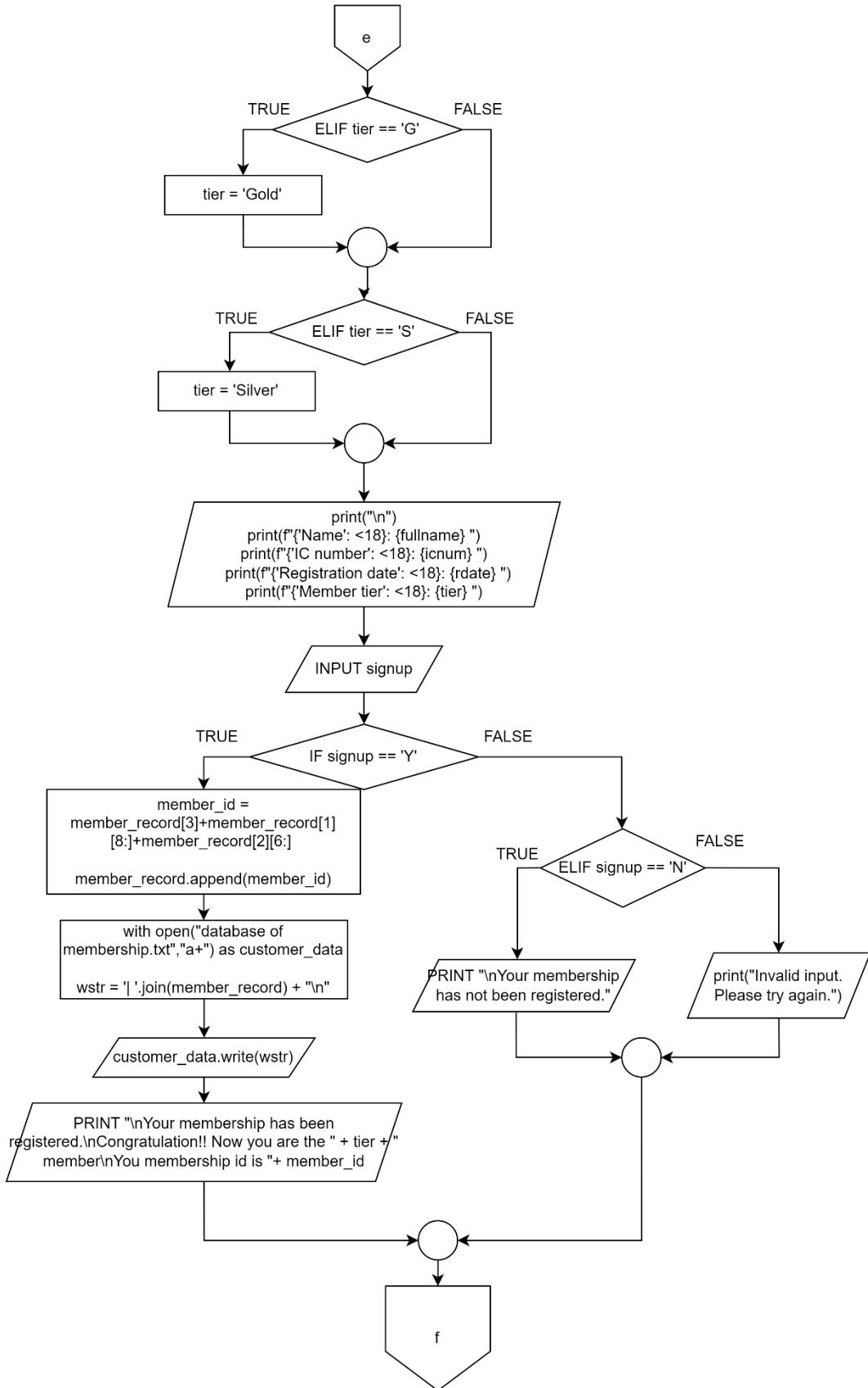
```
def print3():
```

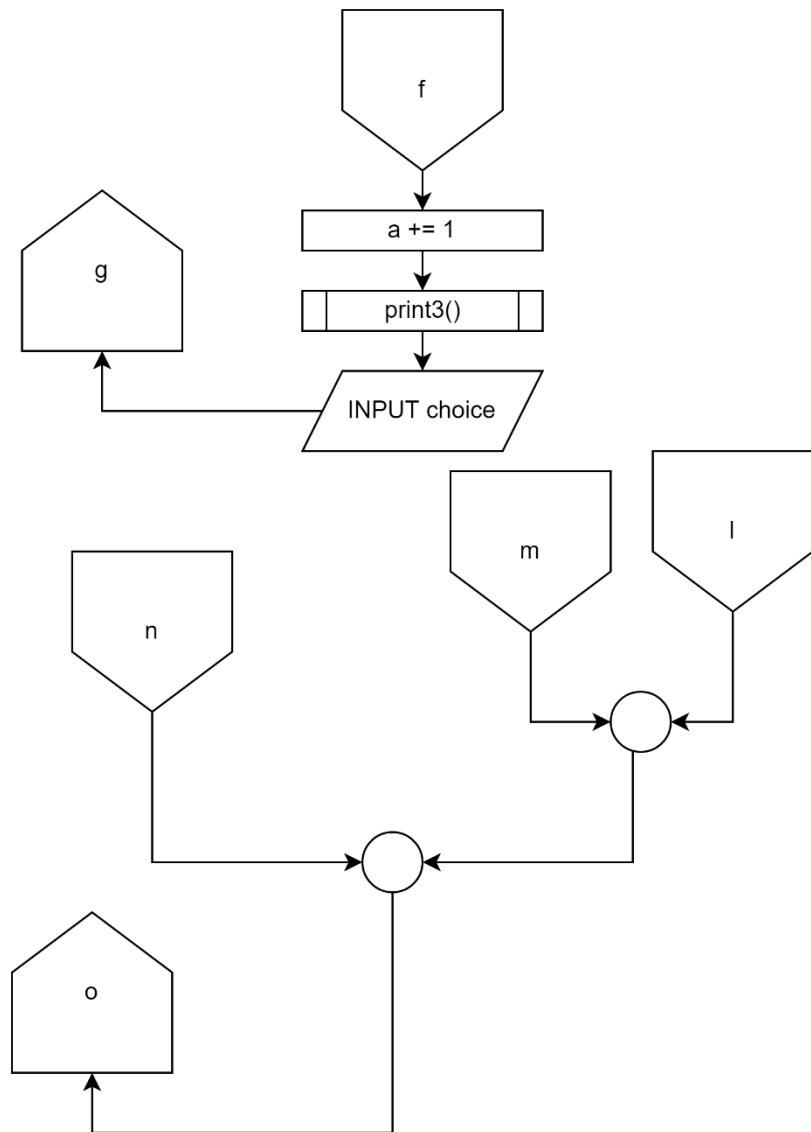




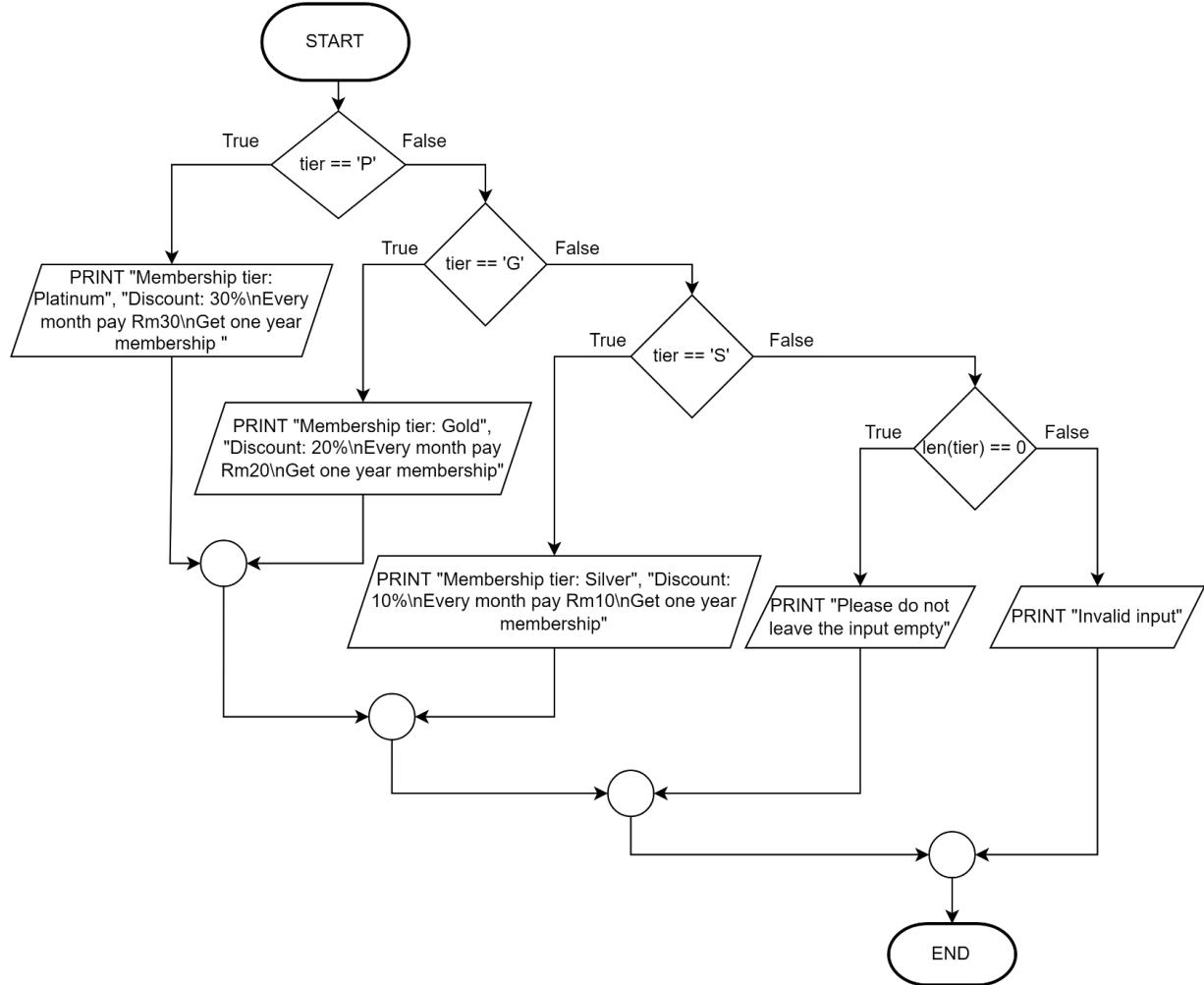




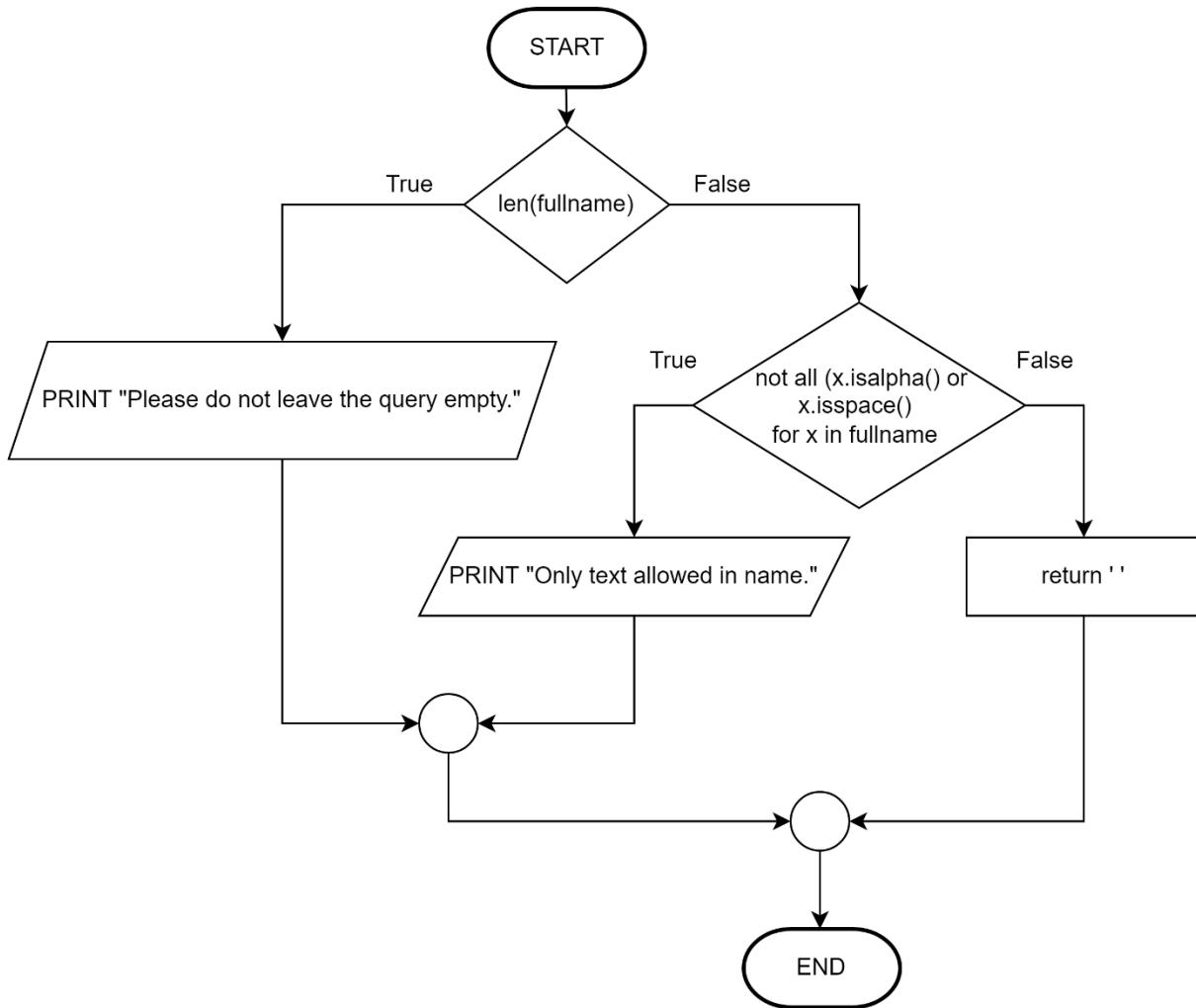




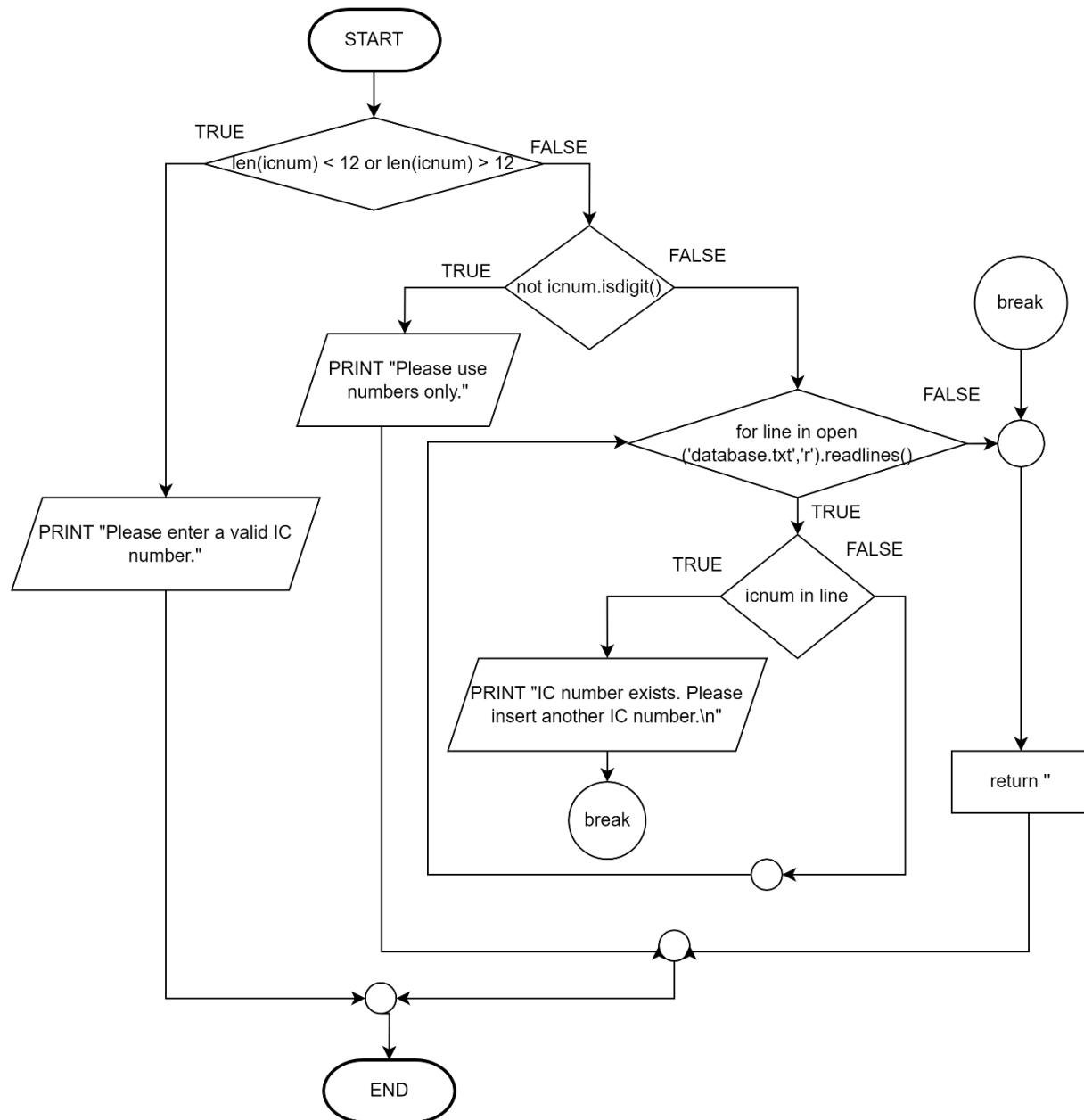
```
def member_tier():
```



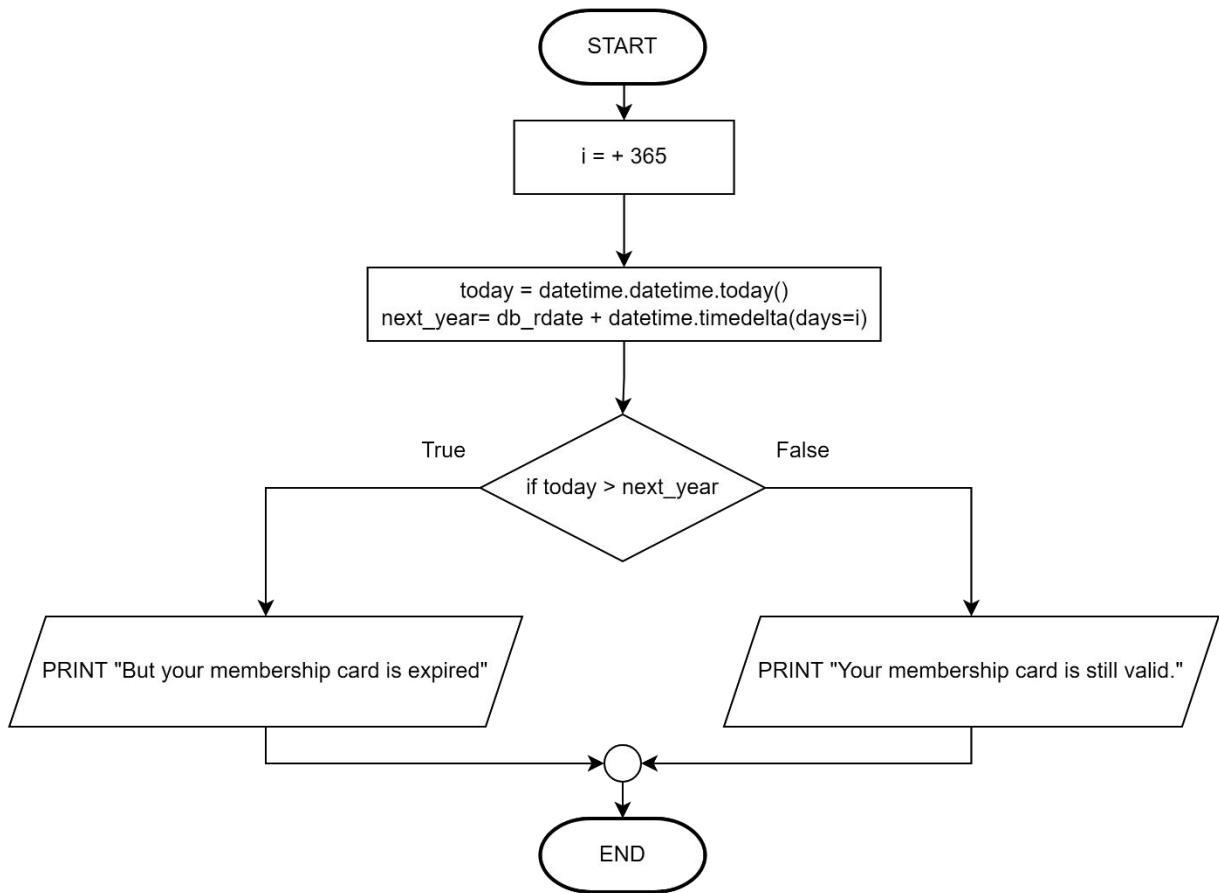
```
def name(fullname):
```

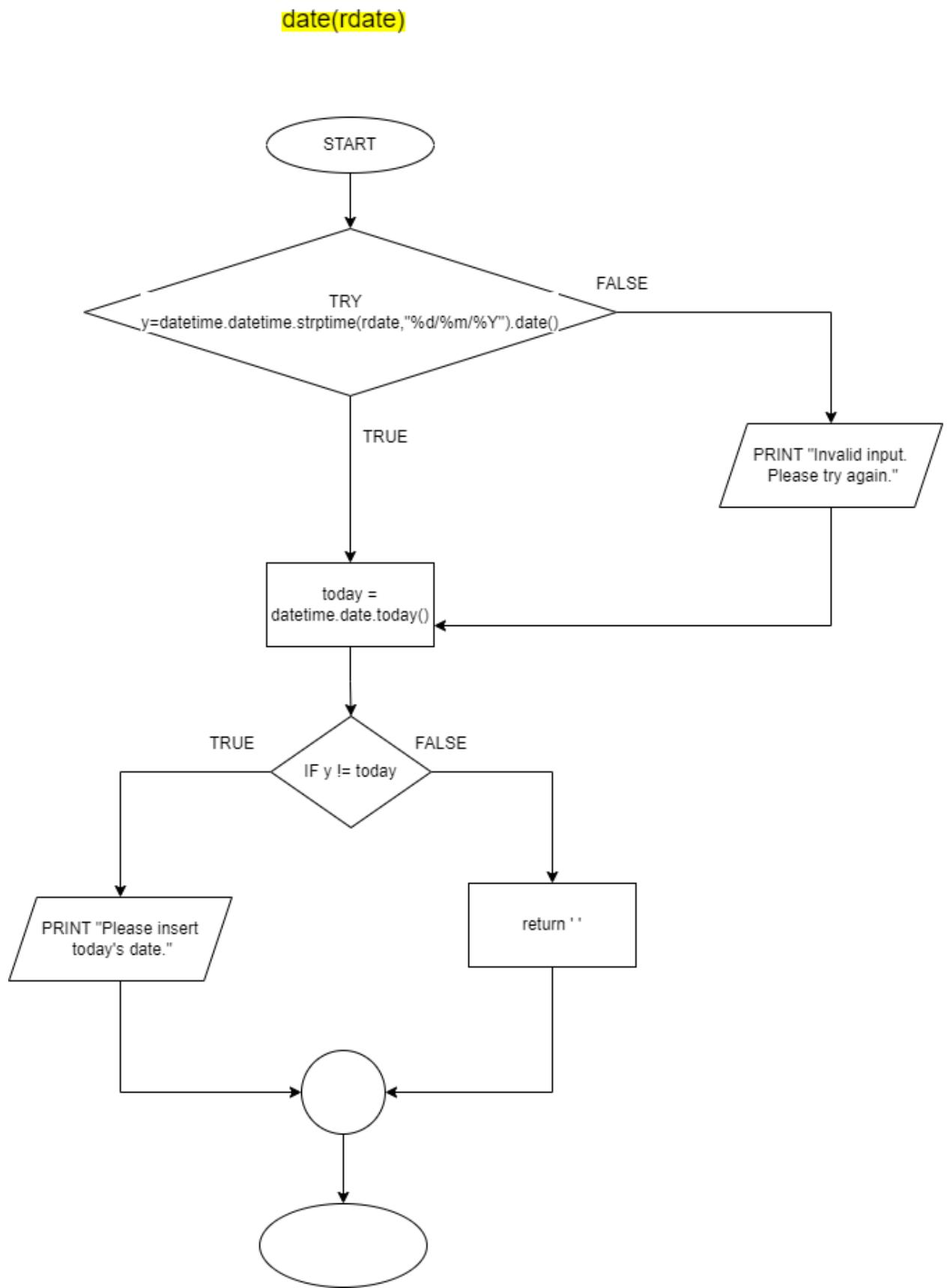


```
def icno(icnum):
```

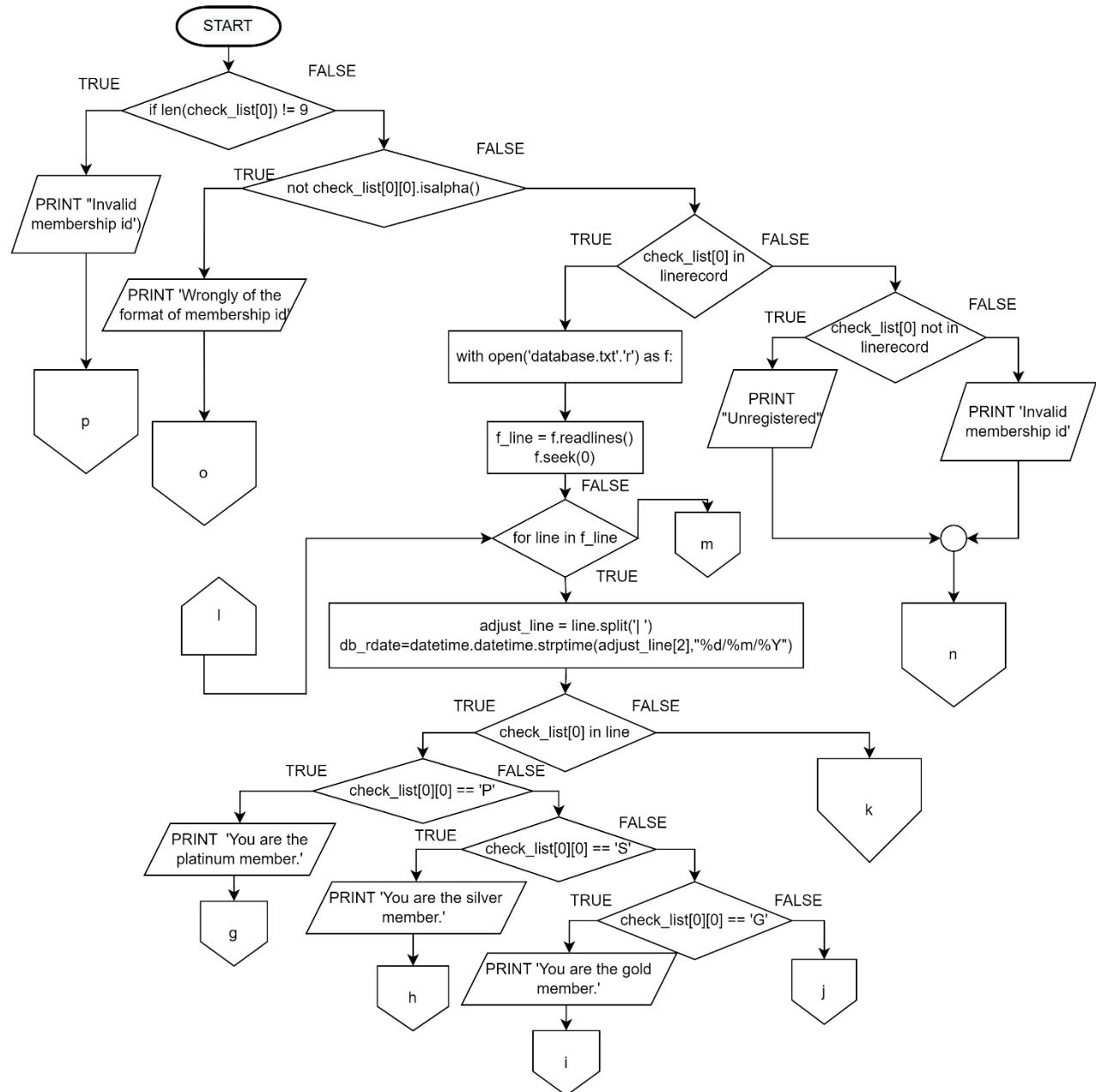


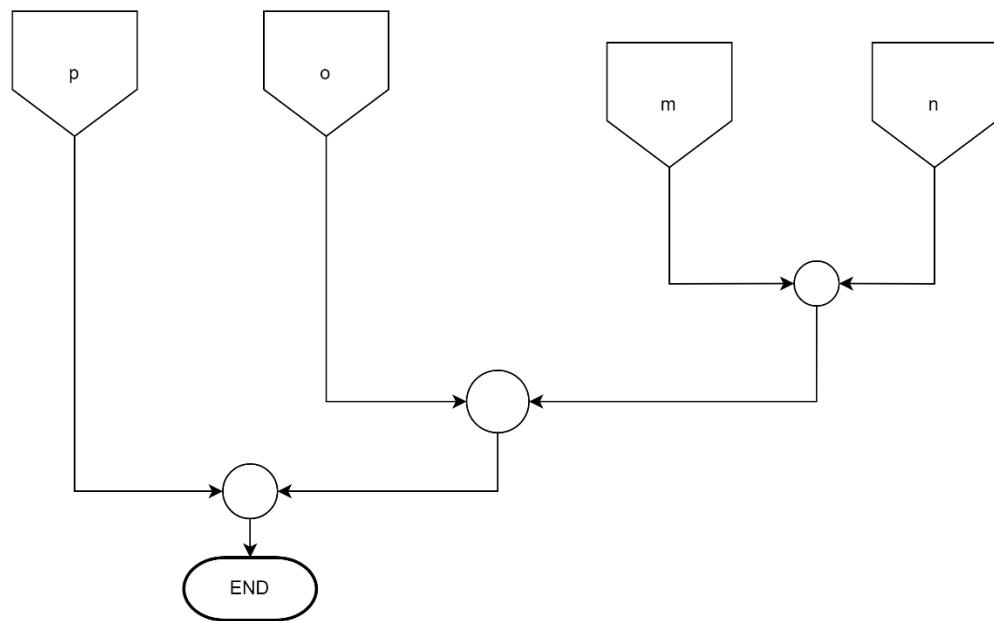
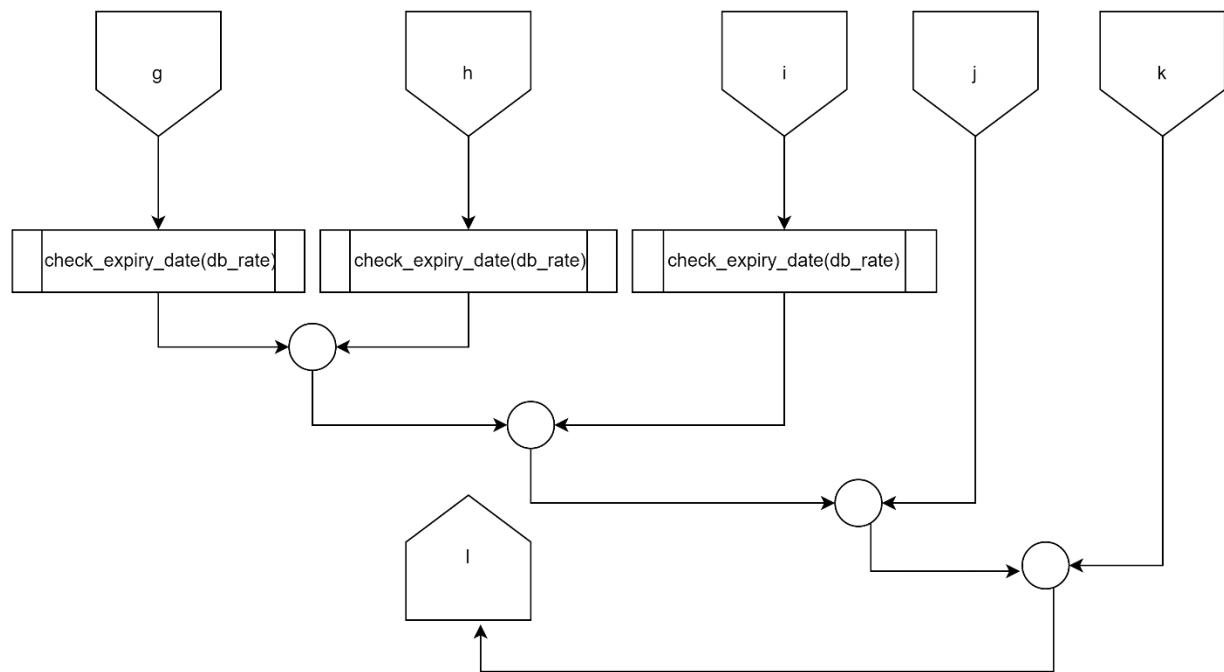
```
def check_expiry_date(db_rdate):
```

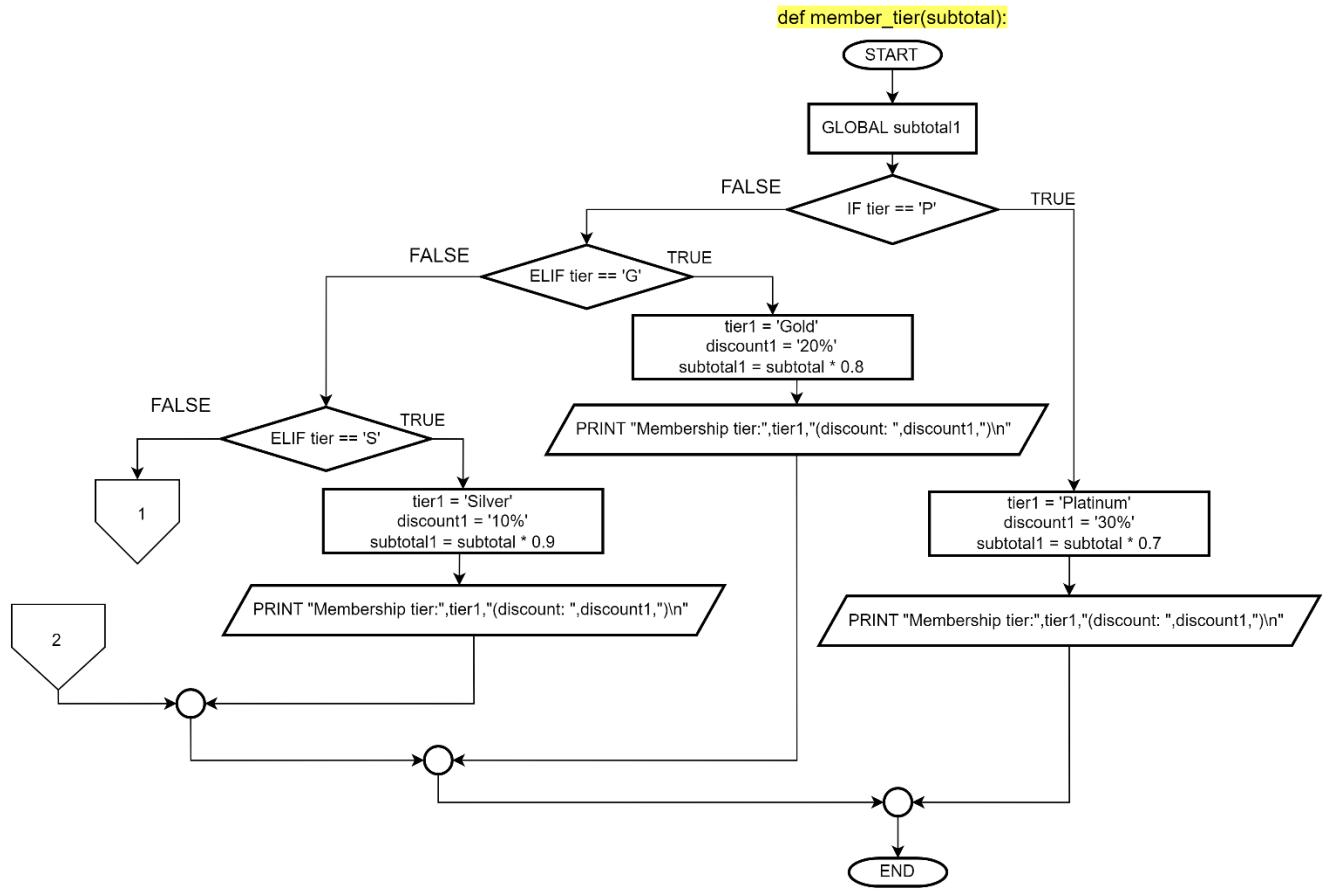


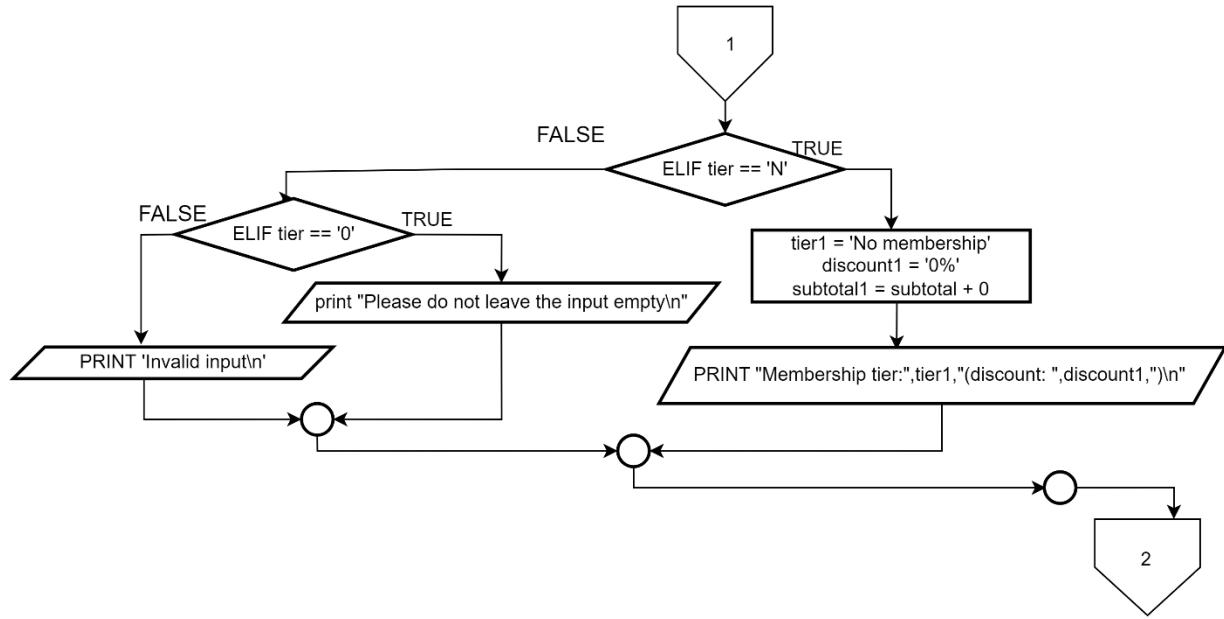


```
def check_memberstatus(linerecord):
```

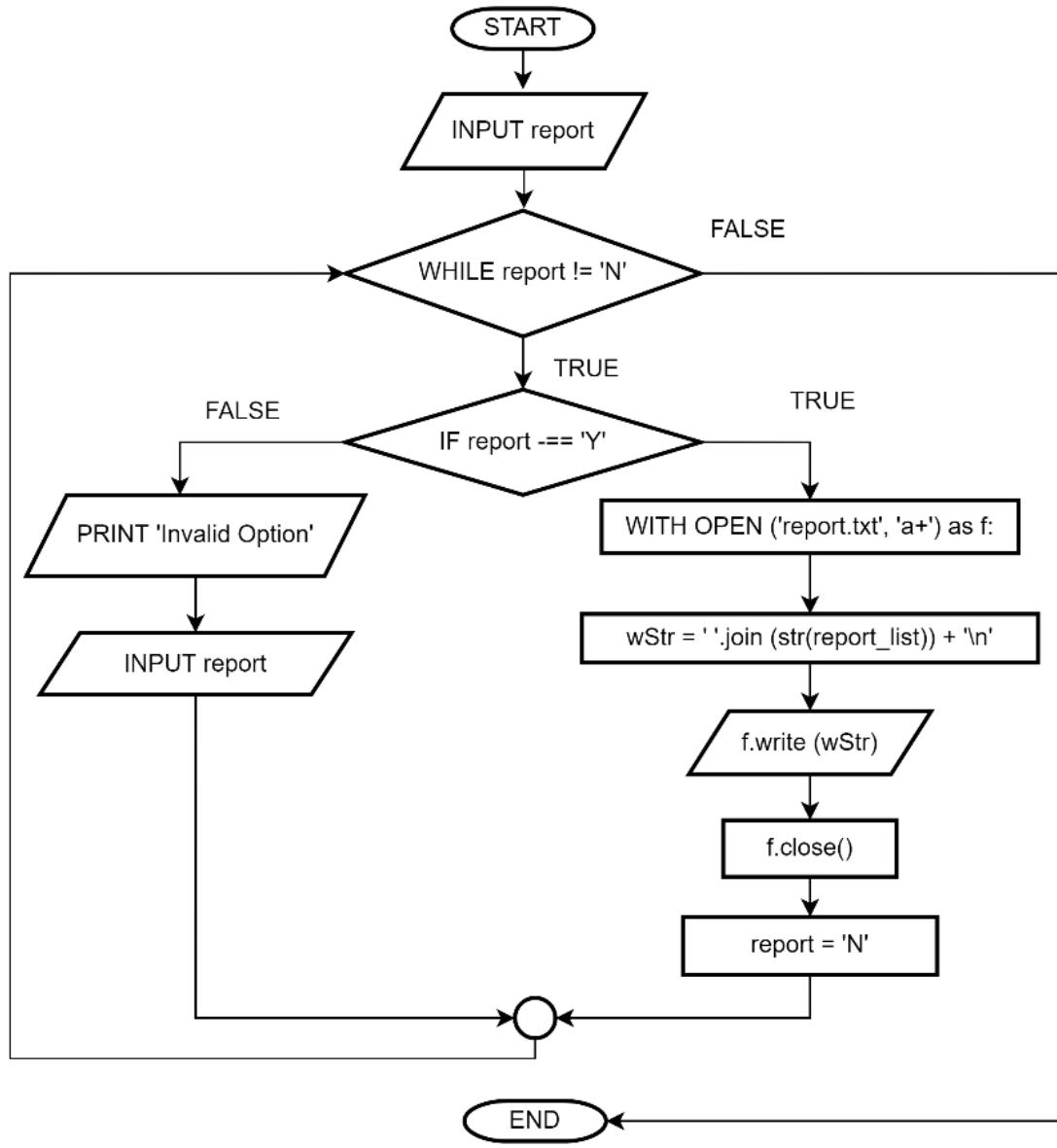


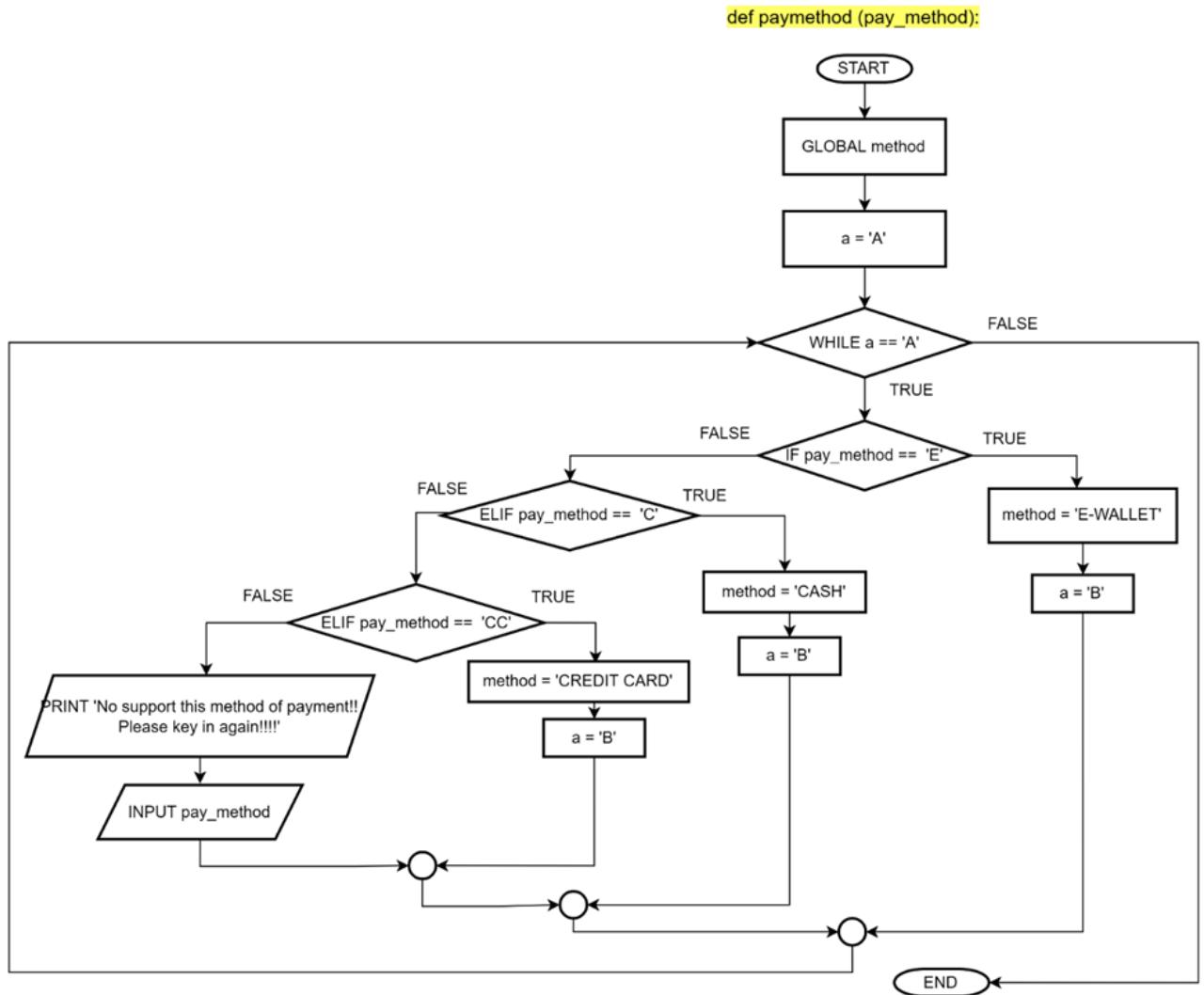




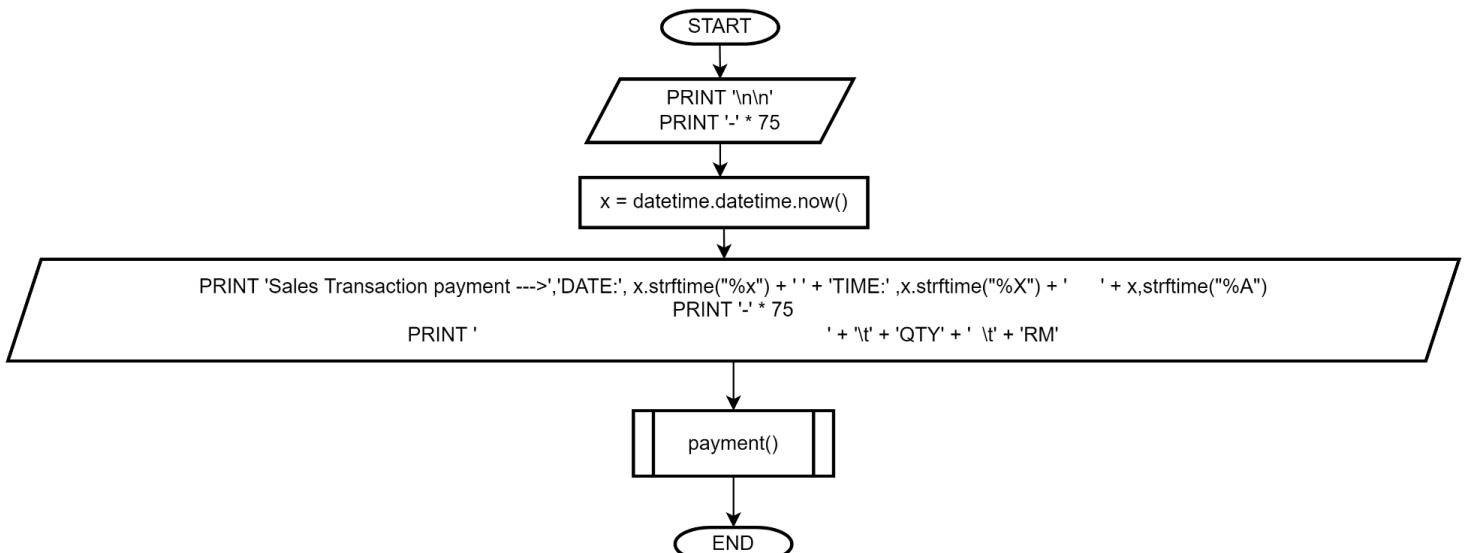


```
def report (report_list):
```

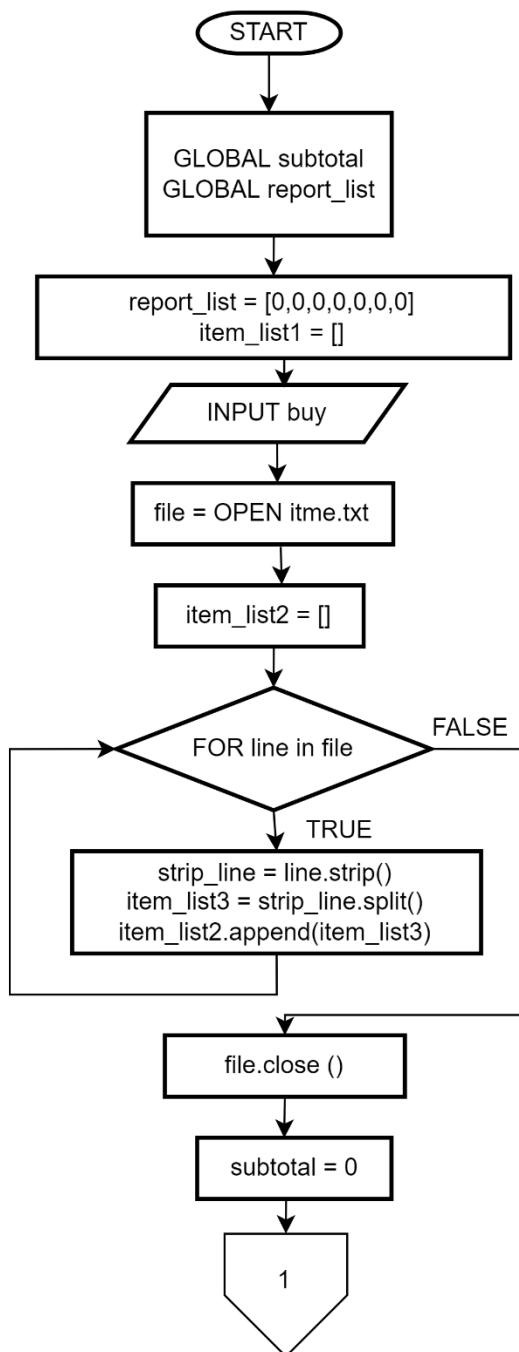


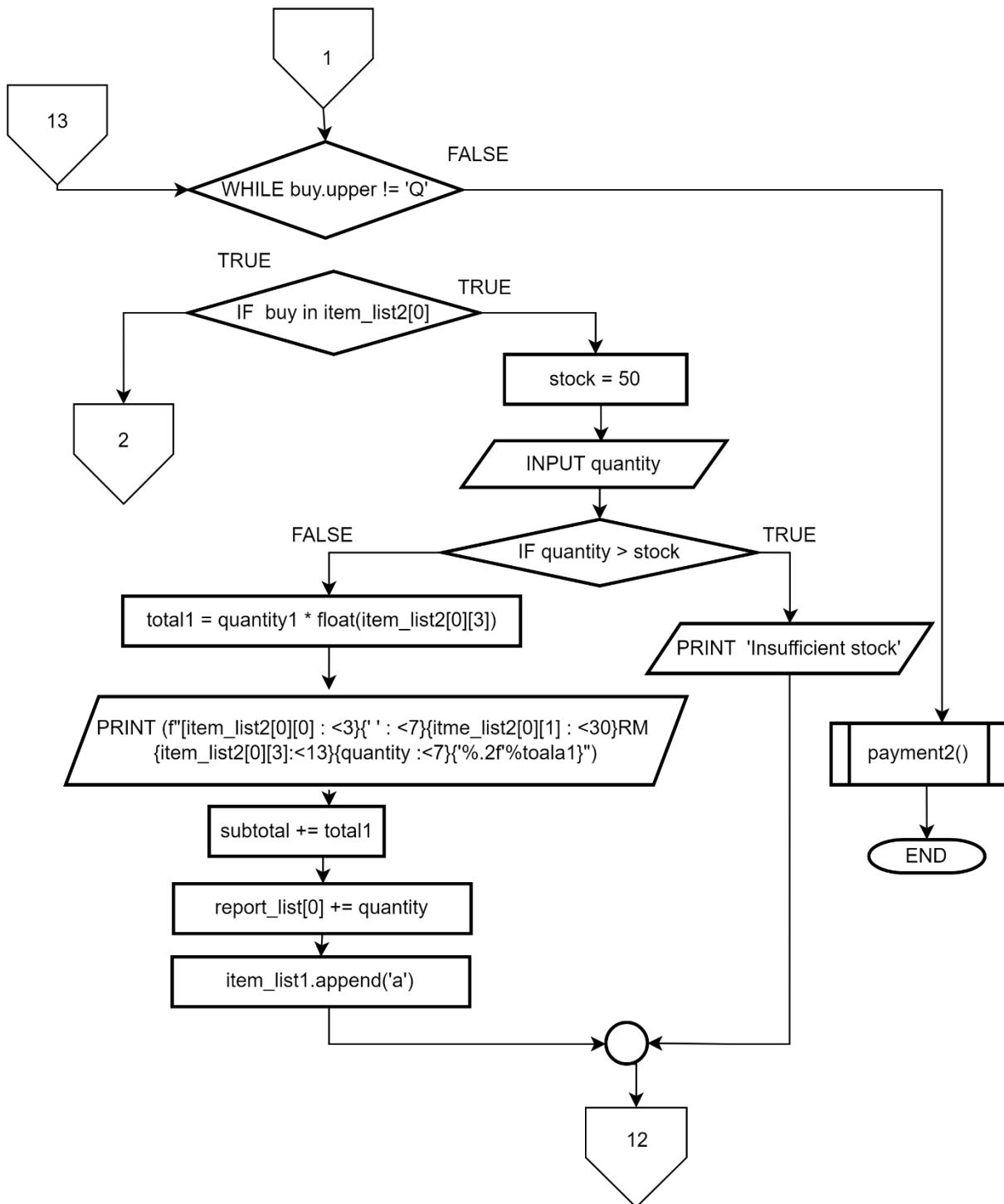


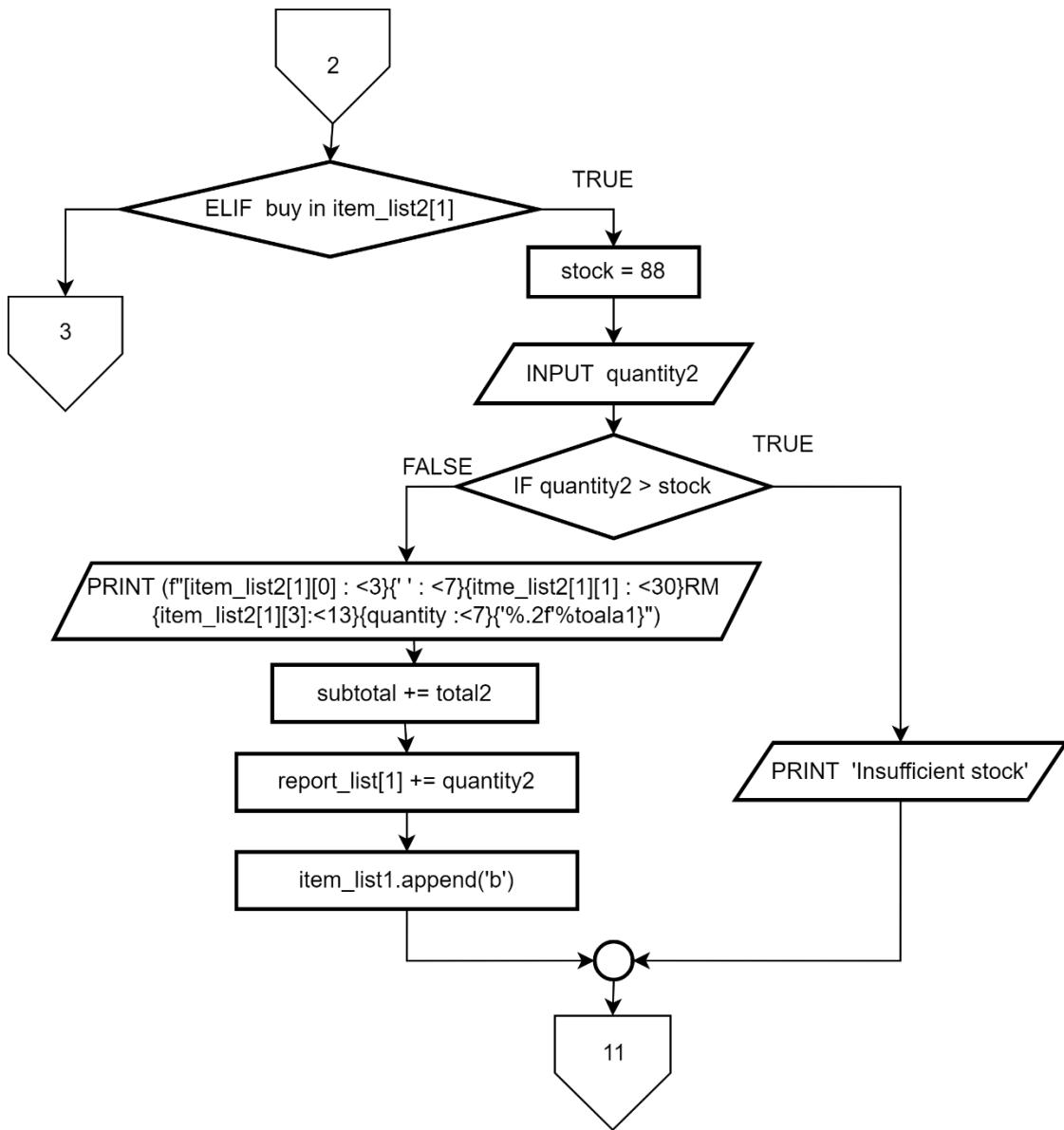
`def choice4():`

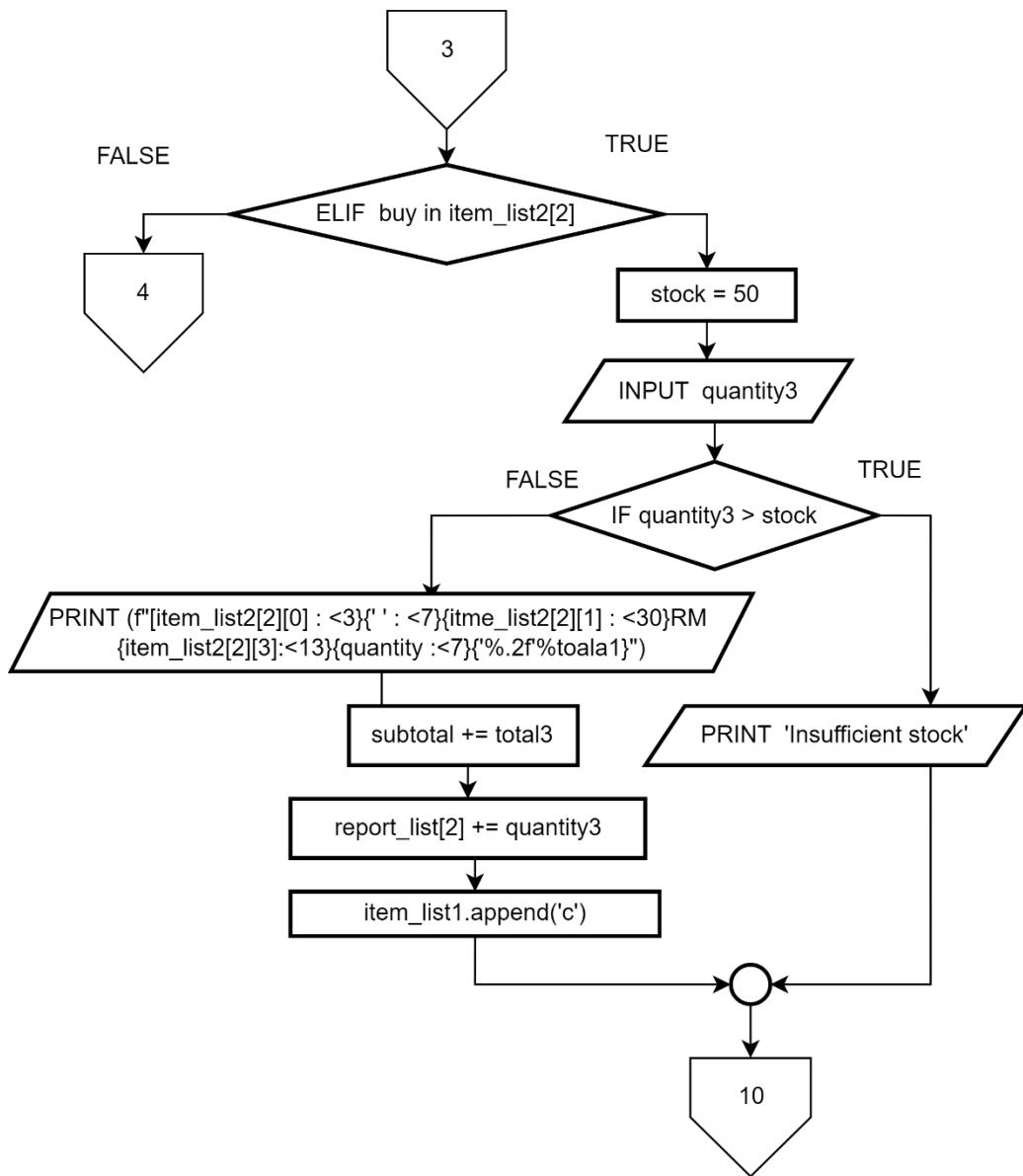


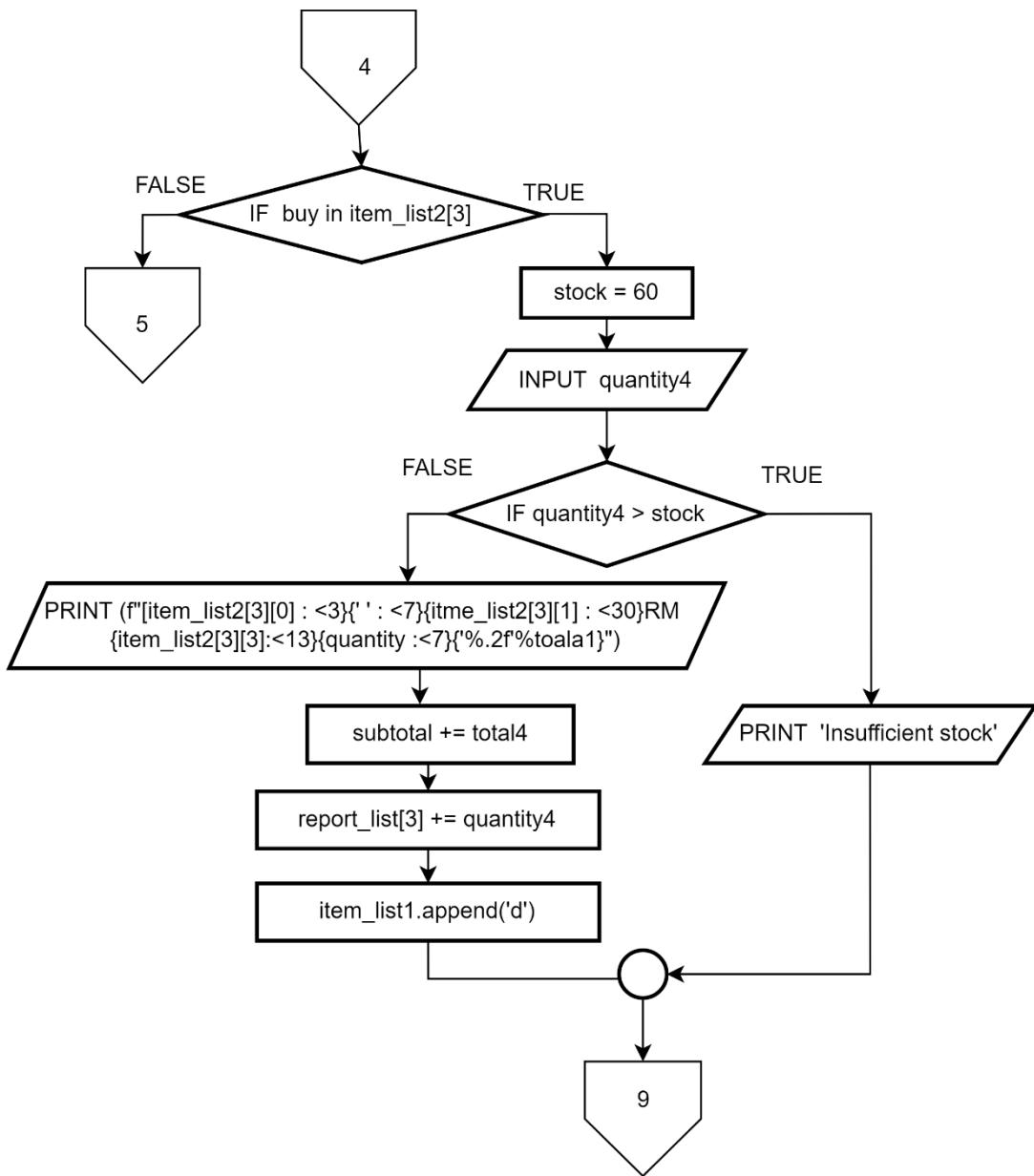
```
def payment():
```



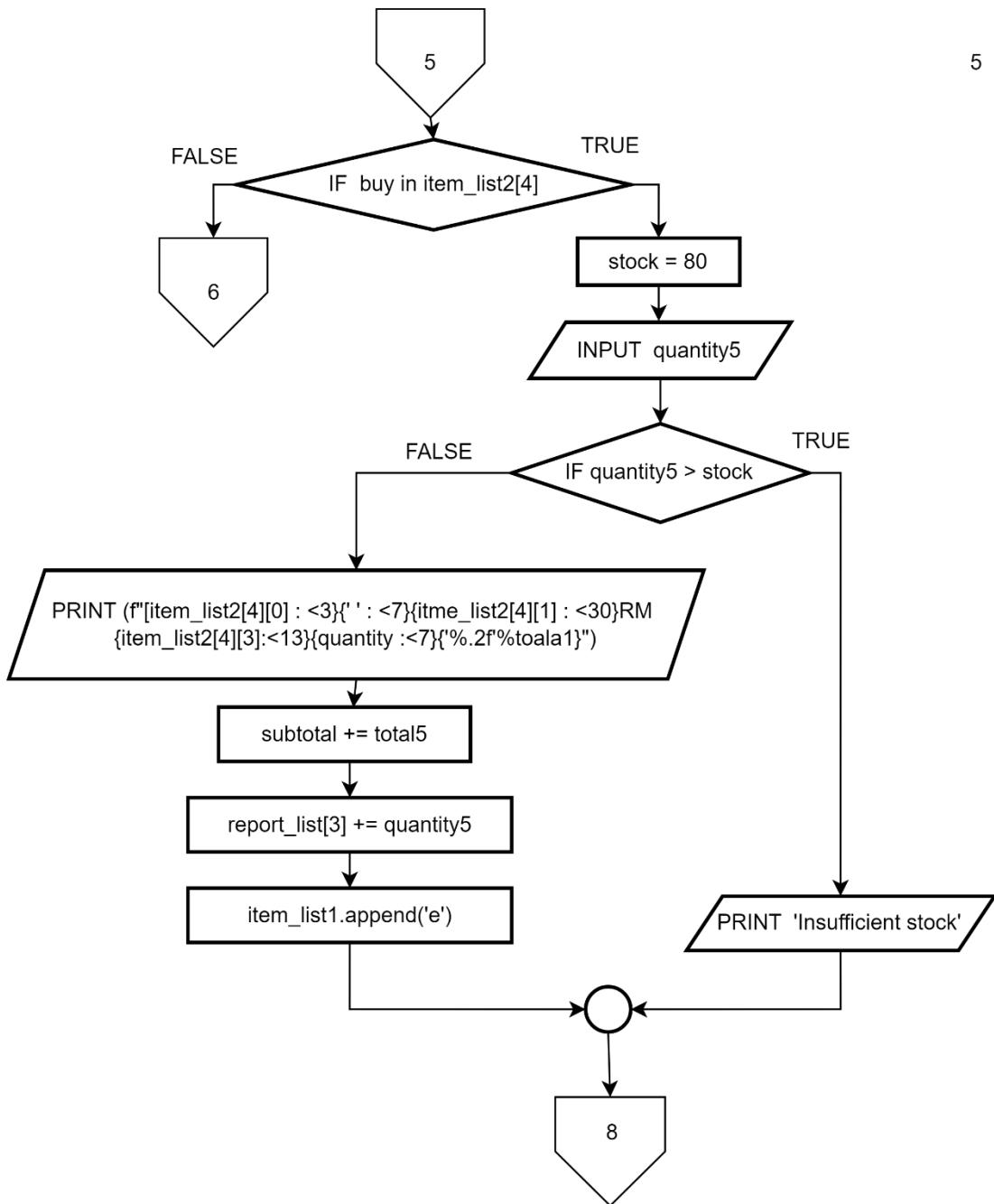


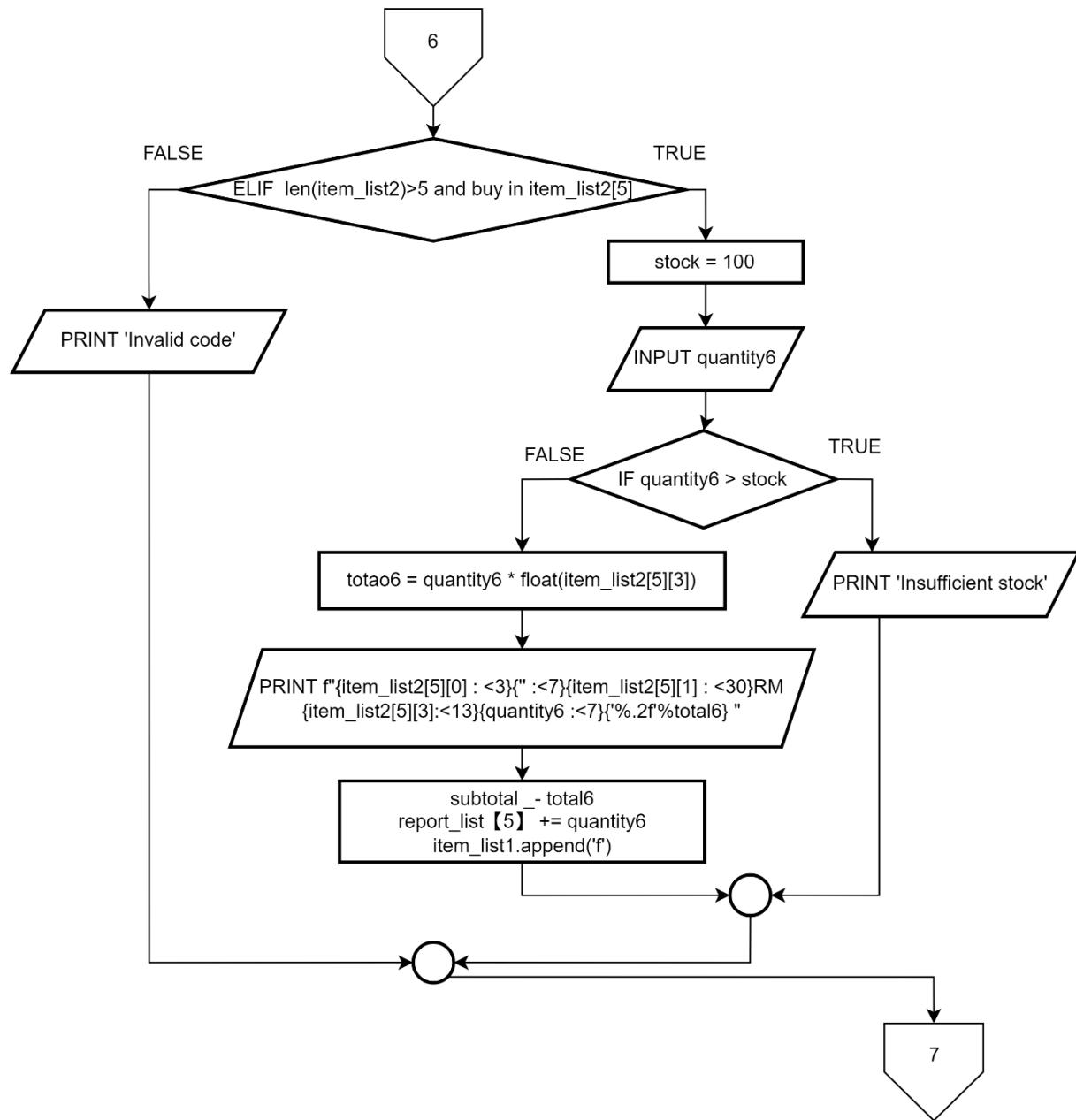


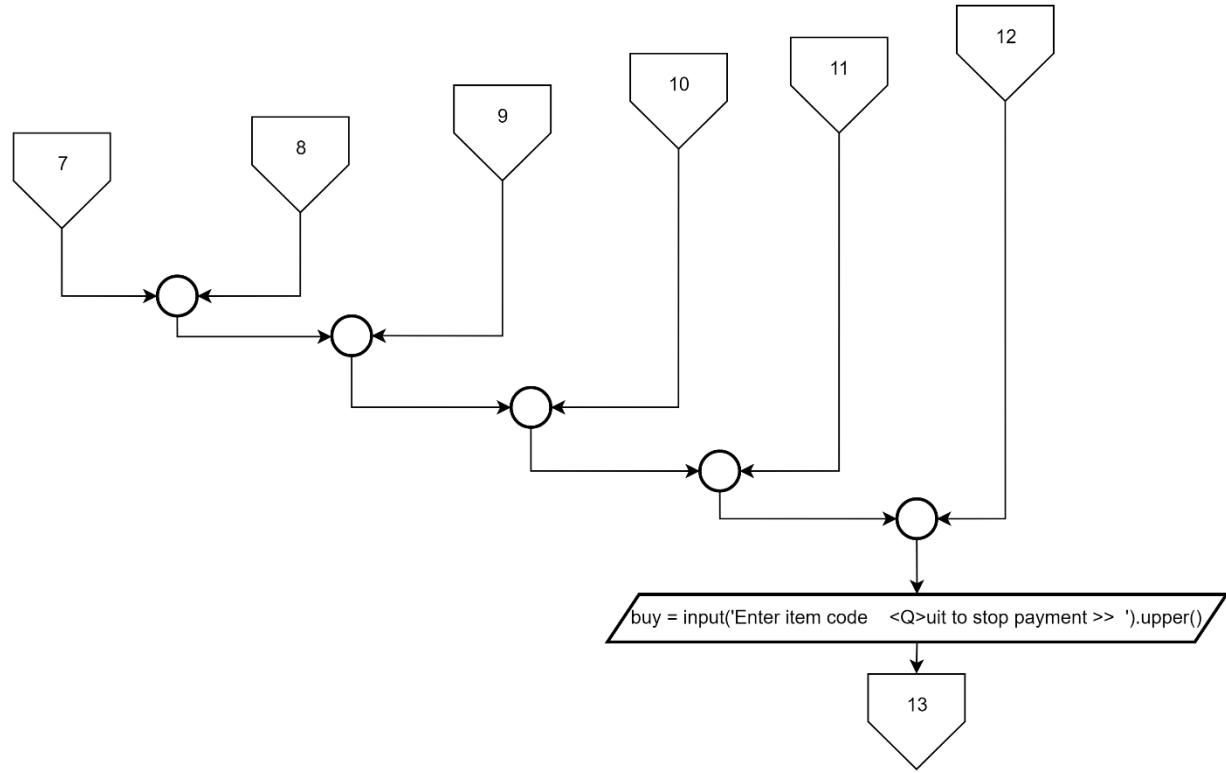




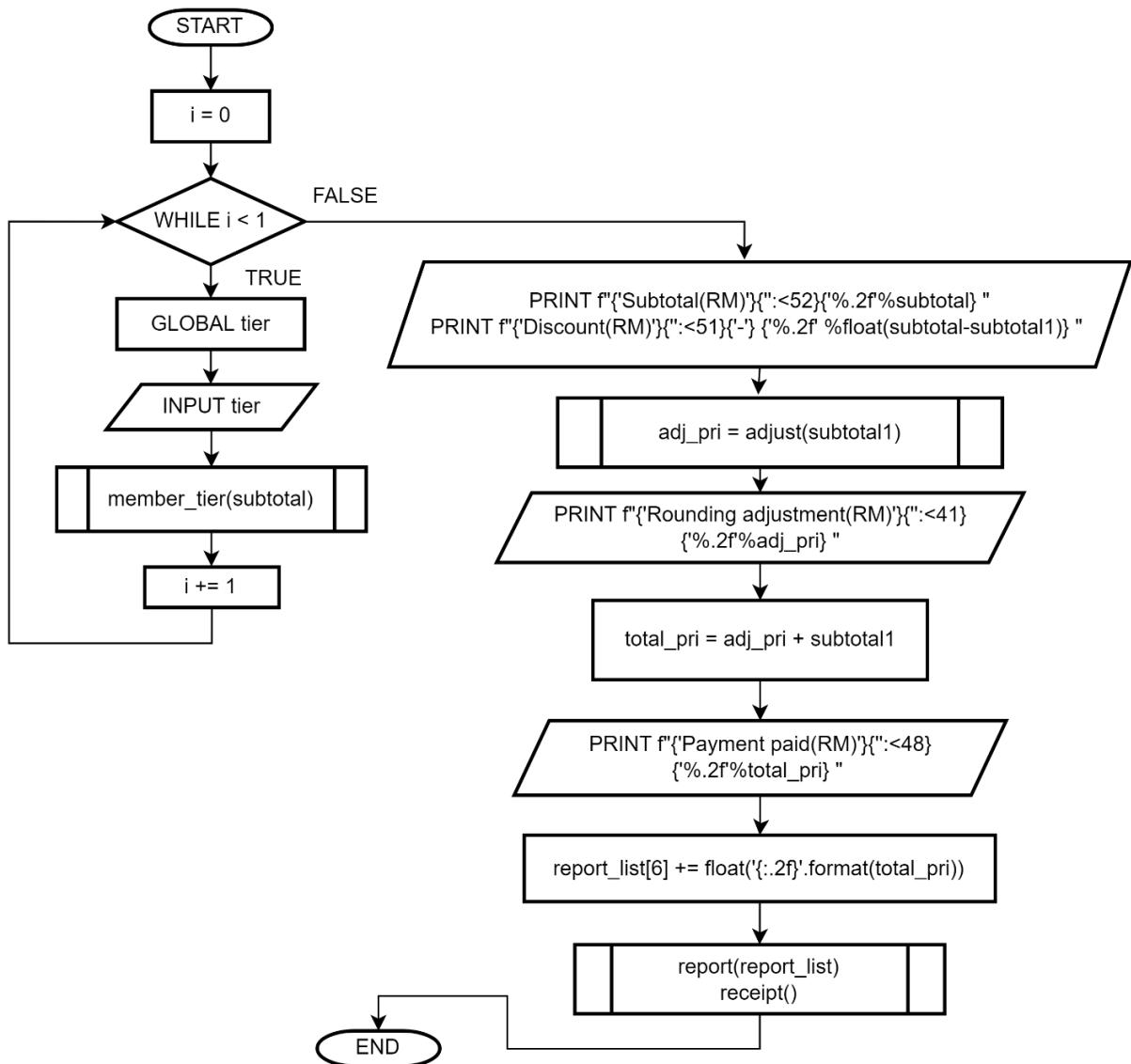
5



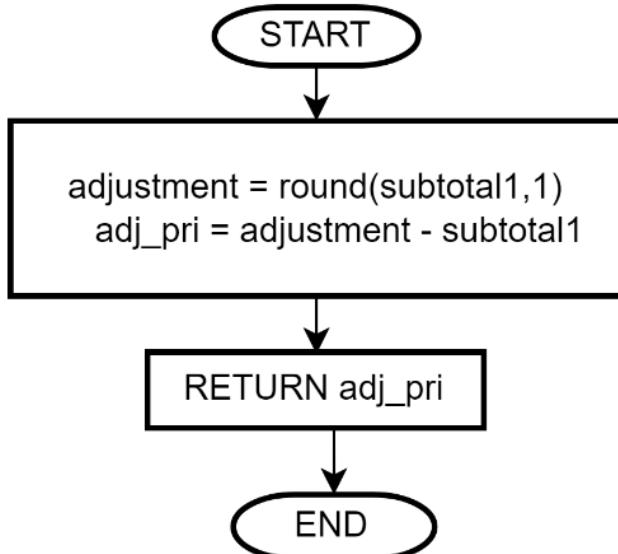




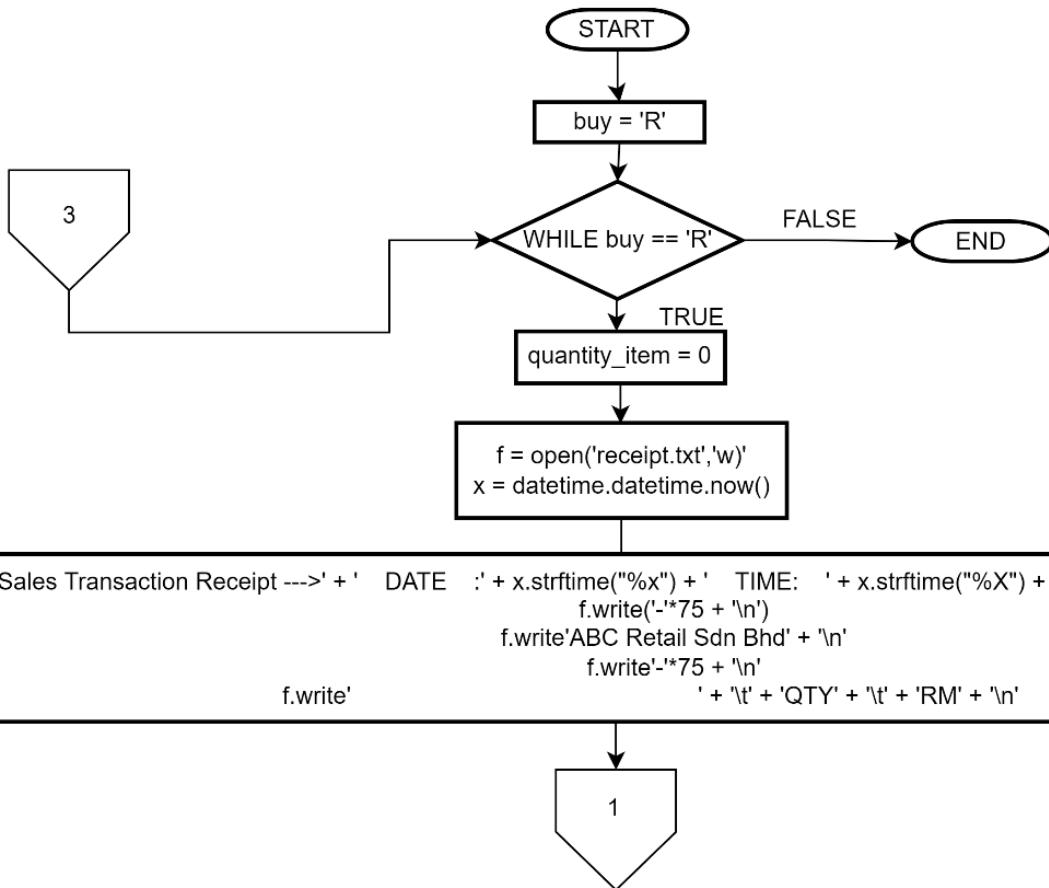
```
def payment2():
```

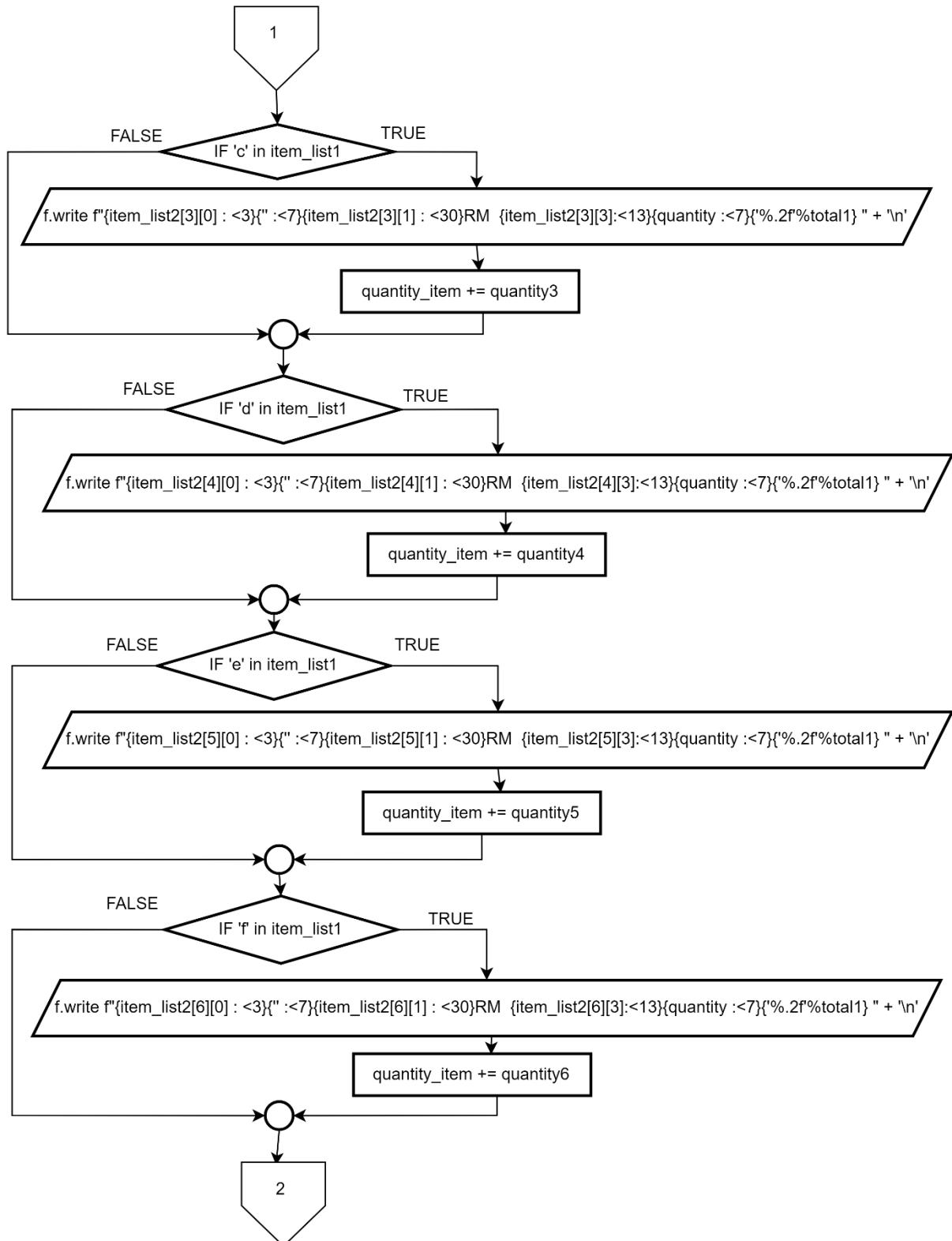


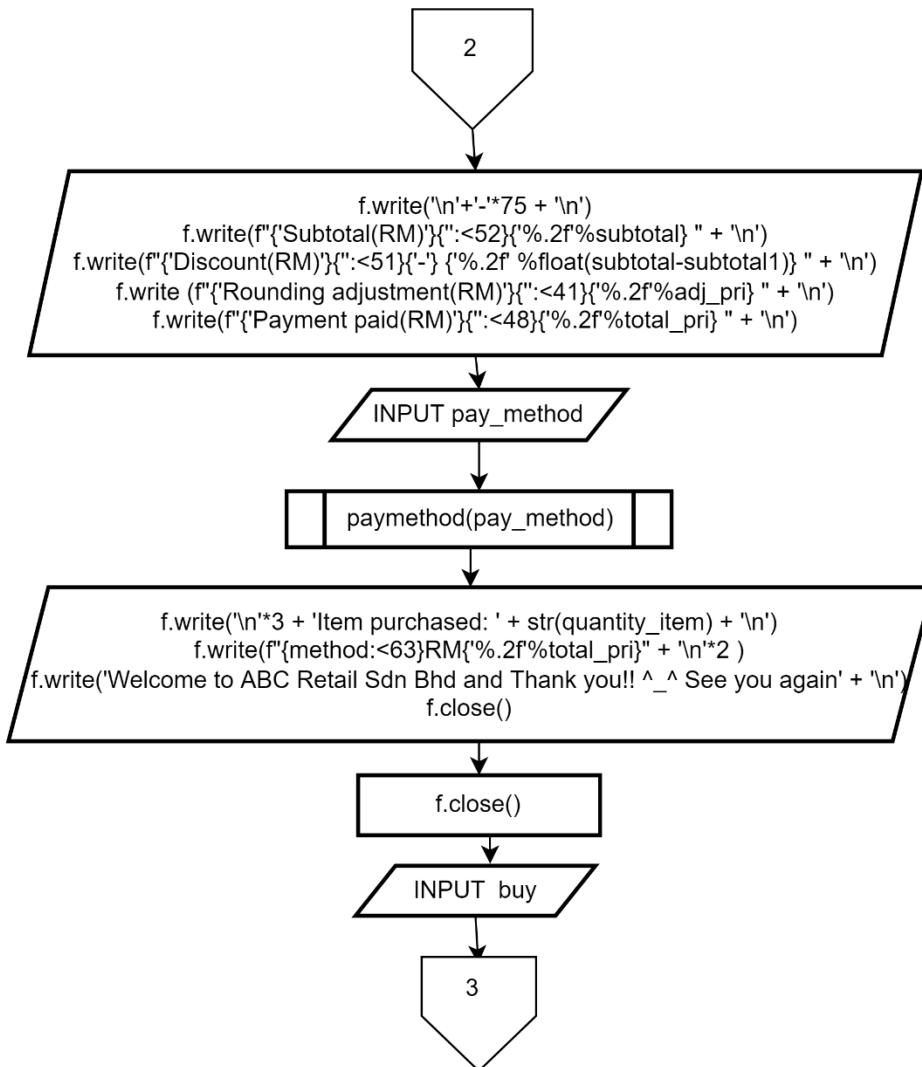
```
def adjust(subtotal1):
```

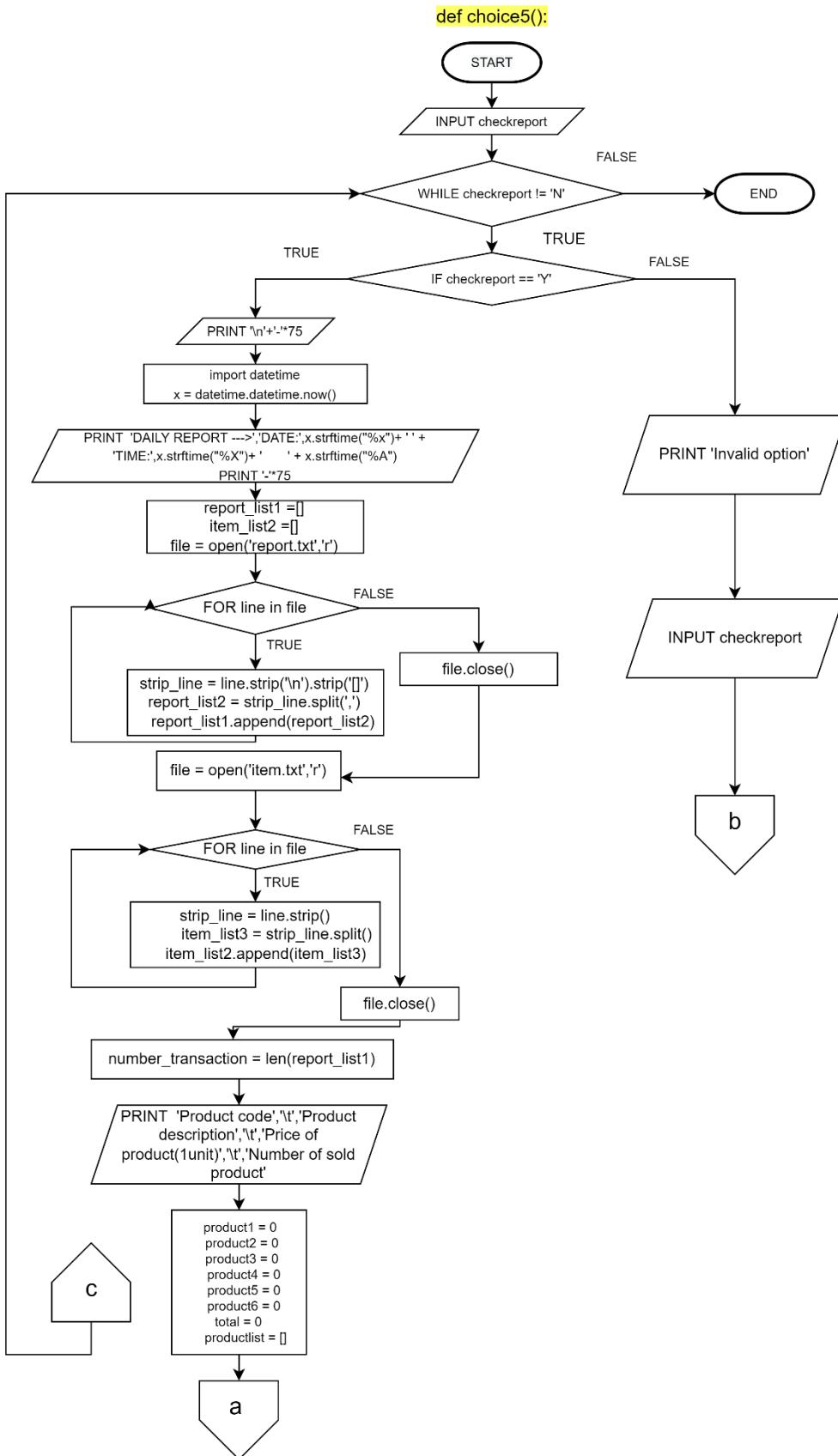


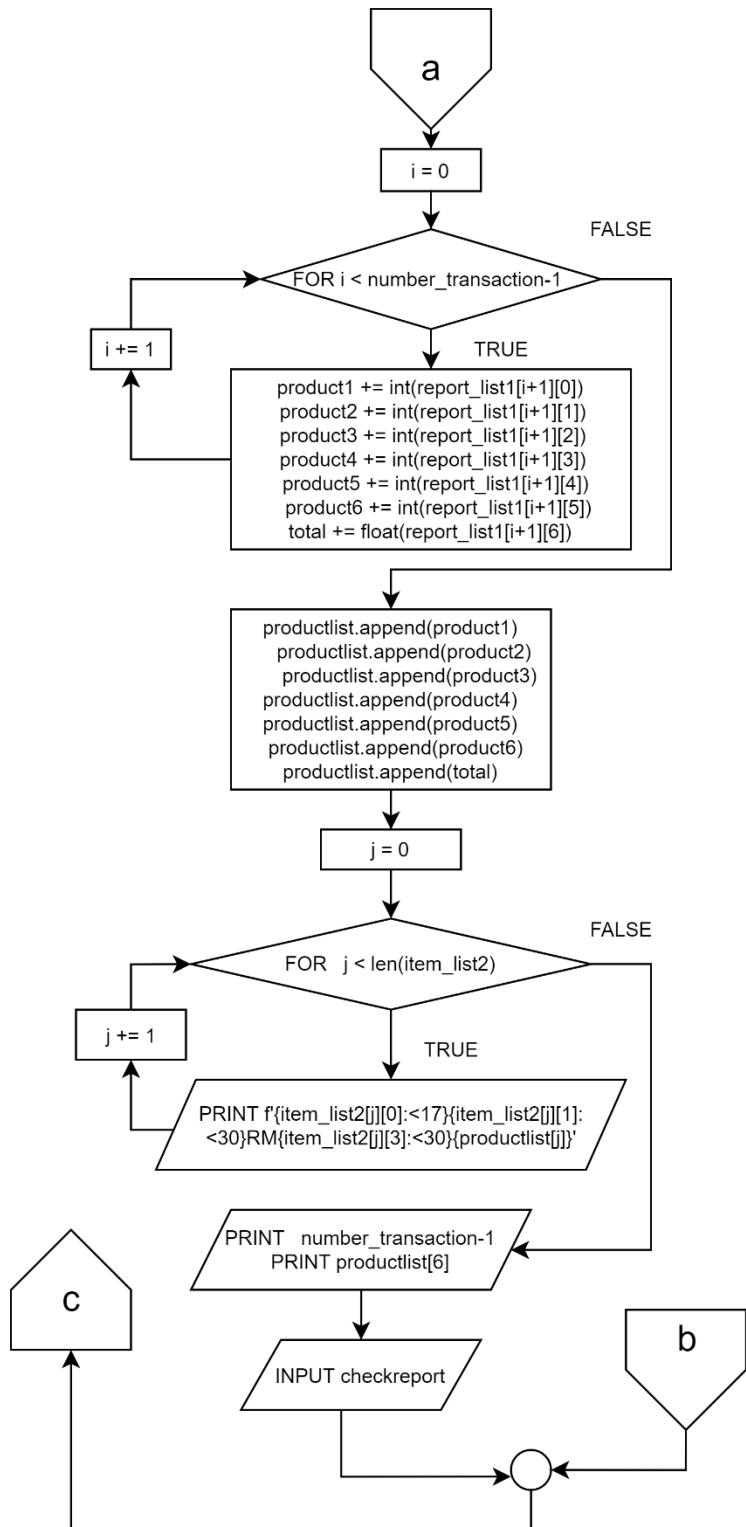
```
def receipt():
```







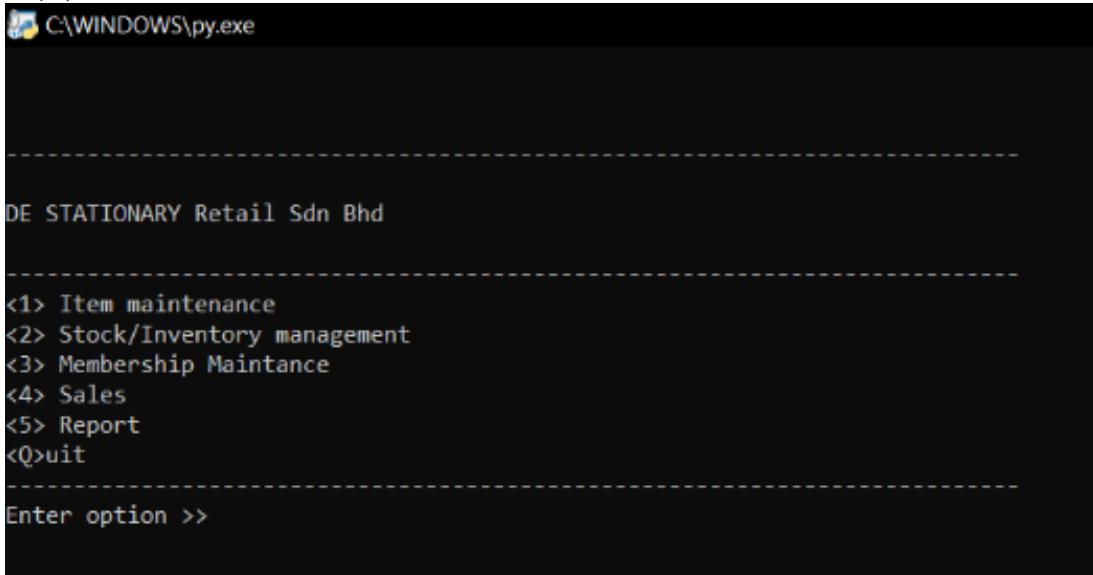




3. Screenshots

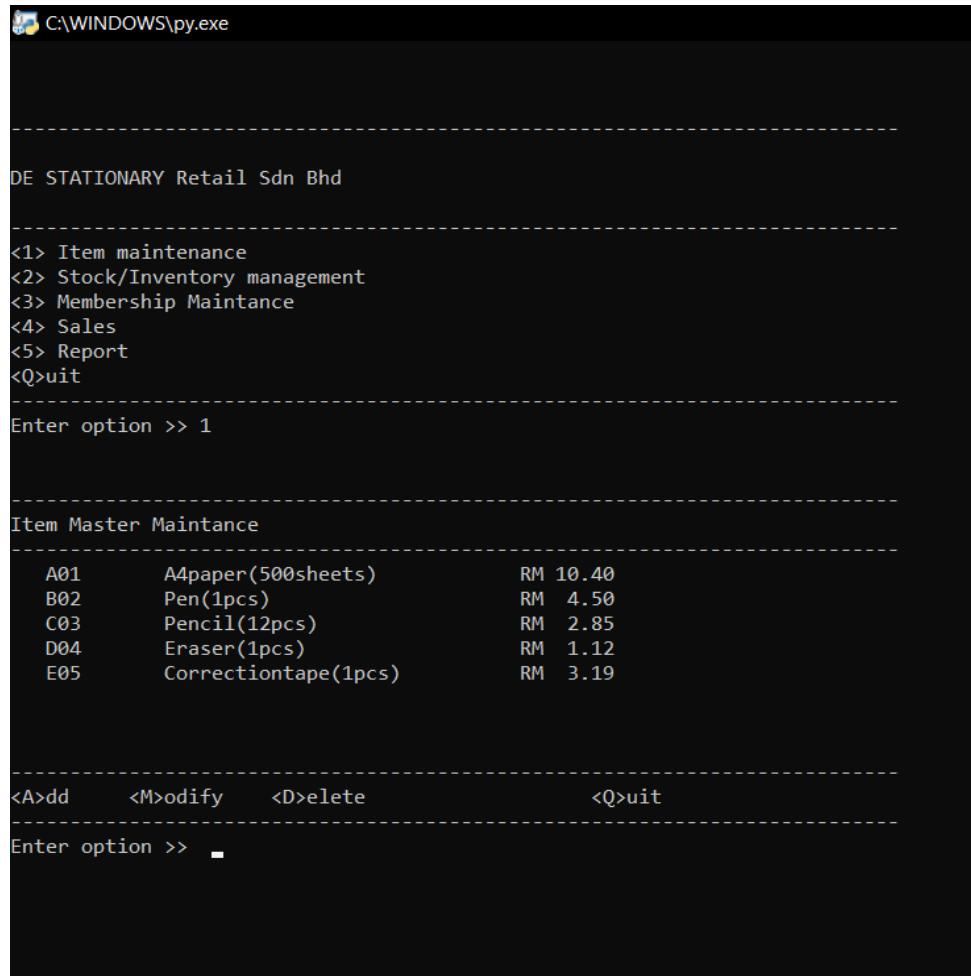
(Screenshot the program, just like the examples in the guideline.)

(A) Main Menu



```
C:\WINDOWS\py.exe
-----
DE STATIONARY Retail Sdn Bhd
-----
<1> Item maintenance
<2> Stock/Inventory management
<3> Membership Maintance
<4> Sales
<5> Report
<Q>uit
-----
Enter option >>
```

(B) Item Maintenance



```
C:\WINDOWS\py.exe
-----
DE STATIONARY Retail Sdn Bhd
-----
<1> Item maintenance
<2> Stock/Inventory management
<3> Membership Maintance
<4> Sales
<5> Report
<Q>uit
-----
Enter option >> 1

-----
Item Master Maintance
-----
A01      A4paper(500sheets)      RM 10.40
B02      Pen(1pcs)              RM  4.50
C03      Pencil(12pcs)          RM  2.85
D04      Eraser(1pcs)           RM  1.12
E05      Correctiontape(1pcs)   RM  3.19

-----
<A>dd      <M>odify      <D>elete      <Q>uit
-----
Enter option >> -
```

(B) (i) Adding New Item

```
C:\WINDOWS\py.exe

-----
<1> Item maintenance
<2> Stock/Inventory management
<3> Membership Maintance
<4> Sales
<5> Report
<Q>uit

Enter option >> 1

-----
Item Master Maintance

A01      A4paper(500sheets)      RM 10.40
B02      Pen(1pcs)              RM  4.50
C03      Pencil(12pcs)          RM  2.85
D04      Eraser(1pcs)           RM  1.12
E05      Correctiontape(1pcs)   RM  3.19

-----
<A>dd      <M>odify      <D>elete      <Q>uit

Enter option >> a

Enter item code >> F06
Enter item description  >>  Ruler(1pcs)
Enter item price    >> RM 3.2
```

(B) (ii) New Item Added

```
-----
Item Master Maintance

A01      A4paper(500sheets)      RM 10.40
B02      Pen(1pcs)              RM  4.50
C03      Pencil(12pcs)          RM  2.85
D04      Eraser(1pcs)           RM  1.12
E05      Correctiontape(1pcs)   RM  3.19
F06      Ruler(1pcs)            RM  3.20

-----
<A>dd      <M>odify      <D>elete      <Q>uit

Enter option >> ■
```

(B) (iii) Deleting Item

```
C:\WINDOWS\py.exe

Item Master Maintance

A01      A4paper(500sheets)      RM 10.40
B02      Pen(1pcs)              RM  4.50
C03      Pencil(12pcs)          RM  2.85
D04      Eraser(1pcs)           RM  1.12
E05      Correctiontape(1pcs)   RM  3.19
F07      Highlightpen(5pcs)     RM 10.40

-----  
<A>dd    <M>odify    <D>elete        <Q>uit  
Enter option >> d

Enter item code that want to delete >> F07
```

(B) (iv) Item Deleted

```
C:\WINDOWS\py.exe

Item Master Maintance

A01      A4paper(500sheets)      RM 10.40
B02      Pen(1pcs)              RM  4.50
C03      Pencil(12pcs)          RM  2.85
D04      Eraser(1pcs)           RM  1.12
E05      Correctiontape(1pcs)   RM  3.19
F07      Highlightpen(5pcs)     RM 10.40

-----  
<A>dd    <M>odify    <D>elete        <Q>uit  
Enter option >> d

Enter item code that want to delete >> F07

-----  
Item Master Maintance

A01      A4paper(500sheets)      RM 10.40
B02      Pen(1pcs)              RM  4.50
C03      Pencil(12pcs)          RM  2.85
D04      Eraser(1pcs)           RM  1.12
E05      Correctiontape(1pcs)   RM  3.19

-----  
<A>dd    <M>odify    <D>elete        <Q>uit  
Enter option >> q
```

(B) (v) Modifying Item

```
Item Master Maintance
-----
A01      A4paper(500sheets)      RM 10.40
B02      Pen(1pcs)              RM  4.50
C03      Pencil(12pcs)          RM  2.85
D04      Eraser(1pcs)           RM  1.12
E05      Correctiontape(1pcs)   RM  3.19
F06      Ruler(1pcs)            RM  3.20

-----
<A>dd    <M>odify    <D>elete    <Q>uit
-----
Enter option >> m

Enter original item code that want to modify >> F06
modified item code(new) >>   F07
modified item description(new) >>  Highlightpen(5pcs)
modified item price(new) >> RM 10.4
```

(B) (vi) Item Modified

```
Item Master Maintance
-----
A01      A4paper(500sheets)      RM 10.40
B02      Pen(1pcs)              RM  4.50
C03      Pencil(12pcs)          RM  2.85
D04      Eraser(1pcs)           RM  1.12
E05      Correctiontape(1pcs)   RM  3.19
F07      Highlightpen(5pcs)      RM  10.40

-----
<A>dd    <M>odify    <D>elete    <Q>uit
-----
Enter option >>
```

(C) Inventory Shortfall

C:\WINDOWS\py.exe

```
Enter item Code <Q>uit >>C03
```

```
C03 Pencil(12pcs) RM 2.85
```

```
Current inventory >> 50
```

```
Enter quantity >>69
```

```
Insuffiecient stock
```

```
Remaining stocks: -19
```

```
Enter item Code <Q>uit >>D04
```

```
D04 Eraser(1pcs) RM 1.12
```

```
Current inventory >> 60
```

```
Enter quantity >>55
```

```
Sufficient stock
```

```
Remaining stocks: 5
```

```
Enter item Code <Q>uit >>E05
```

```
E05 Correctiontape(1pcs) RM 3.19
```

```
Current inventory >> 80
```

```
Enter quantity >>54
```

```
Sufficient stock
```

```
Remaining stocks: 26
```

```
Enter item Code <Q>uit >>q
```

(D) Membership Maintenance

```
C:\WINDOWS\py.exe

-----
DE STATIONARY Retail Sdn Bhd

-----
<1> Item maintenance
<2> Stock/Inventory management
<3> Membership Maintance
<4> Sales
<5> Report
<Q>quit
-----
Enter option >> 3

-----
Membership info --> Date: 2022-04-15 Time: 07 : 36 : 34
-----
<1> Introduction of membership
<2> Membership Registration
<3> Check membership status
<Q> Quit
Enter option >> -
```

(D) (i) Introduction of Membership

```
Membership info --> Date: 2022-04-14 Time: 23 : 53 : 52
-----
<1> Introduction of membership
<2> Membership Registration
<3> Check membership status
<Q> Quit
Enter option >> 1
Please input Membership tier (<P>latinum <G>old <S>ilver <Q>uit): p
Membership tier: Platinum
Discount: 30%
Every month pay Rm30
Get one year membership

Please input Membership tier (<P>latinum <G>old <S>ilver <Q>uit): g
Membership tier: Gold
Discount: 20%
Every month pay Rm20
Get one year membership

Please input Membership tier (<P>latinum <G>old <S>ilver <Q>uit): s
Membership tier: Silver
Discount: 10%
Every month pay Rm10
Get one year membership
```

(D) (ii) Membership Registration

```
Membership info --> Date: 2022-04-14 Time: 23 : 54 : 57
-----
<1> Introduction of membership
<2> Membership Registration
<3> Check membership status
<Q> Quit
Enter option >> 2

Membership registration fee = RM10
Please provide information below to complete registration
Please insert full name: KEHYIQIAN
Please input IC number: 030106040105
Please select date (eg.dd/mm/yyyy): 14/04/2022
Please select the member tier that you want (<P>latinum <G>old <S>ilver) >> p
Membership tier: Platinum
Discount: 30%
Every month pay Rm30
Get one year membership

Name           : KEHYIQIAN
IC number      : 030106040105
Registration date : 14/04/2022
Member tier    : Platinum
Signup for Membership?  <Y>es <N>o : y

Your membership has been registered.
Congratulation!! Now you are the Platinum member
Your membership id is P01052022
```

(D) (iii) Checking Membership Status

```
Membership info --> Date: 2022-04-14 Time: 23 : 55 : 38
-----
<1> Introduction of membership
<2> Membership Registration
<3> Check membership status
<Q> Quit
Enter option >> 3
Please insert membership id  <Q>uit : P01052022
You are the platinum member.
Your membership card is still valid.
Please insert membership id  <Q>uit : q
```

(E) Adding Item for Payment

```
Sales Transaction payment ---> DATE: 04/14/22 TIME: 23:56:33 Thursday
-----
Enter item code <Q>uit to stop payment>> A01
Enter quantity >> 23
A01      A4paper(500sheets)           RM 10.40    23    239.20
Enter item code <Q>uit to stop payment >> B02
Enter quantity >> 33
B02      Pen(1pcs)                 RM 4.50     33    148.50
Enter item code <Q>uit to stop payment >> D04
Enter quantity >> 3
D04      Eraser(1pcs)              RM 1.12      3     3.36
Enter item code <Q>uit to stop payment >> E05
Enter quantity >> 12
E05      Correctiontape(1pcs)       RM 3.19     12    38.28
Enter item code <Q>uit to stop payment >> q
```

(F) Payment

```
Please input Membership tier (<P>latinum <G>old <S>ilver <N>o): p
Membership tier: Platinum (discount: 30% )

Subtotal(RM)                                429.34
Discount(RM)                                 - 128.80
Rounding adjustment(RM)                      -0.04
Payment paid(RM)                             300.50
Do you record this transaction? <Y>es <N>o >> y
Enter option <R>eceipt key in any to exit>> r
Paying method (<E>-wallet <C>ash <C>redit<C>ard) >> c
key in any to exit >>
```

(G)Receipt

Sales Transaction Receipt ---> DATE :04/14/22 TIME: 23:57:22 Thursday

DE STATIONARY Retail Sdn Bhd

		QTY	RM
A01	A4paper(500sheets)	RM 10.40	23 239.20
B02	Pen(1pcs)	RM 4.50	33 148.50
D04	Eraser(1pcs)	RM 1.12	3 3.36
E05	Correctiontape(1pcs)	RM 3.19	12 38.28

Subtotal(RM)	429.34
Discount(RM)	- 128.80
Rounding adjustment(RM)	-0.04
Payment paid(RM)	300.50

Item purchased: 71
CASH RM300.50

Welcome to ABC Retail Sdn Bhd and Thank you!! ^_^ See you again

(H) Report

```
C:\WINDOWS\py.exe

DE STATIONARY Retail Sdn Bhd

-----  
<1> Item maintenance  
<2> Stock/Inventory management  
<3> Membership Maintance  
<4> Sales  
<5> Report  
<Q>uit

Enter option >> 5
Are you want to check the daily report? <Y>es <N>o >> y

DAILY REPORT ----> DATE: 04/15/22 TIME: 09:25:59      Friday
Product code      Product description      Price of product(1unit)      Number of sold product
-----  
A01      A4paper(500sheets)      RM10.40      23  
B02      Pen(1pcs)      RM4.50      33  
C03      Pencil(12pcs)      RM2.85      0  
D04      Eraser(1pcs)      RM1.12      3  
E05      Correctiontape(1pcs)      RM3.19      12

Number of transaction today is 1 times.  
Total price of transaction today is RM300.50  
key in <N> to exit <Y> to continue>> n

-----  
DE STATIONARY Retail Sdn Bhd

-----  
<1> Item maintenance  
<2> Stock/Inventory management  
<3> Membership Maintance  
<4> Sales  
<5> Report  
<Q>uit

Enter option >> -
```

(I) Data Validation

```
C:\WINDOWS\py.exe

-----  
DE STATIONARY Retail Sdn Bhd

-----  
<1> Item maintenance  
<2> Stock/Inventory management  
<3> Membership Maintance  
<4> Sales  
<5> Report  
<Q>uit

Enter option >> 6
Invalid option

-----  
Enter option >> a
Invalid option

-----  
Enter option >>
```

4. Source-code (program code)

Point Of Sale System.py

```
def pointofsale_menu():
    print('\n\n')
    print ('-'*75)
    print ("nDE STATIONARY Retail Sdn Bhd\n")
    print ('-'*75)
    print ('<1> Item maintenance')
    print ('<2> Stock/Inventory management')
    print ('<3> Membership Maintenance')
    print ('<4> Sales')
    print ('<5> Report')
    print ('<Q>uit')
    print ('-'*75)

pointofsale_menu() # user's option
choice_no = input('Enter option >> ')

while choice_no.upper() !='Q': # key in Q to exit the point of sale system
# itemmaintenance part
    if choice_no =='1':

        def print1():
            print ("\n")
            print ('-'*75)
            print ('Item Master Maintance')
            print ('-'*75)
            file = open ('item.txt' , 'r') # open item.txt to read content of its
            print(file.read())
            file.close()

        def print2():
            print('-'*75)
            print ('<A>dd    <M>odify    <D>elete    '           '<Q>uit')
            print('-'*75)

        print1()
        print('\n')
        print2()

    def choice1():

        #selling item just to introduce
        a = ' A01    A4paper(500sheets)      RM 10.40'
        b = ' B02    Pen(3pcs)              RM 4.50'
        c = ' C03    Pencil(12pcs)          RM 2.85'
        d = ' D04    Eraser(1pcs)           RM 1.12'
        e = ' E05    Correctiontape(1pcs)   RM 3.19'
        f = ' F06    Ruler(1pcs)            RM 5.20 #example of item for adding
                                         part,modifying part


```

```

# user choice 'A' to add the item,'M' to modify item,'D' to delete item & 'Q' to exit
option1 = input('Enter option >> ')
# suggest add one item only or number of item max 6 if needed add more item need
# to elongate the coding
# add part even though can add exceed 6 item but maybe affect the below part
while option1.upper() !='Q':
# because key in wrongly of item content can delete,so does not put while loop for quit
if option1.upper() == 'A': # add part
    print('\n')

    add = input('Enter item code >> ') # key in the item's code that want to add
    add1 = input ('Enter item description >> ') # key in the item's name that want to
# add
    add2 = float(input ('Enter item price >> RM ')) # key in the item's price that want to
# add, (must number)

    add_list = (f" {add :<3}{" :<7}{add1 :<30}RM {'%.2f'%add2}") # add the
# (add,add1,add2) above into add_list and adjust the space

    with open('item.txt' , 'a+')as f: # open item.txt append+read add_list into item.txt
        wstr= '\n' + ".join(add_list)
        f.write(wstr)

    print1()
    print('\n')
    print2()
    option1 = input('Enter option >> ')

elif option1.upper() =='M': #modify part
    print('\n')

    modify = input('Enter original item code that want to modify >> ') # key in the original item's code
# that want to modify
    item_modify = input('modified item code(new) >> ') # key in the new item's code that want to
# modify
    item_modify1 = input('modified item description(new) >> ') # key in the new item's name that want to
# modify
    item_modify2 = float(input ('modified item price(new) >> RM ')) # key in the new item's price that
# want to modify, (must number)

    item_modifylist =(f" {item_modify :<3}{" :<7}{item_modify1 :<30}RM
# add the (modify and other) above into
        '%.2f%{item_modify2}') # add the (modify and other) above into
# item_modifylist and adjust the space

    with open('item.txt' , 'r+')as f: #read+write item_modifylist into item.txt
        f_lines = f.readlines()
        f.seek(0)
        for line in f_lines: # if modify/original code not in item.txt, item.txt remains unchanged
            if modify not in line:
                f.write(line)

            else: # else change the original item to item that want to modify
                wstr = ".join(item_modifylist) + '\n'
                f.write(wstr)
        f.truncate()

```

```

print1()
print('\n')
print2()
option1 = input('Enter option >> ')

elif option1.upper() == 'D': #delete part
    print('\n')

    delete = input('Enter item code that want to delete >> ')

    with open ('item.txt' , 'r+') as f:
        f_lines= f.readlines()
        f.seek(0)
        for line in f_lines:
            if delete not in line: # if wanted deleted item code not in item.txt, item.txt remain unchanged
                f.write(line)

            else: # if wanted deleted item code in the item.txt, item code will be deleted
                wstr = ".join()"
                f.write(wstr)
        f.truncate()

    print1()
    print('\n')
    print2()
    option1 = input('Enter option >> ')

else: # option invalidation
    print('invalid option!!! Please key again')
    print1()
    print2()
    option1 = input('Enter option >> ')
    print('\n\n\n')

choice1()
pointofsale_menu()
choice_no = input('Enter option >> ')
print('\n\n')

#2 inventory management
elif choice_no == '2':
    def choice2():

        from datetime import datetime
        from datetime import date
        now = datetime.now()
        current_time = now.strftime("%H : %M : %S")
        today = date.today()
        print('\n\n')
        print ('-*75')
        print ("Company's inventory --> Date:",today,"Time:",current_time)
        print ('-*75')

```

```

def check_inventory(): # check inventory

    if quantity <= stock:
        print("Sufficient stock")
    elif quantity > stock:
        print("Insufficient stock")
    left_stock = stock - quantity
    print ("Remaining stocks:", left_stock)
    print('\n')

file = open('item.txt','r') # read and then print the content in the item.txt
print(file.read())
file.close()

print("\n")
item = input("Enter item Code <Q>uit >>") # enter item code that want to check

while item.upper() != 'Q': # Q to exit

    file= open('item.txt','r')
    list_of_file=[]
    for line in file:      #change content in the item.txt to list
        strip_line=line.strip()
        line_list = strip_line.split()
        list_of_file.append(line_list)
    file.close()

    #check inventory
    if item in list_of_file[0] :
        item1=''.join(list_of_file[0])
        print(item1)
        stock = 50
        print('Current inventory >> ',stock)
        quantity = int(input("Enter quantity >>"))
        check_inventory()
        item = input("Enter item Code <Q>uit >>")

    elif item in list_of_file[1] :
        item1=''.join(list_of_file[1])
        print(item1)
        stock = 88
        print('Current inventory >> ',stock)
        quantity = int(input("Enter quantity >>"))
        check_inventory()
        item = input("Enter item Code <Q>uit >>")

    elif item in list_of_file[2] :
        item1=''.join(list_of_file[2])
        print(item1)
        stock = 50
        print('Current inventory >> ',stock)
        quantity = int(input("Enter quantity >>"))
        check_inventory()
        item = input("Enter item Code <Q>uit >>")

```

```

elif item in list_of_file[3] :
    item1=''.join(list_of_file[3])
    print(item1)
    stock = 60
    print('Current inventory >> ',stock)
    quantity = int(input("Enter quantity >>"))
    check_inventory()
    item = input("Enter item Code <Q>uit >>")

elif item in list_of_file[4] :
    item1=''.join(list_of_file[4])
    print(item1)
    stock = 80
    print('Current inventory >> ',stock)
    quantity = int(input("Enter quantity >>"))
    check_inventory()
    item = input("Enter item Code <Q>uit >>")

elif len(list_of_file)>5 and item in list_of_file[5] :
    item1=''.join(list_of_file[5])
    print(item1)
    stock = 100
    print('Current inventory >> ',stock)
    quantity = int(input("Enter quantity >>"))
    check_inventory()
    item = input("Enter item Code <Q>uit >>")

else:
    print('invalid code!! Please key again')
    print('\n')
    item = input("Enter item Code <Q>uit >>")

choice2()
pointofsale_menu()
choice_no = input('Enter option >> ')

elif choice_no == '3':
# 3 Membership Maintenance
def choice3():
    import datetime
    import time
    import ast

def print3():
    from datetime import datetime
    from datetime import date
    now = datetime.now()
    current_time = now.strftime("%H : %M : %S")
    today = date.today()
    print('\n\n')
    print('-' * 75)
    print("Membership info --> Date:", today, "Time:", current_time)
    print('-' * 75)

```

```

print("<1> Introduction of membership\n<2> Membership Registration\n<3> Check membership
status\n<Q> Quit")

def name(fullname): # check the valid of name
    if len(fullname)==0:
        print("Please do not leave the query empty.")

    elif not all(x.isalpha() or x.isspace() for x in fullname):
        print("Only text allowed in name.")

    else:
        return"

def icno(icnum): # check the valid of ic number
    if len(icnum)<12 or len(icnum)>12:
        print("Please enter a valid IC number.")

    elif not icnum.isdigit():
        print("Please use numbers only.")

    else:
        for line in open("database of membership.txt", "r").readlines(): # check the ic number have exists in the
            database.txt or not
            if icnum in line:
                print("IC number exists. Please insert another IC number.\n")
                break

        else:
            return"

def date(rdate): # check the date that inserts is today
    try:
        y=datetime.datetime.strptime(rdate,"%d/%m/%Y").date()

    except:
        print("Invalid input. Please try again.")

    else:
        today = datetime.date.today()
        if y != today:
            print("Please insert today's date.")

    else:
        return"

def check_expiry_date(db_rdate): # check the membership expired or not, if exceed one year for
registration date, it expired
    i = +365
    today = datetime.datetime.today()
    next_year= db_rdate + datetime.timedelta(days=i)
    if today > next_year:
        print("But your membership card is expired")

    else:
        print("Your membership card is still valid.")

```

```

def member_tier(): # membership tier
    if tier == "P":
        print("Membership tier: Platinum")
        print("Discount: 30%\nEvery month pay Rm30\nGet one year membership ")

    elif tier == "G":
        print("Membership tier: Gold")
        print("Discount: 20%\nEvery month pay Rm20\nGet one year membership")

    elif tier == "S":
        print("Membership tier: Silver")
        print("Discount: 10%\nEvery month pay Rm10\nGet one year membership")

    elif len(tier) == 0:
        print("Please do not leave the input empty")

    else:
        print("Invalid input")

def check_memberstatus(linerecord): # check the membership id that insert and check the
                                   membership status that is expired or not
    if len(check_list[0]) != 9:
        print('Invalid membership id')

    elif not (check_list[0][0].isalpha() and check_list[0][1:10].isdigit()):
        print('Wrongly of the format of membership id')

    elif check_list[0] in linerecord:

        with open('database of membership.txt','r') as f:
            f_line = f.readlines()
            f.seek(0)

            for line in f_line:
                adjust_line = line.split(' ')
                db_rdate=datetime.datetime.strptime(adjust_line[2],"%d/%m/%Y")

                if check_list[0] in line:

                    if check_list[0][0] == 'P':
                        print('You are the platinum member.')
                        check_expiry_date(db_rdate)

                    elif check_list[0][0] == 'S':
                        print('You are the silver member.')
                        check_expiry_date(db_rdate)

                    elif check_list[0][0] == 'G':
                        print('You are the gold member.')
                        check_expiry_date(db_rdate)

                elif check_list[0] not in linerecord:
                    print('Unregistered')

            else:
                print('Invalid membership id')

```

```

print3()
choice = input('Enter option >> ') # option

while choice.upper() != 'Q': # Q to exit

    #key in 1 for introduction of member tier
    if choice == '1':
        tier = input("Please input Membership tier (<P>latinum <G>old <S>ilver <Q>uit): ").upper()

        while tier != 'Q':
            member_tier()
            print('\n')
            tier = input("Please input Membership tier (<P>latinum <G>old <S>ilver <Q>uit): ").upper()

    print3()
    choice = input('Enter option >> ')

# key in 2 for Membership Registration
elif choice == "2":
    a = 0
    member_record=[] # to record name,ic no,registration date,tier
    while a<1 :

        print("\nMembership registration fee = RM10\nPlease provide information below to complete
              registration")

        while True:
            fullname = input("Please insert full name: ") # insert name
            if name(fullname) ==":": # if fullname is valid, define function(name(fullname)) will return ", then
                                         to append fullname into member_record
                member_record.append(fullname)
                break

        while True:
            icnum = input("Please input IC number: ") # insert ic
            if icno(icnum) ==":": # if ic number is valid, define function(icno(icnum)) will return ", then
                                         to append icnum into member_record
                member_record.append(icnum)
                break

        while True:
            rdate = input("Please select date (eg.dd/mm/yyyy): ") # insert today date,must follow the
                                         format dd/mm/yyyy

            if date(rdate) ==":": # if date that insert is today, define function(date(rdate)) will return ", then to
                                         append rdate into member_record
                member_record.append(rdate)
                break

        while True:
            tier = input('Please select the member tier that you want (<P>latinum <G>old <S>ilver) >>
                         ').upper() # insert tier that wanted

            member_tier()

```

```

member_record.append(tier) # append tier that chooses into member_record
break

#Confirmation
if tier == 'P':
    tier = 'Platinum'

elif tier == 'G':
    tier = 'Gold'

elif tier == 'S':
    tier = 'Silver'

print("\n")
print(f"{'Name': <18}: {fullname} ")
print(f"{'IC number': <18}: {icnum} ")
print(f"{'Registration date': <18}: {rdate} ")
print(f"{'Member tier': <18}: {tier} ")

signup = input("Signup for Membership? <Y>es <N>o : ").upper() # key Y to complete the
                                                               registration of membership, N to cancel
if signup =='Y':

    # membership id composed of first letter of tier, last 4 digits of ic and year of registration date
    member_id = member_record[3]+member_record[1][8:]+member_record[2][6:]
    member_record.append(member_id) # append member_id into member_record

    with open("database of membership.txt","a+") as customer_data: #open database of membership
                                                               .txt, append+read member_record into it
        wstr = '|'.join(member_record) + "\n"
        customer_data.write(wstr)
        customer_data.close()

    print("\nYour membership has been registered.\nCongratulation!! Now you are the " + tier + "
          member\nYou membership id is "+ member_id)

elif signup =='N':
    print("\nYour membership has not been registered.")

else:
    print("Invalid input. Please try again.")

a += 1 # automatic exit the loop
print3()
choice = input('Enter option >> ')

#key in 3 to check membership status
elif choice == "3":

    check_list=[] #create empty list
    check = input('Please insert membership id <Q>uit : ').upper() # insert membership id

    while check != 'Q': # Q to exit

        check_list.append(check) #check_list append check

```

```

with open('database of membership.txt','r') as f: # open and read database of membership.txt
    global linerecord
    f_line = f.readlines()
    f.seek(0)
    linerecord = [] #create empty list

for line in f_line: #modify line into lists and record the firth element(membership id) of every list
    into linerecord

    line_strip = line.strip('\n')
    line_split = line_strip.split(' ')
    linerecord.append(line_split[4])

check_memberstatus(linerecord)
check_list.clear() #clean the list
check = input('Please insert membership id <Q>uit : ').upper()

print3()
choice = input('Enter option >> ')

# Invalid option
else:
    print('Invalid option\n')
    choice = input('Enter option >> ')

choice3()
pointofsale_menu()
choice_no = input('Enter option >> ')

#sales
elif choice_no =='4':

def member_tier(subtotal):
    global subtotal1
    if tier.upper() == "P":
        tier1 = "Platinum"
        discount1 = "30%"
        subtotal1 = subtotal*.7
        print("Membership tier:",tier1,"(discount: ",discount1,")\n")

    elif tier.upper() == "G":
        tier1 = "Gold"
        discount1 = "20%"
        subtotal1 = subtotal*.8
        print("Membership tier:",tier1,"(discount: ",discount1,")\n")

    elif tier.upper() == "S":
        tier1 = "Silver"
        discount1 = "10%"
        subtotal1 = subtotal*.9
        print("Membership tier:",tier1,"(discount: ",discount1,")\n")

    elif tier.upper() == "N":
        tier1 = "No membership"

```

```

discount1 = "0%"
subtotal1 = subtotal+0
print("Membership tier:",tier1,"(discount: ",discount1,")\n")

elif len(tier) == 0:
    print("Please do not leave the input empty\n")

else:
    print("Invalid input\n")

#record transaction in the report
def report(report_list):

    report= input('Do you record this transaction? <Y>es <N>o >> ') # key Y to record the transaction in
                                                                report.txt
    while report.upper() != 'N':
        if report.upper() == 'Y':
            with open('report.txt','a+') as f:      #append+read report_list into report.txt
                wStr= ".join(str(report_list)) + '\n'
                f.write(wStr)
                f.close()
            report = 'N'

        else:
            print('invalid option')
            report = input('Do you record this transaction? <Y>es <N>o >> ')

def paymethod(pay_method):
    global method
    a = 'A'
    while a == 'A':  #pay method

        if pay_method.upper() == 'E' :
            method = 'E-WALLET'
            a = 'B'

        elif pay_method.upper() == 'C' :
            method = 'CASH'
            a = 'B'

        elif pay_method.upper() == 'CC' :
            method = 'CREDIT CARD'
            a = 'B'

        else:
            print('No support this method of payment!!Please key in again!!!!')
            pay_method = input('Paying method >> ').upper()

def choice4():
    print("\n\n")
    print('*'*75)
    import datetime

    x = datetime.datetime.now()

    print('Sales Transaction payment ---> ', 'DATE:', x.strftime("%x") + ' ' + 'TIME:', x.strftime("%X") + ' ')

```

```

        + x.strftime("%A"))
print ('-'*75)

print('' + '\t' + 'QTY' + ' \t' + 'RM')

#payment
def payment():
    global subtotal
    global report_list
    report_list=[0,0,0,0,0,0] #used to record quantity of product 1,2,3,4,5,6 and total price
    item_list1 = []
    buy = input('Enter item code <Q>uit to stop payment> ').upper()

    file= open('item.txt','r')
    item_list2=[]
    for line in file:
        strip_line = line.strip()
        item_list3 = strip_line.split()
        item_list2.append(item_list3)
    file.close()

    subtotal= 0

    while buy != 'Q':

        if buy in item_list2[0] :
            stock = 50
            quantity = int(input('Enter quantity >> '))

            if quantity > stock:
                print('Insufficient stock')

            else:
                total1 = quantity*float(item_list2[0][3])
                print(f'{item_list2[0][0] :<3} {item_list2[0][1] :<30}RM {item_list2[0][3]:<13 }'
                      f'{quantity :<7}{%.2f%total1} ')
                subtotal += total1
                report_list[0]+= quantity
                item_list1.append('a')

        elif buy in item_list2[1]:
            stock = 88
            quantity2 = int(input('Enter quantity >> '))

            if quantity2 > stock:
                print('Insufficient stock')

            else:
                total2 = quantity2*float(item_list2[1][3])
                print(f'{item_list2[1][0] :<3} {item_list2[1][1] :<30}RM {item_list2[1][3]:<13 }'
                      f'{quantity2 :<7}{%.2f%total2} ')
                subtotal += total2
                report_list[1]+= quantity2
                item_list1.append('b')

        elif buy in item_list2[2]:

```

```

stock = 50
quantity3 = int(input('Enter quantity >> '))

if quantity3 > stock:
    print('Insufficient stock')

else:
    total3 = quantity3*float(item_list2[2][3])
    print(f'{item_list2[2][0]} : <3} {" :<7} {item_list2[2][1] : <30}RM {item_list2[2][3]:<13}
        {quantity3 :<7}{%.2f%total3} ')
    subtotal += total3
    report_list[2]+= quantity3
    item_list1.append('c')

elif buy in item_list2[3]:
    stock = 60
    quantity4 = int(input('Enter quantity >> '))

    if quantity4 > stock:
        print('Insufficient stock')

    else:
        total4 = quantity4*float(item_list2[3][3])
        print(f'{item_list2[3][0]} : <3} {" :<7} {item_list2[3][1] : <30}RM {item_list2[3][3]:<13}
            {quantity4 :<7}{%.2f%total4} ')
        subtotal += total4
        report_list[3]+= quantity4
        item_list1.append('d')

elif buy in item_list2[4]:
    stock = 80
    quantity5 = int(input('Enter quantity >> '))

    if quantity5 > stock:
        print('Insufficient stock')

    else:
        total5 = quantity5*float(item_list2[4][3])
        print(f'{item_list2[4][0]} : <3} {" :<7} {item_list2[4][1] : <30}RM {item_list2[4][3]:<13}
            {quantity5 :<7}{%.2f%total5} ')
        subtotal += total5
        report_list[4]+= quantity5
        item_list1.append('e')

elif len(item_list2)>5 and buy in item_list2[5]:
    stock = 100
    quantity6 = int(input('Enter quantity >> '))

    if quantity6 > stock:
        print('Insufficient stock')

    else:
        total6 = quantity6*float(item_list2[5][3])
        print(f'{item_list2[5][0]} : <3} {" :<7} {item_list2[5][1] : <30}RM {item_list2[5][3]:<13}
            {quantity6 :<7}{%.2f%total6} ')
        subtotal += total6

```

```

report_list[5] += quantity6
item_list1.append(f)

else:
    print('invalid code')

buy = input('Enter item code <Q>uit to stop payment >> ').upper()

print('*75)
print('*75)

def payment2():

    i = 0
    while i < 1:
        global tier #member tier: platinum(30% discount), Gold(20% discount), Silver(10% discount)
        tier = input("Please input Membership tier (<P>latinum <G>old <S>ilver <N>o): ").upper()
        member_tier(subtotal) #define function
        i += 1

        #subtotal
        print(f"{'Subtotal(RM)':<52}{'.2f' % subtotal} ")
        #price of discount
        print(f"{'Discount(RM)':<51}{'-'} {'%.2f' % float(subtotal-subtotal1)} ")

        #rounded price that after discount
        def adjust(subtotal1):
            adjustment = round(subtotal1,1)
            adj_pri = adjustment - subtotal1 # calculate how much the rounded price
            return adj_pri

        adj_pri = adjust(subtotal1)
        print(f"{'Rounding adjustment(RM)':<41}{'.2f' % adj_pri} ")

        #total price that want to pay
        total_pri = adj_pri + subtotal1
        print(f"{'Payment paid(RM)':<48}{'.2f' % total_pri} ")
        report_list[6] += float('%.2f'.format(total_pri))

report(report_list) #define function

# receipt
def receipt():
    buy = input('Enter option <R>ceipt key in any to exit >> ')
    while buy.upper() == 'R':
        quantity_item = 0
        f = open('receipt.txt', 'w')
        import datetime
        x = datetime.datetime.now()
        f.write('Sales Transaction Receipt --->' + ' DATE : ' + x.strftime("%x") + ' TIME: ' +
               x.strftime("%X") + ' ' + x.strftime("%A") + '\n')
        f.write('*75 + '\n')
        f.write('DE STATIONARY Retail Sdn Bhd' + '\n')
        f.write('*75 + '\n')
        f.write(' + \|t' + ' QTY' + '\|t' + 'RM' + '\n')

```

```

if 'a' in item_list1:
    f.write(f'{item_list2[0][0] :<3}{" :<7}{item_list2[0][1] :<30}RM {item_list2[0][3]:<13}
        {quantity :<7}{%.2f%total1} " + '\n')
    quantity_item += quantity

if 'b' in item_list1:
    f.write(f'{item_list2[1][0] :<3}{" :<7}{item_list2[1][1] :<30}RM {item_list2[1][3]:<13}
        {quantity2 :<7}{%.2f%total2} " + '\n')
    quantity_item += quantity2

if 'c' in item_list1:
    f.write(f'{item_list2[2][0] :<3}{" :<7}{item_list2[2][1] :<30}RM {item_list2[2][3]:<13}
        {quantity3 :<7}{%.2f%total3} " + '\n')
    quantity_item += quantity3

if 'd' in item_list1:
    f.write(f'{item_list2[3][0] :<3}{" :<7}{item_list2[3][1] :<30}RM {item_list2[3][3]:<13}
        {quantity4 :<7}{%.2f%total4} " + '\n')
    quantity_item += quantity4

if 'e' in item_list1:
    f.write(f'{item_list2[4][0] :<3}{" :<7}{item_list2[4][1] :<30}RM {item_list2[4][3]:<13}
        {quantity5 :<7}{%.2f%total5} " + '\n')
    quantity_item += quantity5

if 'f' in item_list1:
    f.write(f'{item_list2[5][0] :<3}{" :<7}{item_list2[5][1] :<30}RM {item_list2[5][3]:<13}
        {quantity6 :<7}{%.2f%total6} " + '\n')
    quantity_item += quantity6

f.write('\n'+ '-'*75 + '\n')
f.write(f'{ "Subtotal(RM)"}{":<52}{%.2f%subtotal} " + '\n')
f.write(f'{ "Discount(RM)"}{":<51}{'-} {%.2f %float(subtotal-subtotal1)} " + '\n')
f.write(f'{ "Rounding adjustment(RM)"}{":<41}{%.2f%adj_pri} " + '\n')
f.write(f'{ "Payment paid(RM)"}{":<48}{%.2f%total_pri} " + '\n')

pay_method = input('Paying method (<E>-wallet <C>ash <C>redit<C>ard) >> ')
pay_method = pay_method.upper() #pay method

paymethod(pay_method) #define function

f.write('\n'*3 + 'Item purchased: ' + str(quantity_item) + '\n')
f.write(f'{method:<63}RM{%.2f%total_pri}" + "\n'*2)
f.write('Welcome to ABC Retail Sdn Bhd and Thank you!! ^_^ See you again' + '\n')
f.close()
buy = input('key in any to exit >> ')
receipt()

payment2()

payment()

choice4()
pointofsale_menu()
choice_no = input('Enter option >> ')

```

```

elif choice_no == '5': #need everyday clean report.txt in order to record daily transaction.....  

def choice5():  

    checkreport = input('Are you want to check the daily report? <Y>es <N>o >> ')  

    while checkreport.upper() != 'N':  

        if checkreport.upper() == 'Y':  

            print('\n+*75)  

            import datetime  

            x = datetime.datetime.now()  

            print('DAILY REPORT --->','DATE:',x.strftime("%x")+' '+ 'TIME:',x.strftime("%X")+' '+' '+  

                  x.strftime("%A"))  

            print('*75)  

            report_list1 =[]  

            item_list2 =[]  

            file = open('report.txt','r')  

            for line in file: # used to check transaction from report.txt  

                strip_line = line.strip('\n').strip('[]')  

                report_list2 = strip_line.split(',')  

                report_list1.append(report_list2)  

            file.close()  

            file = open('item.txt','r')  

            for line in file: # used to check product from item.txt  

                strip_line = line.strip()  

                item_list3 = strip_line.split()  

                item_list2.append(item_list3)  

            file.close()  

            number_transaction = len(report_list1)  

            print('Product code','\t','Product description','\t','Price of product(1unit)','\t','Number of sold product')  

            print('*75)  

            product1 = 0  

            product2 = 0  

            product3 = 0  

            product4 = 0  

            product5 = 0  

            product6 = 0  

            total = 0  

            productlist = []  

            for i in range (number_transaction-1):  

                product1 += int(report_list1[i+1][0])  

                product2 += int(report_list1[i+1][1])  

                product3 += int(report_list1[i+1][2])  

                product4 += int(report_list1[i+1][3])  

                product5 += int(report_list1[i+1][4])  

                product6 += int(report_list1[i+1][5])  

                total += float(report_list1[i+1][6])

```

```

productlist.append(product1)
productlist.append(product2)
productlist.append(product3)
productlist.append(product4)
productlist.append(product5)
productlist.append(product6)
productlist.append(total)

for j in range (len(item_list2)):
    print(f'{item_list2[j][0]:<17}{item_list2[j][1]:<30}RM{item_list2[j][3]:<30}{productlist[j]}')

print('*75)
print('Number of transaction today is' , number_transaction-1 , 'times.')
print('Total price of transaction today is RM%.2f%productlist[6]')
checkreport= input('Key any to exit >> ')
break

else:
    print('Invalid option')
    checkreport = input('Are you want to check the daily report? <Y>es <N>o >> ')

choice5()
pointofsale_menu()
choice_no = input('Enter option >> ')

# data validation
else:
    print('Invalid option')
    print('\n\n')
    choice_no = input('Enter option >> ')

print('Thank you & GOOD DAY ^-^')

```