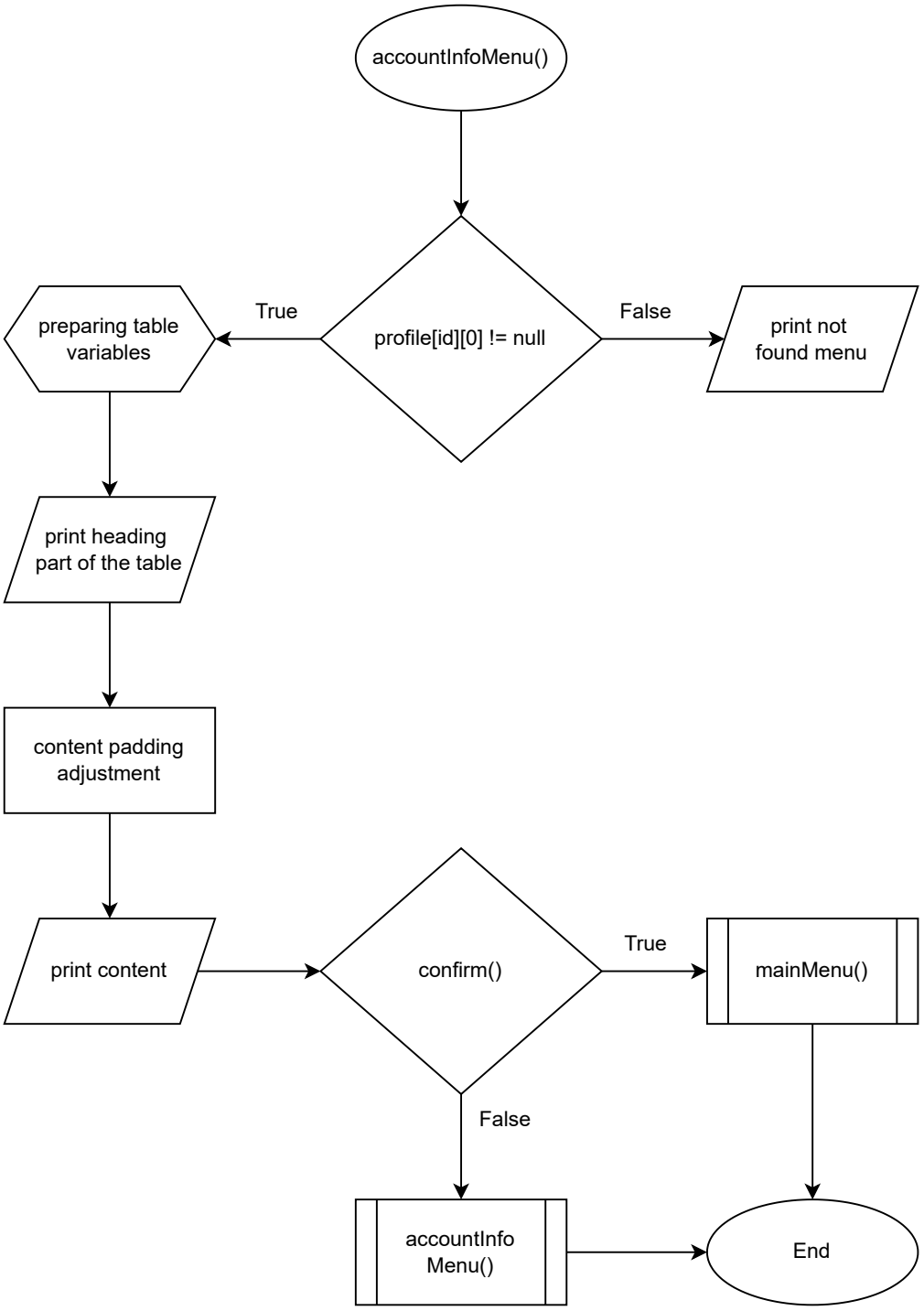
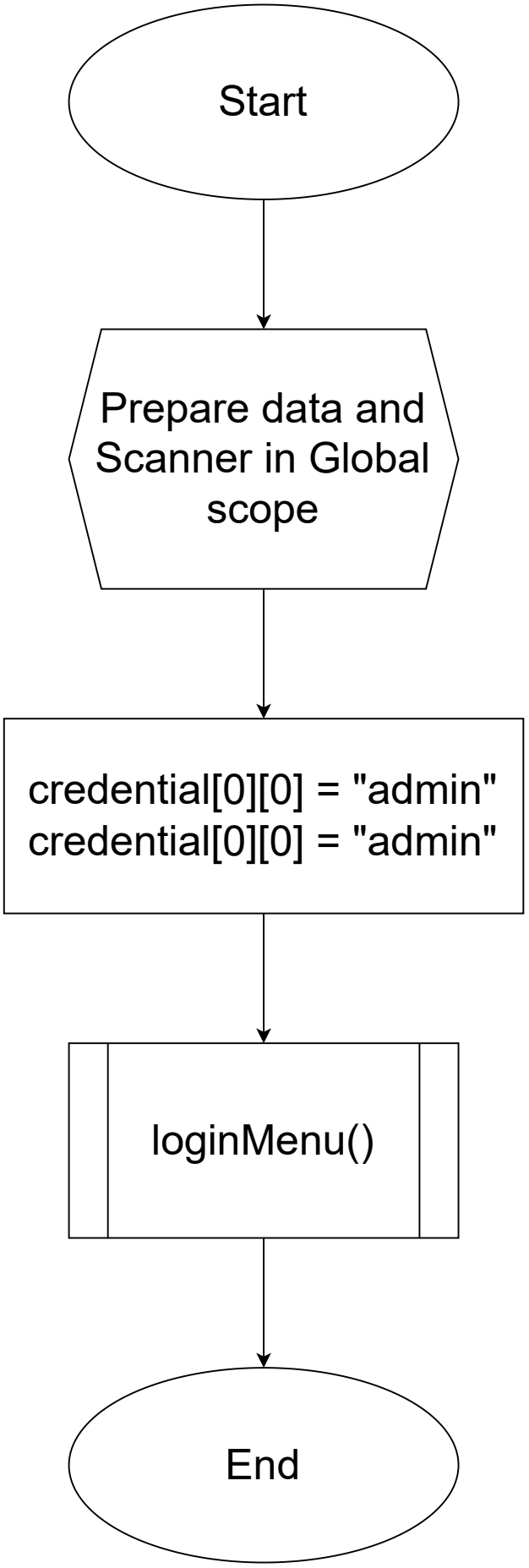
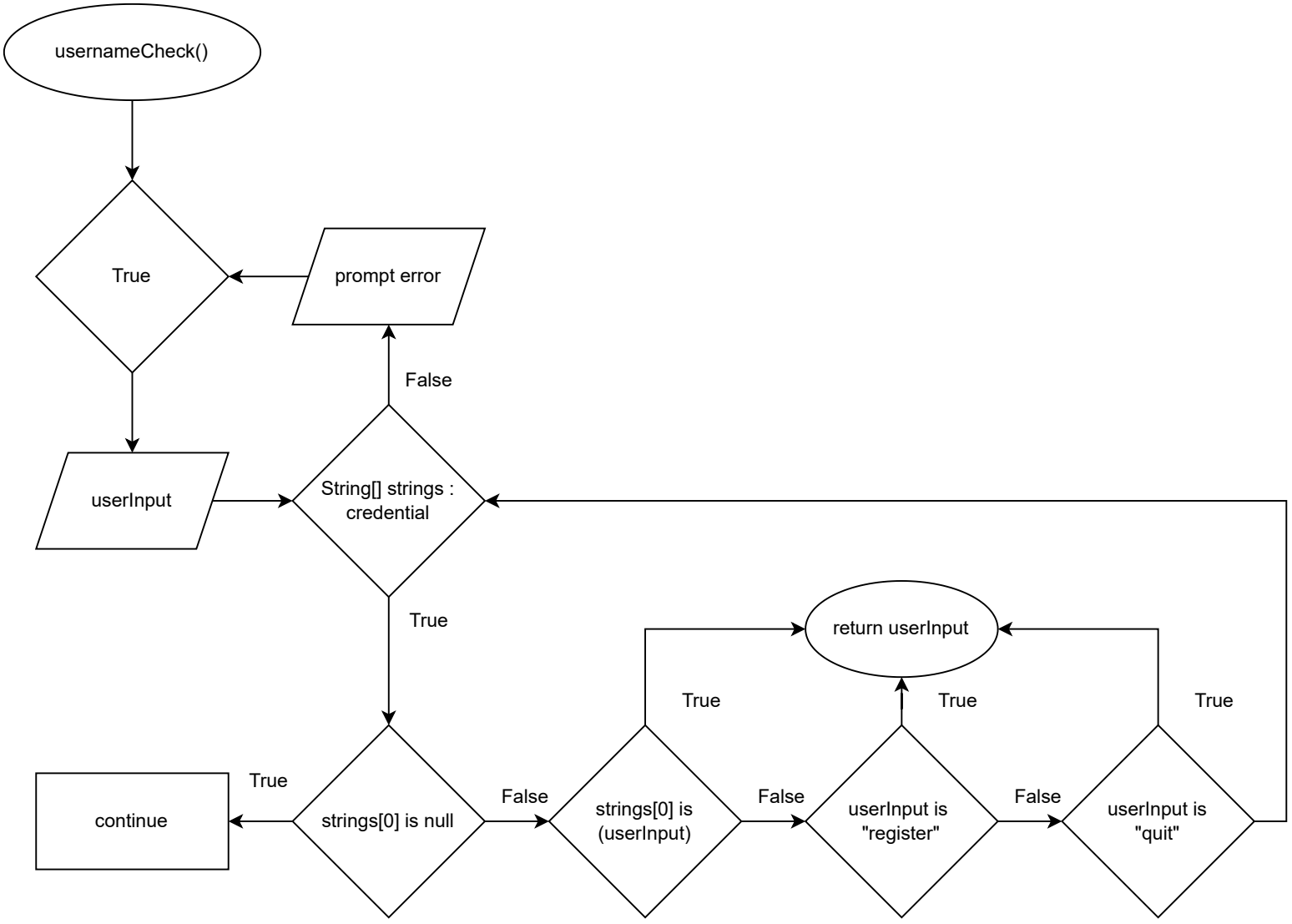
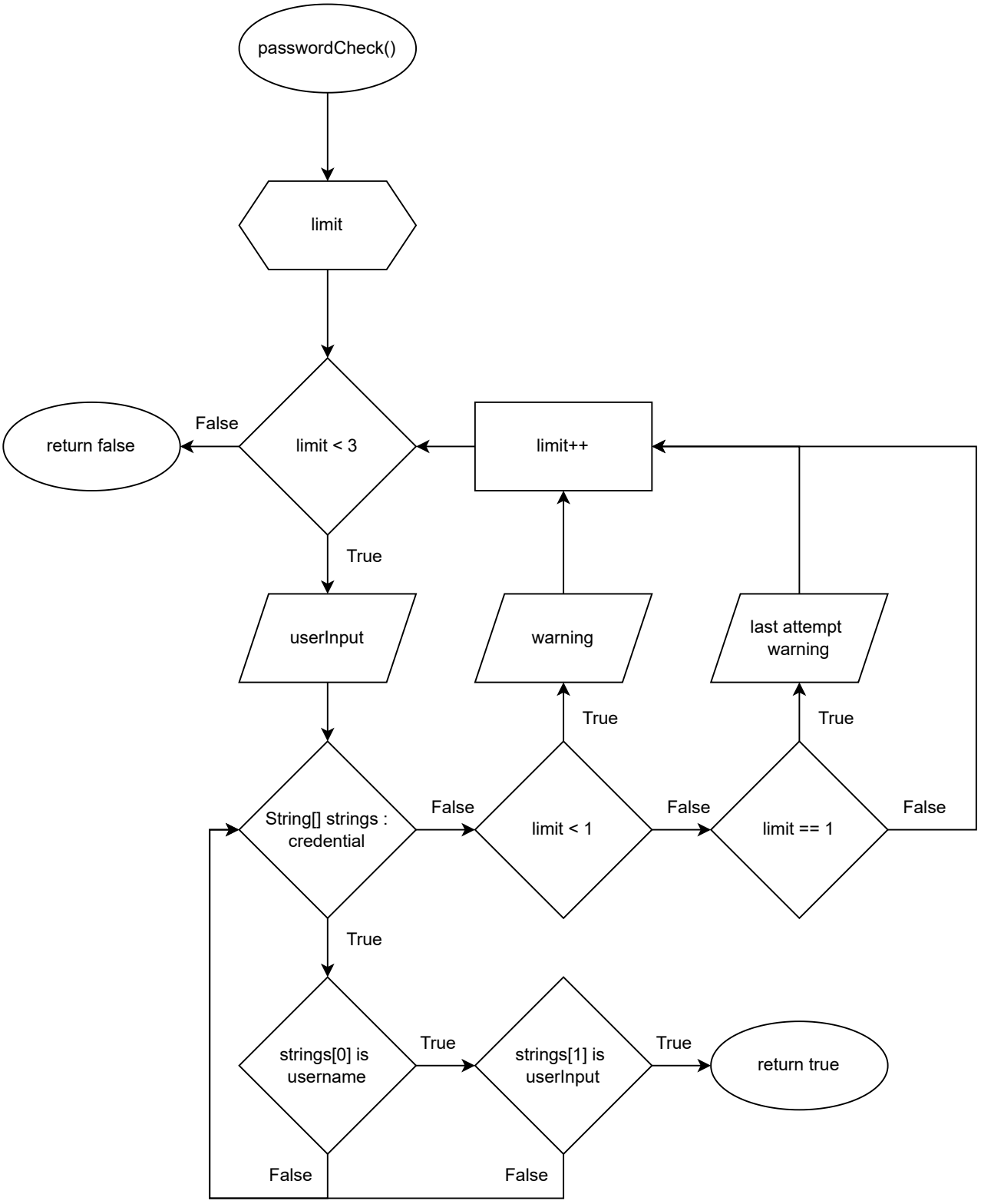
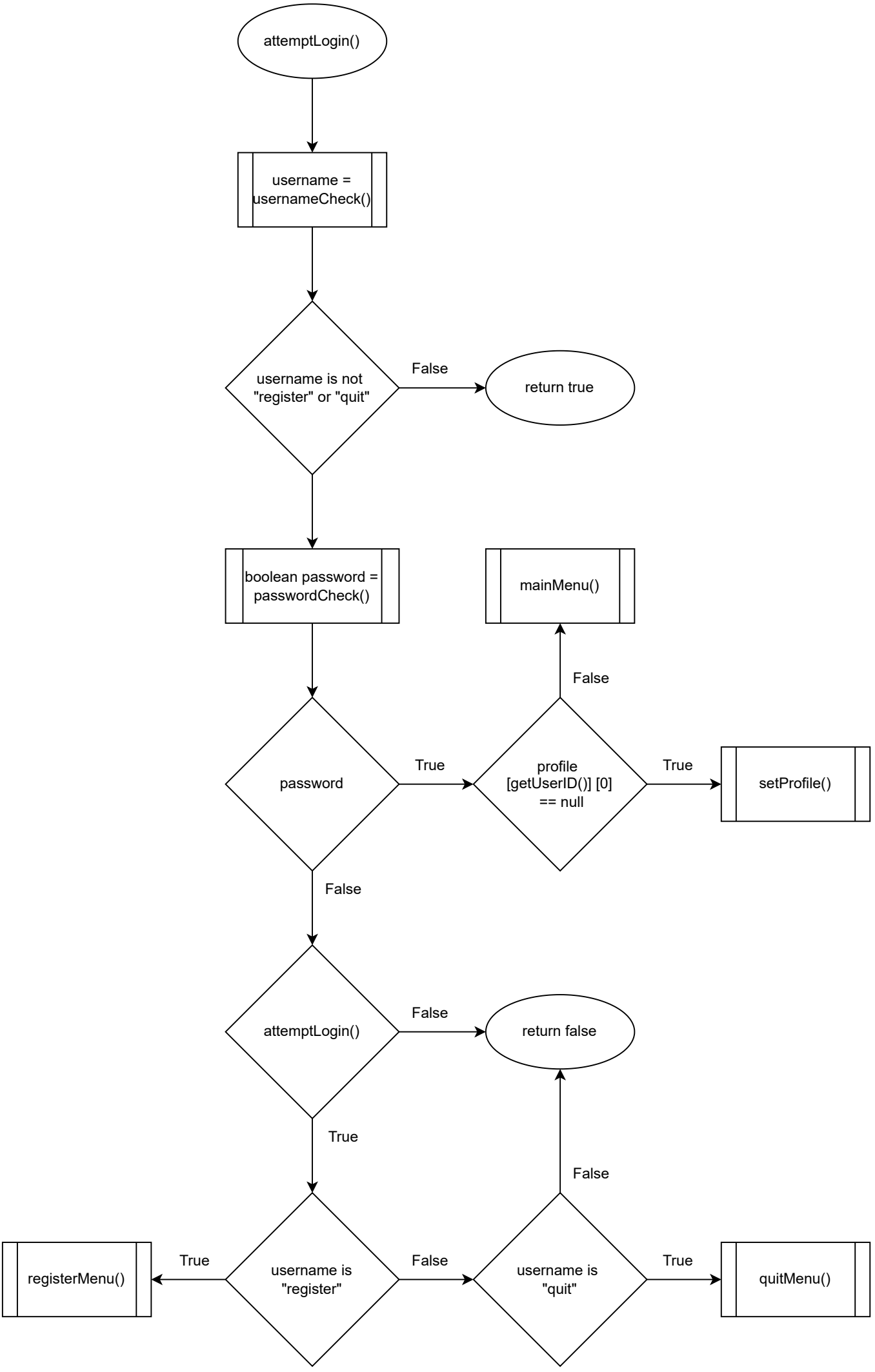
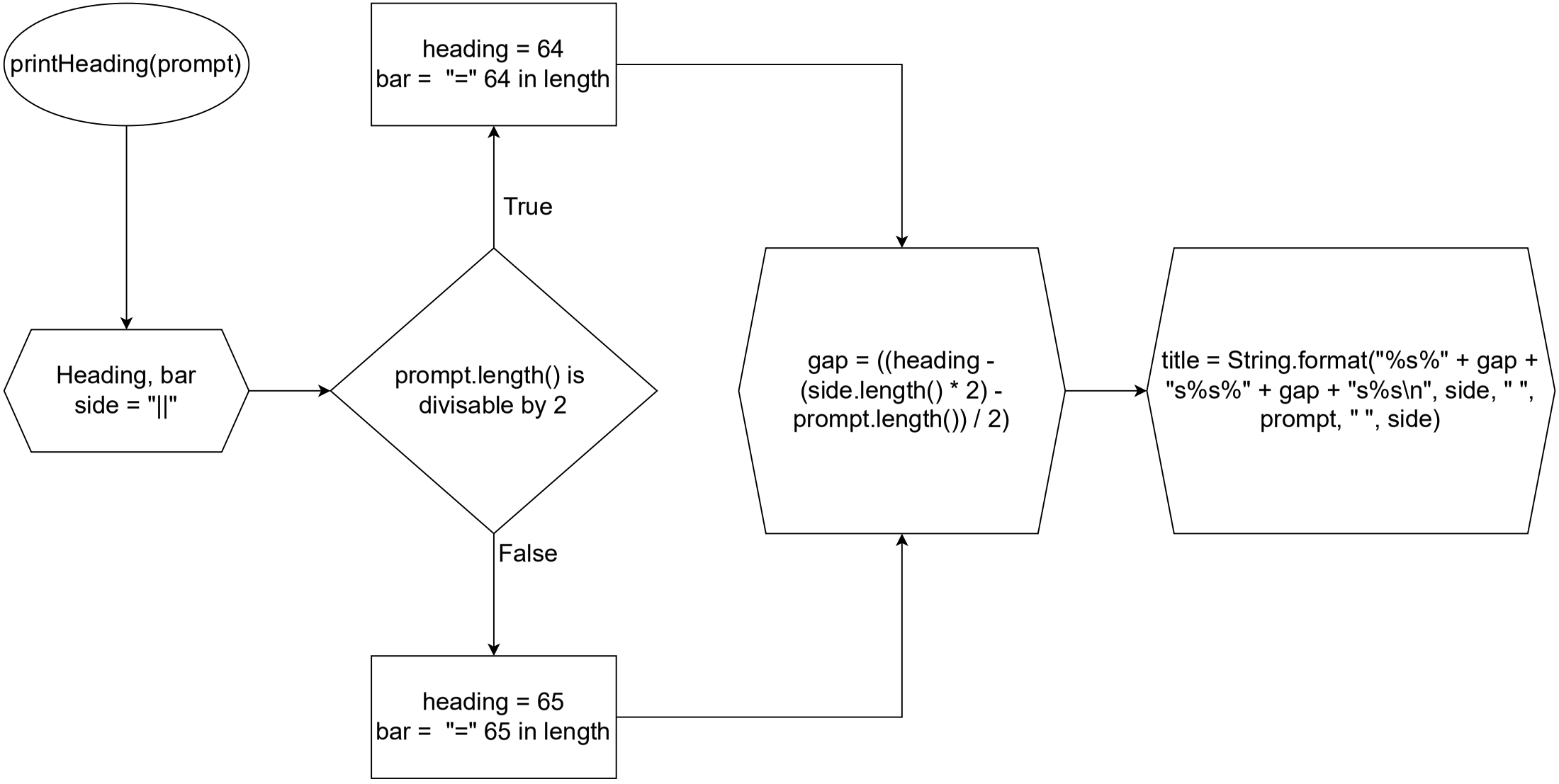

accountInfoMenu()
profile[id][0] != null
True
False
preparing table variables
print not found menu
print heading part of the table
content padding adjustment
print content
confirm()
True
mainMenu()
False
accountInfo Menu()
End

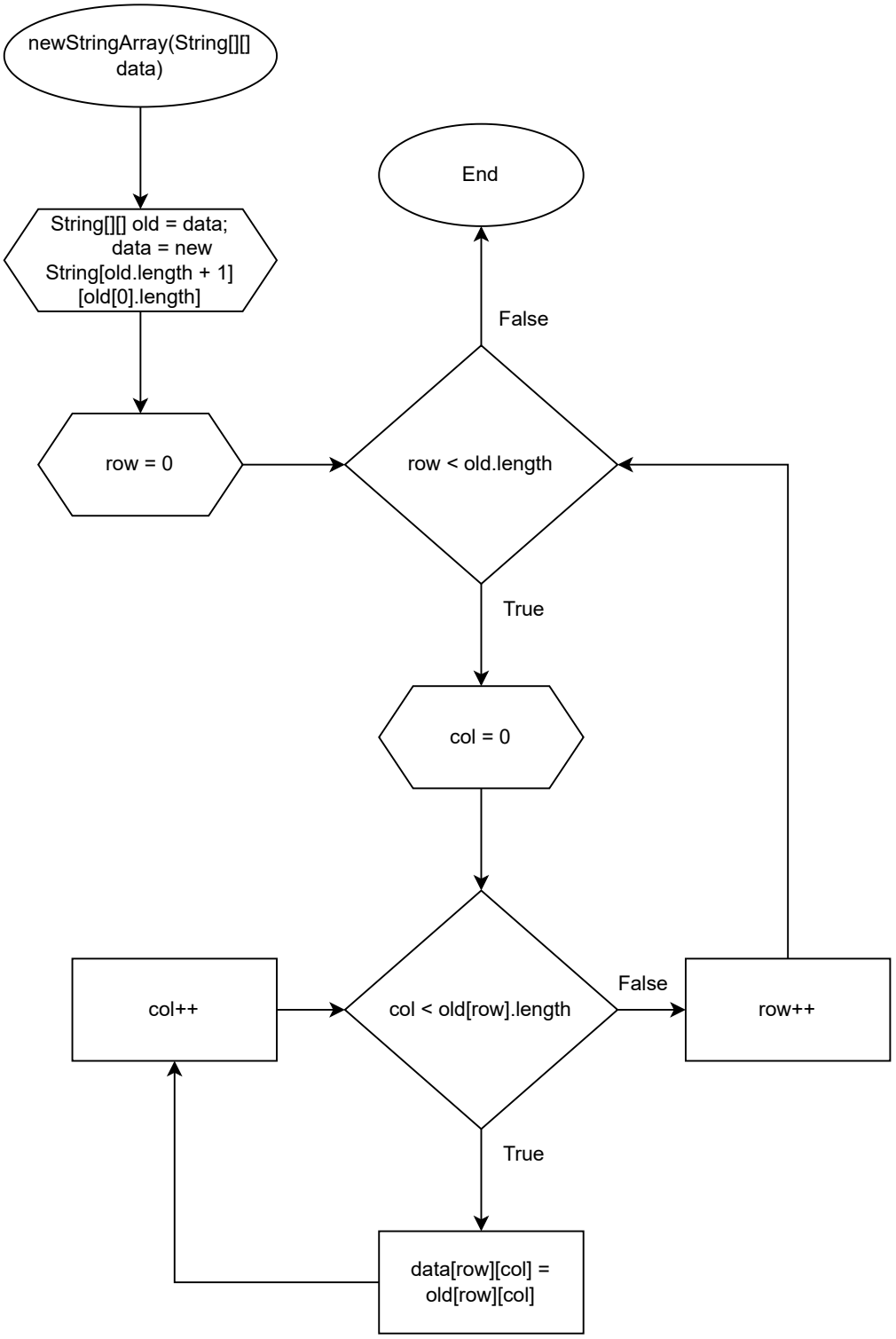

Start
Prepare data and Scanner in Global scope
credential[0][0] = "admin"
credential[0][0] = "admin"
loginMenu()
End

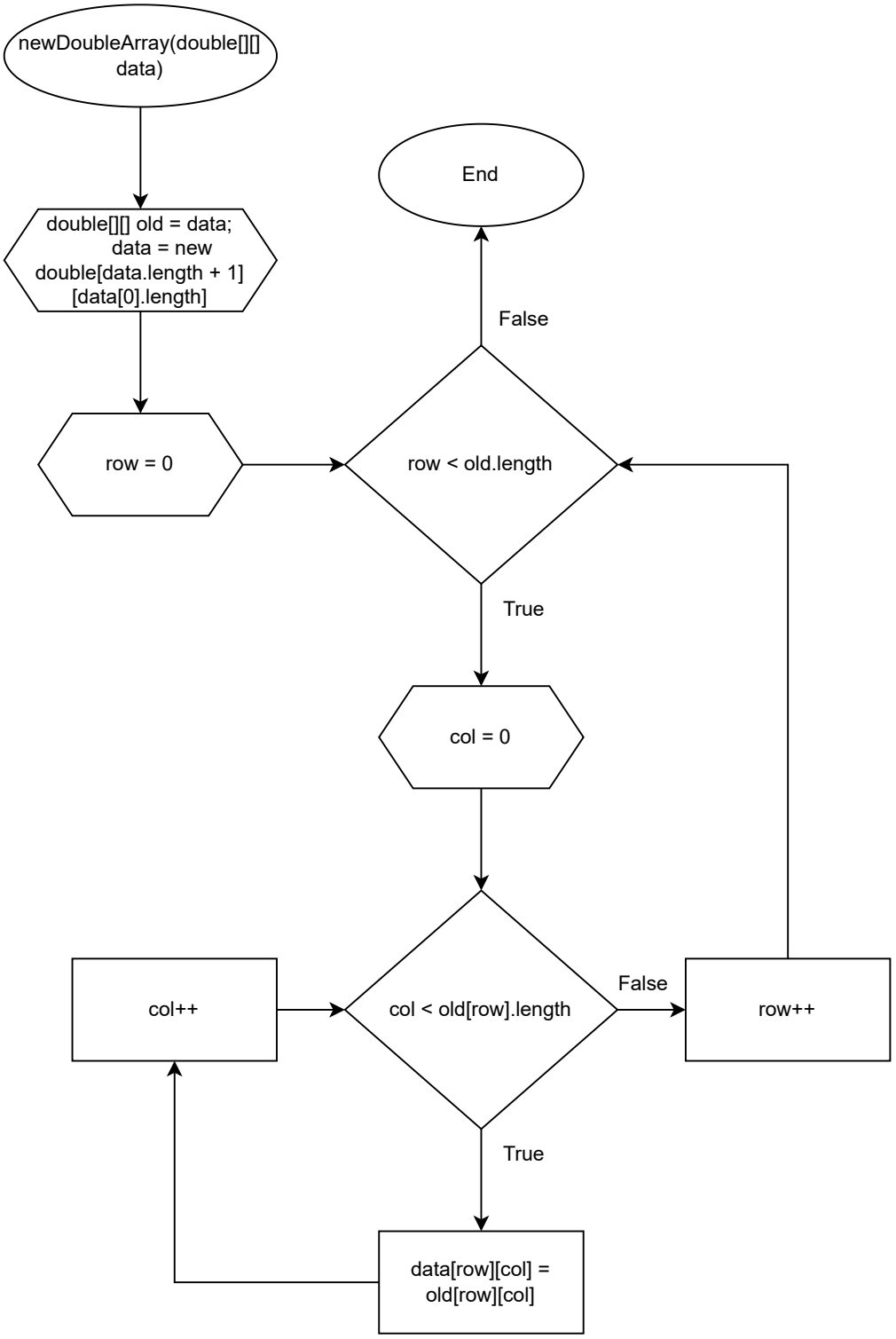

usernameCheck()
True
prompt error
False
userInput
String[] strings : credential
True
return userInput
True
True
True
continue
True
strings[0] is null
False
strings[0] is (userInput)
False
userInput is "register"
False
userInput is "quit"

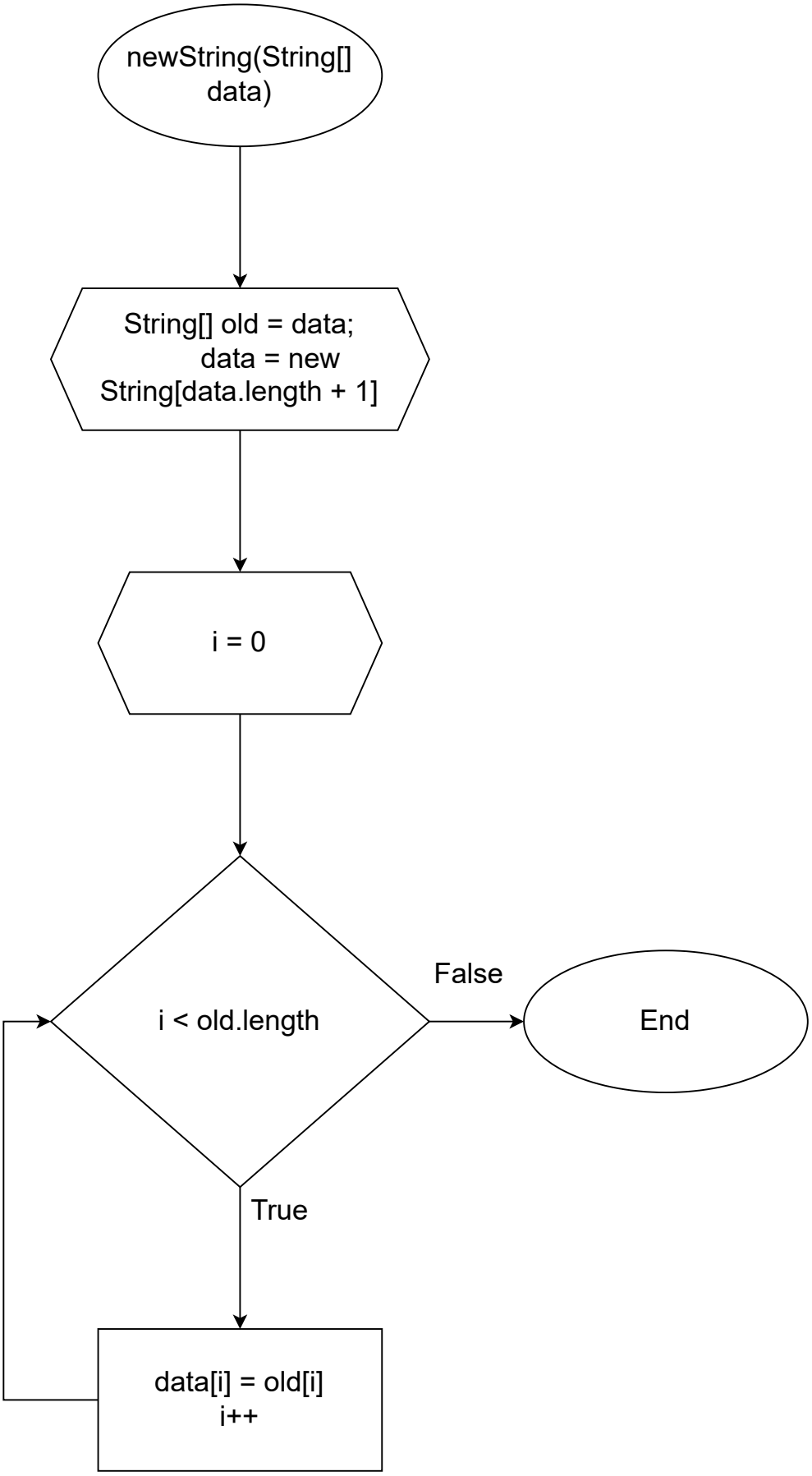

passwordCheck()
limit
limit < 3
False
return false
True
limit++
userInput
warning
last attempt warning
True
True
String[] strings : credential
False
limit < 1
False
limit == 1
False
True
strings[0] is username
True
strings[1] is userInput
True
return true
False
False

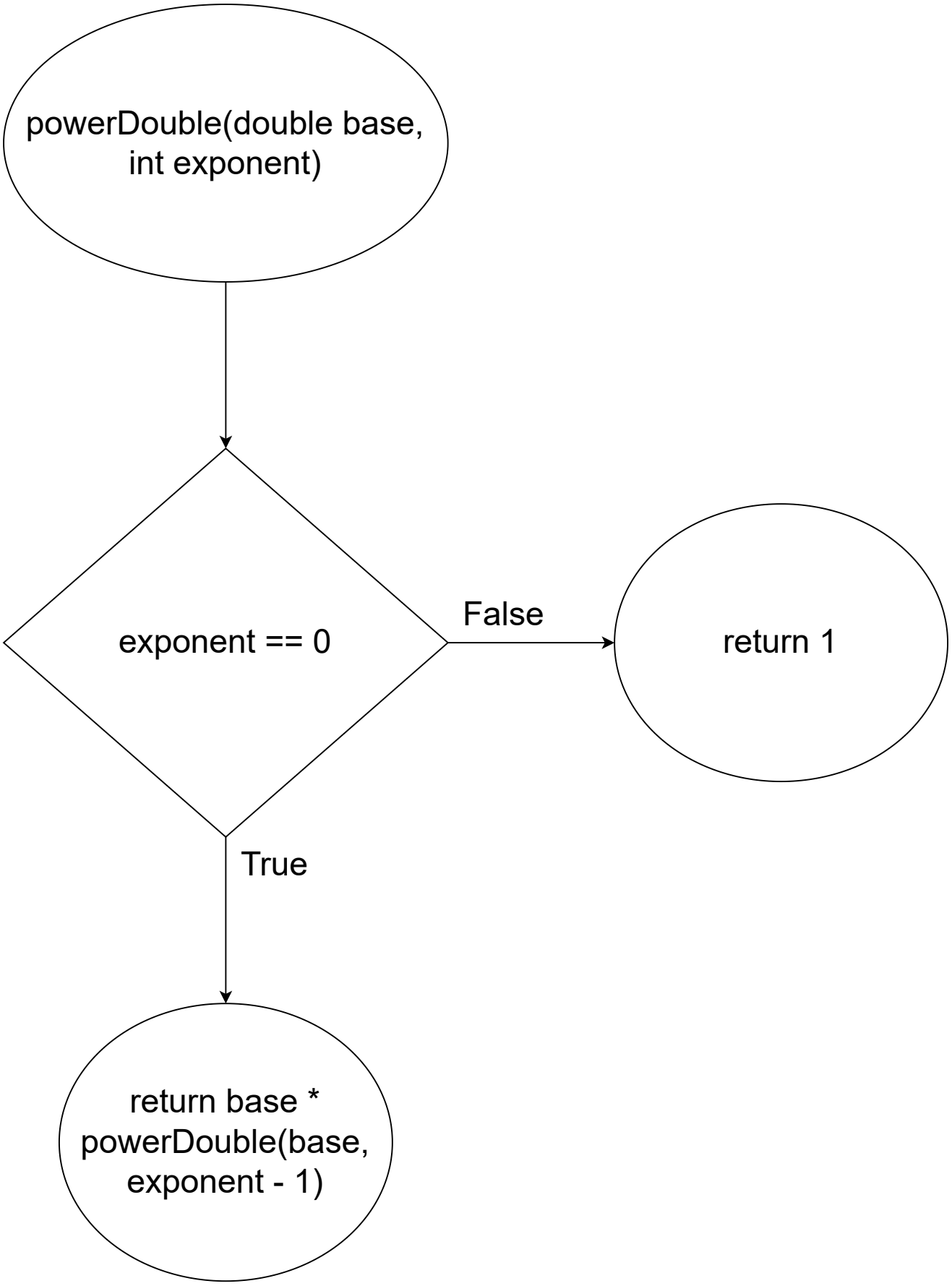

attemptLogin()
username = usernameCheck()
username is not "register" or "quit"
False
return true
boolean password = passwordCheck()
mainMenu()
False
password
True
profile [getUserID()] [0] == null
True
setProfile()
False
attemptLogin()
False
return false
True
False
registerMenu()
True
username is "register"
False
username is "quit"
True
quitMenu()

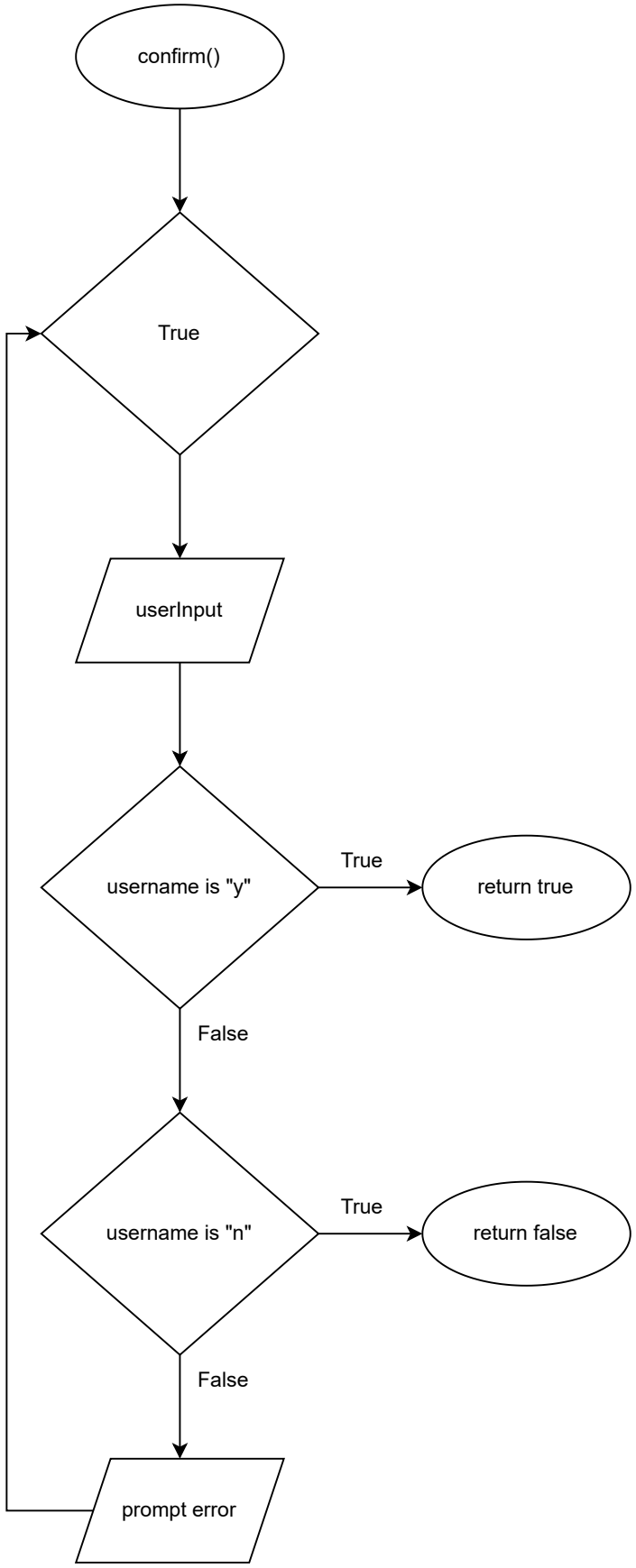

printHeading(prompt)
Heading, bar
side = "||"
prompt.length() is
divisable by 2
True
heading = 64
bar = "=" 64 in length
False
heading = 65
bar = "=" 65 in length
gap = ((heading -
(side.length() * 2) -
prompt.length()) / 2)
title = String.format("%s%" + gap +
"s%s%" + gap + "s%s\n", side, " ",
prompt, " ", side)

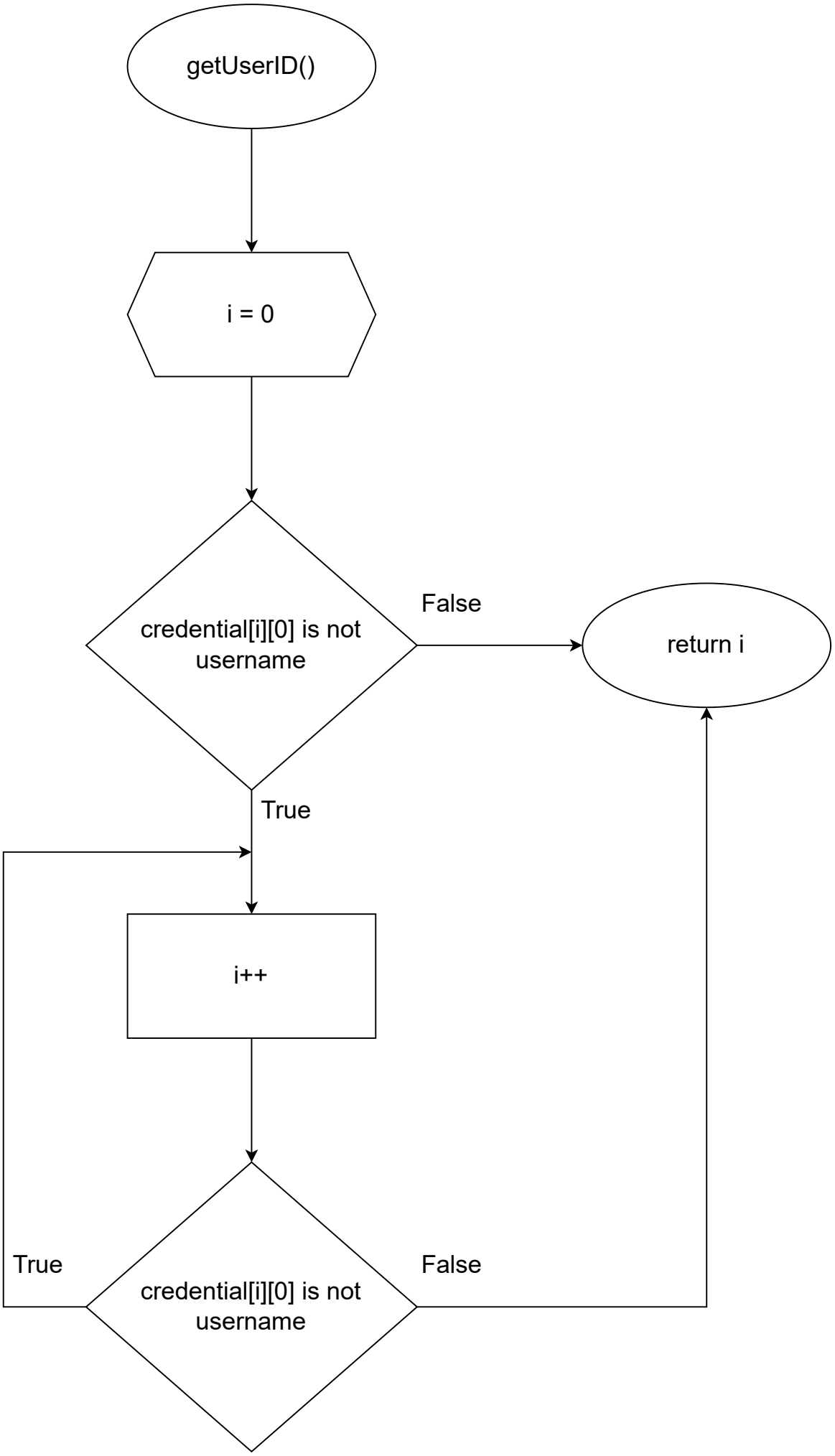

newStringArray(String[][] data)
String[][] old = data; data = new String[old.length + 1] [old[0].length]
row = 0
row < old.length
False
End
True
col = 0
col < old[row].length
False
row++
True
col++
data[row][col] = old[row][col]

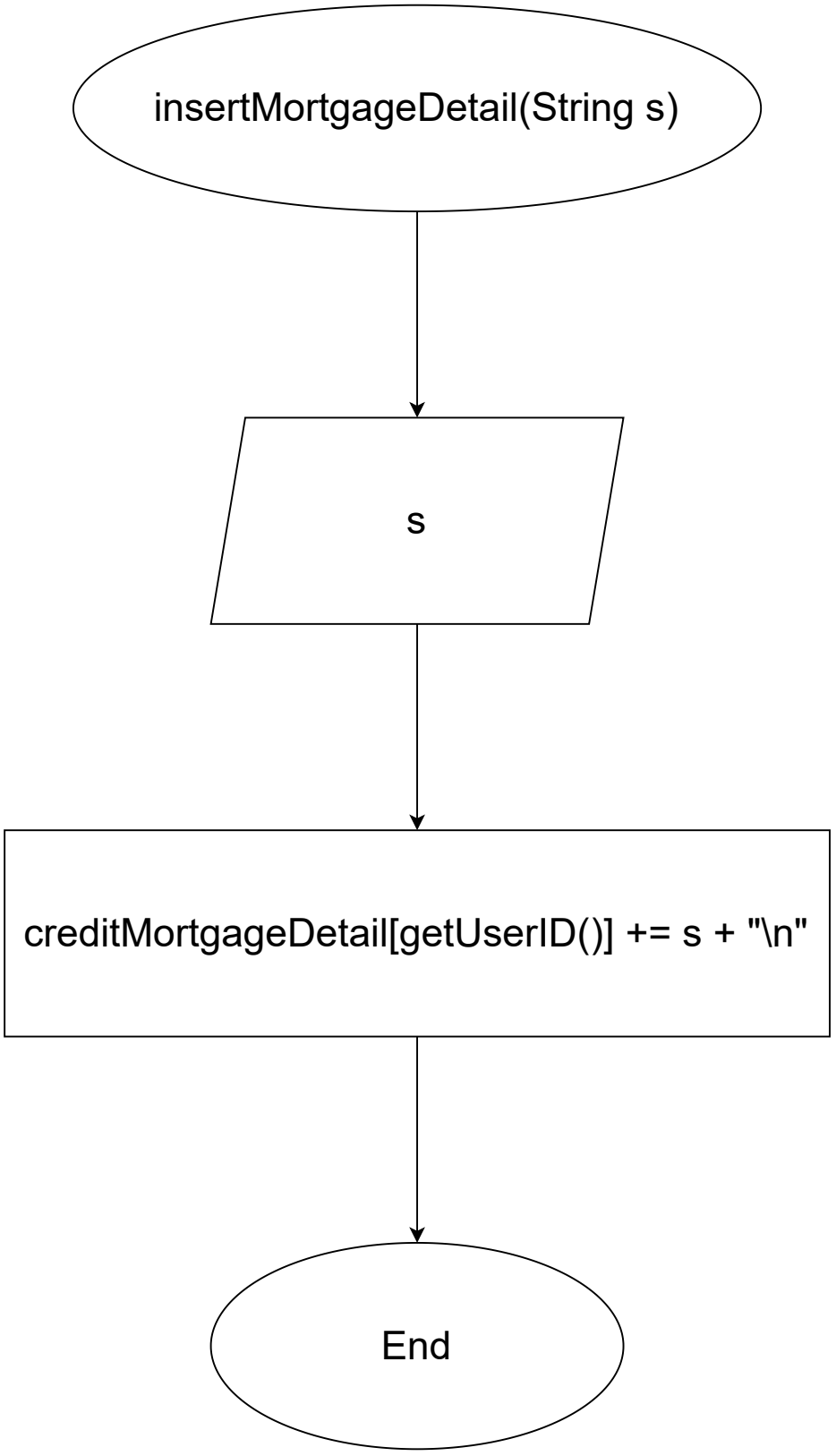

newDoubleArray(double[][] data)
double[][] old = data; data = new double[data.length + 1][data[0].length]
row = 0
row < old.length
False
End
True
col = 0
col < old[row].length
False
row++
True
col++
data[row][col] = old[row][col]

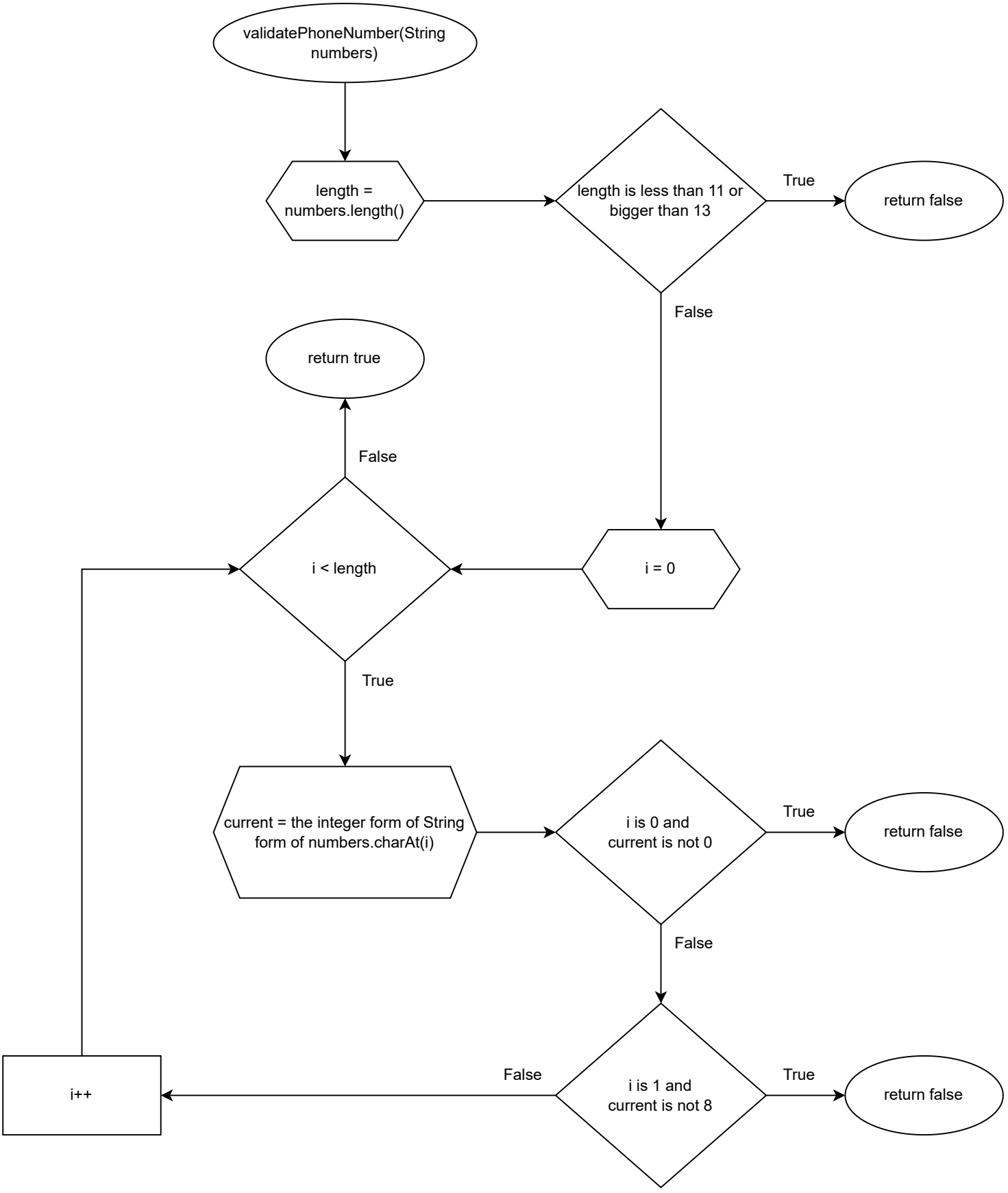

newString(String[] data)
String[] old = data;
data = new String[data.length + 1]
i = 0
i < old.length
False
End
True
data[i] = old[i]
i++


powerDouble(double base, int exponent)
exponent == 0
False
return 1
True
return base * powerDouble(base, exponent - 1)

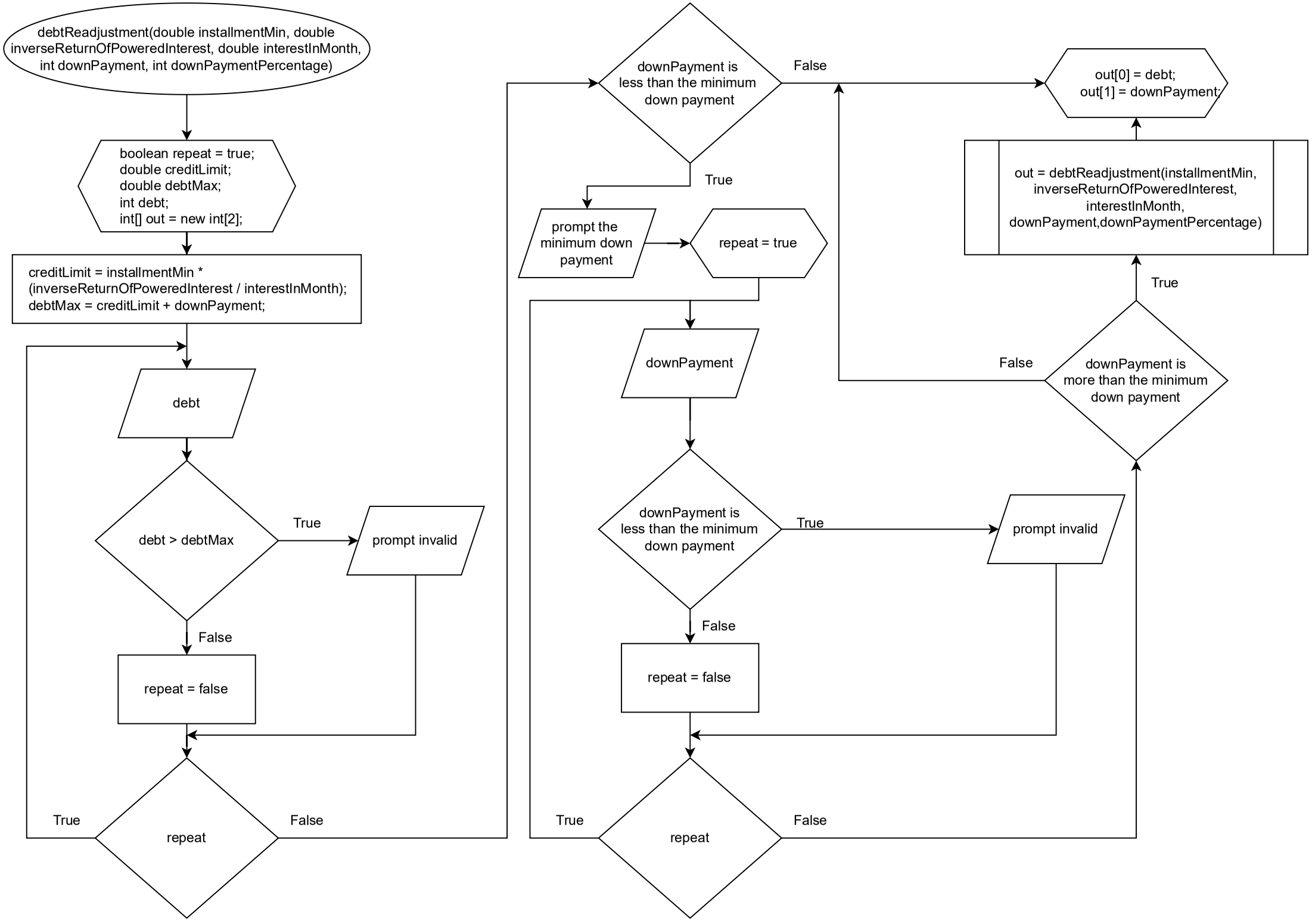

confirm()
True
userInput
username is "y"
True
return true
False
username is "n"
True
return false
False
prompt error

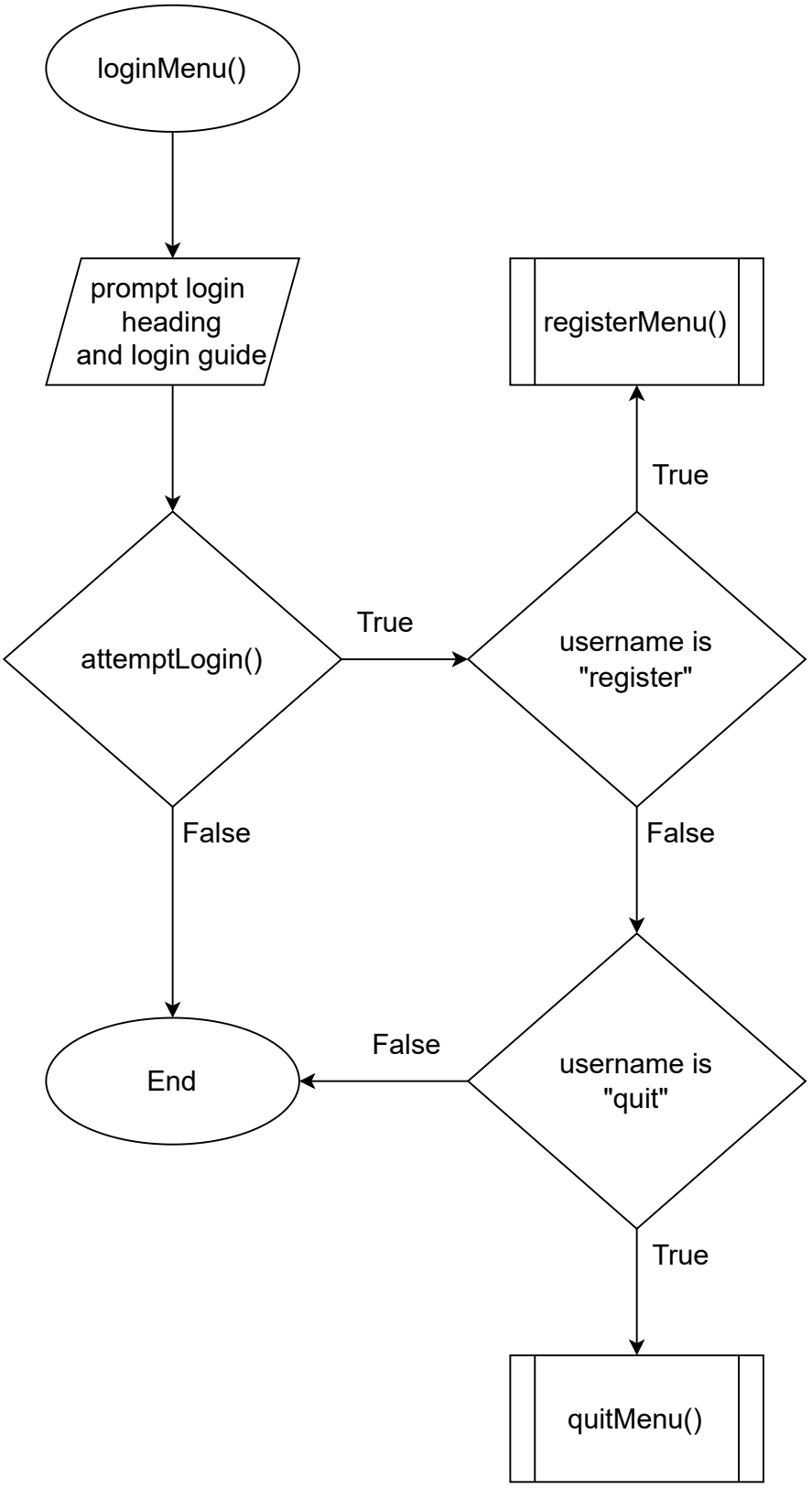

getUserID()
i = 0
credential[i][0] is not username
False
return i
True
i++
credential[i][0] is not username
True
False

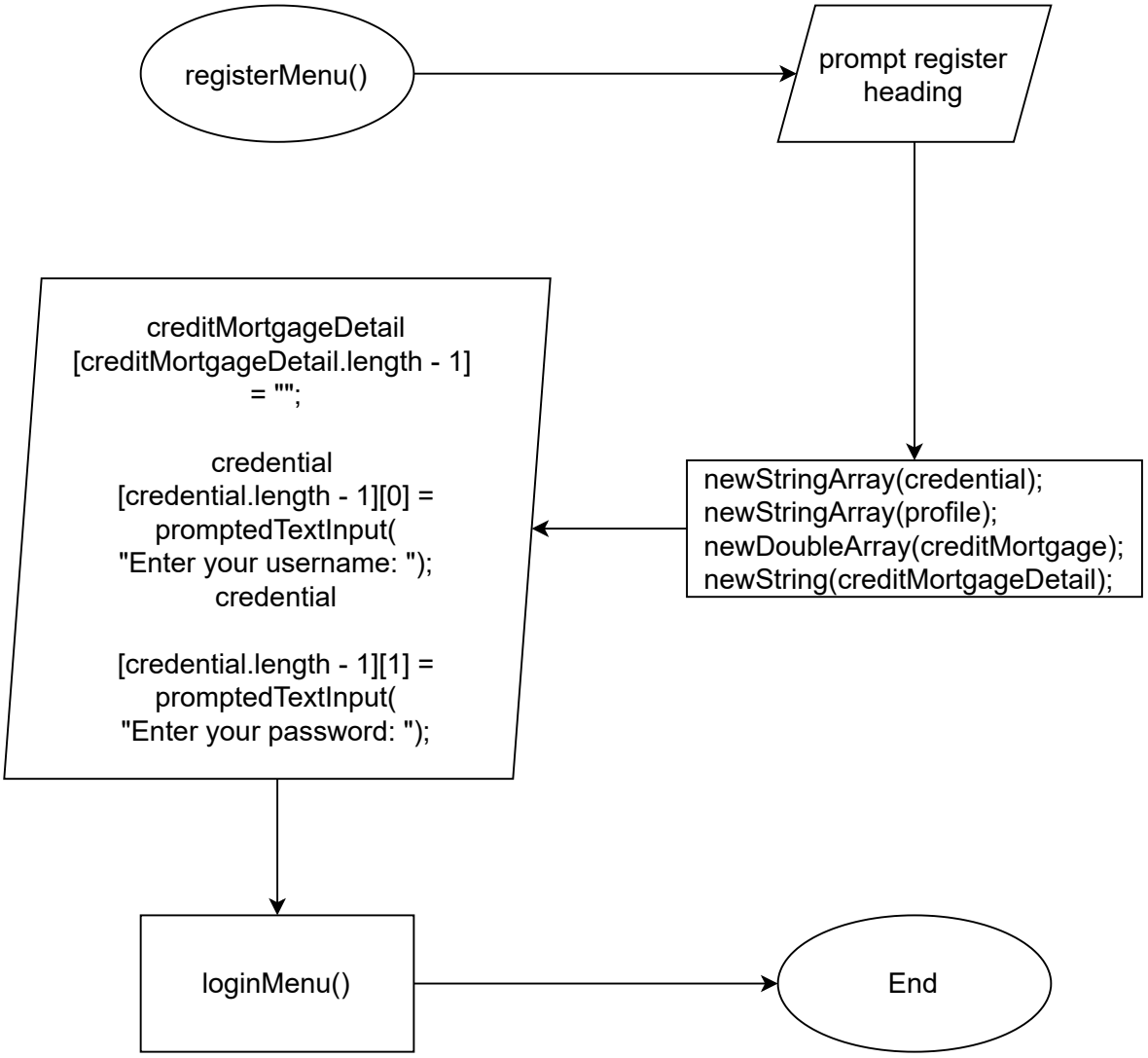

insertMortgageDetail(String s)
s
creditMortgageDetail[getUserID()] += s + "\n"
End

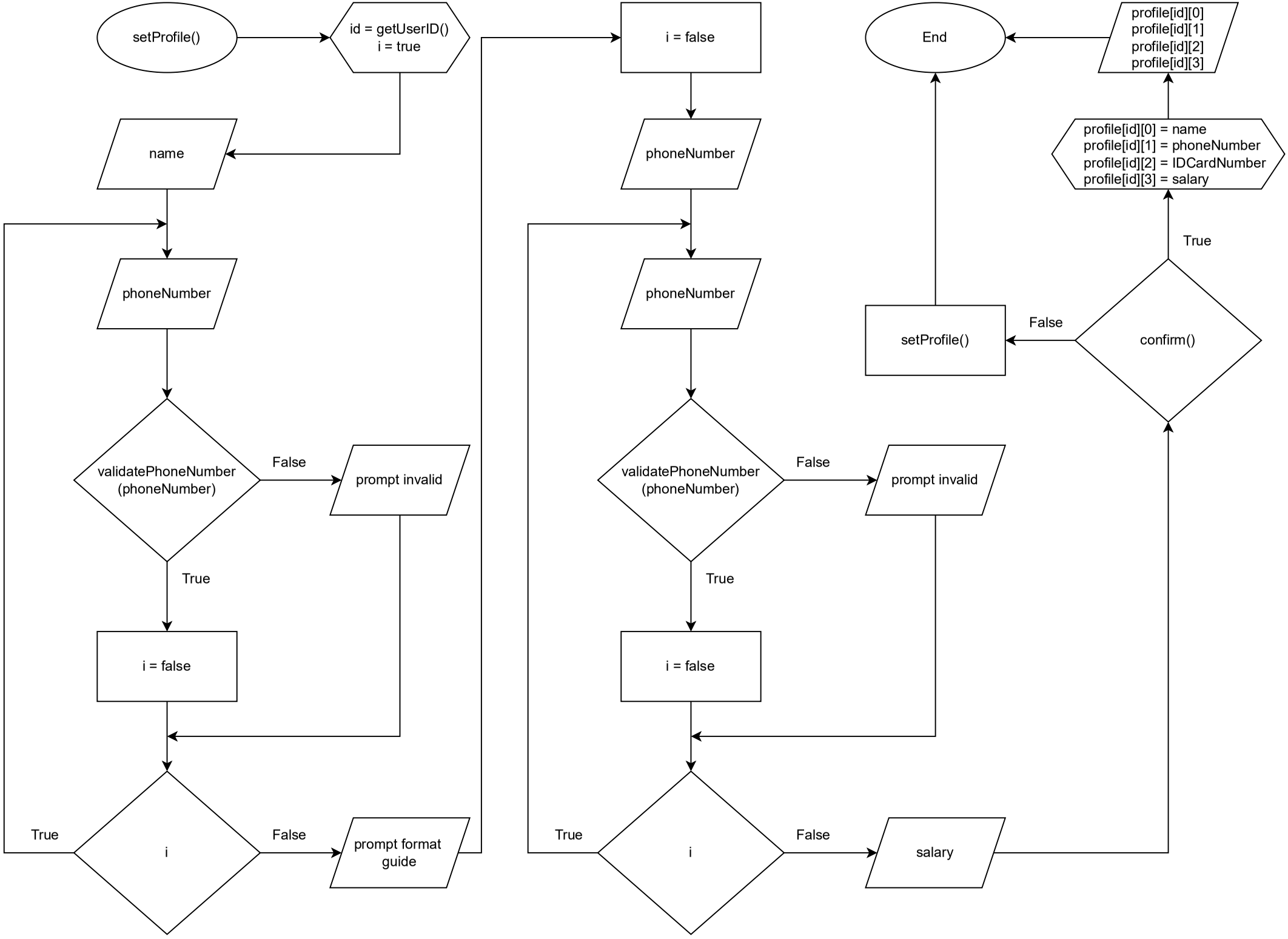

validatePhoneNumber(String numbers)
length = numbers.length()
length is less than 11 or bigger than 13
True
return false
False
return true
False
i < length
i = 0
True
current = the integer form of String form of numbers.charAt(i)
i is 0 and current is not 0
True
return false
False
i++
False
i is 1 and current is not 8
True
return false

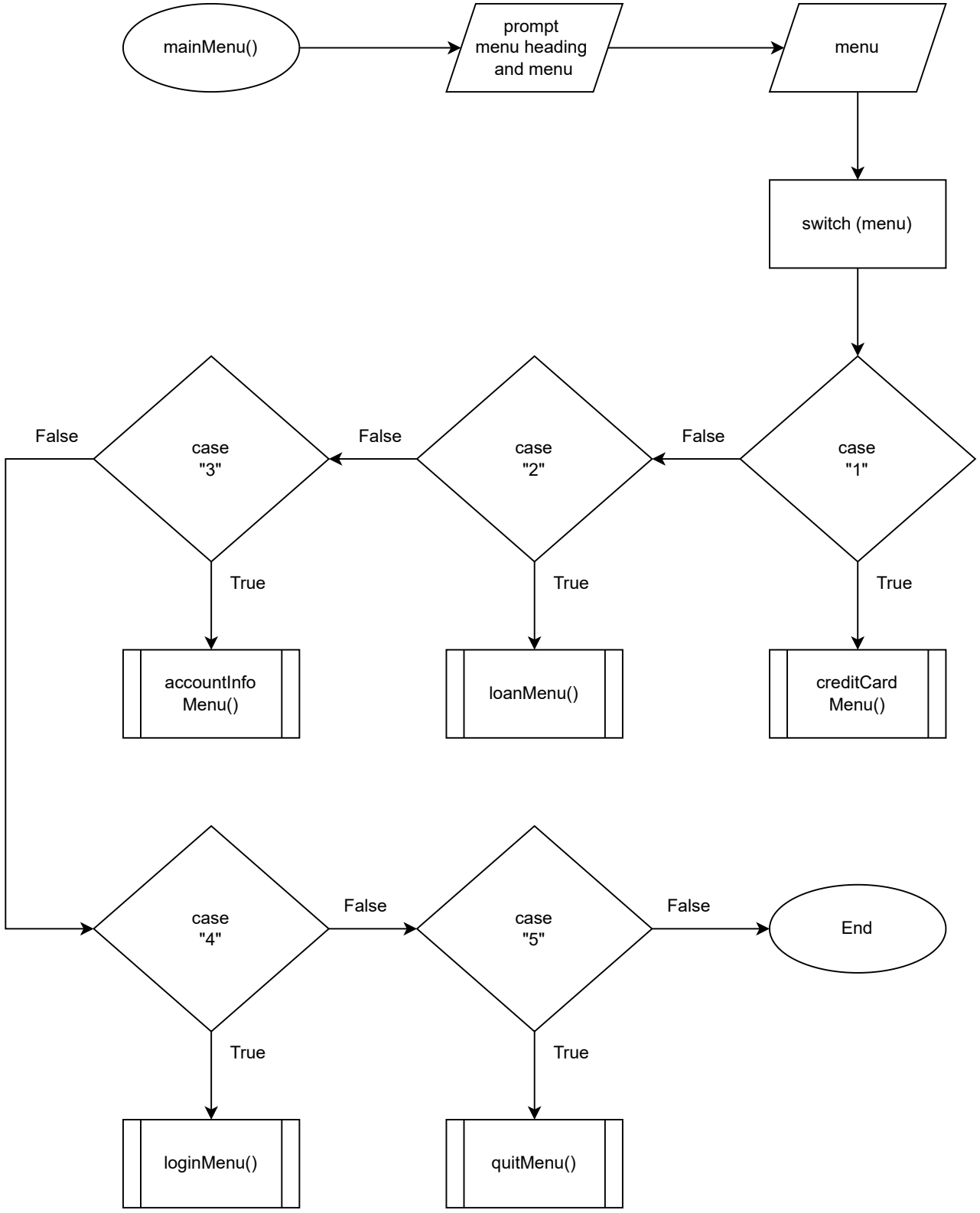
validateIDCardNumber(String numbers)
length = numbers.length()
length != 16
True
return false
False
i = 0
i < length
False
return true
i++
True
switch numbers.charAt(i)
case number
True
False
return false

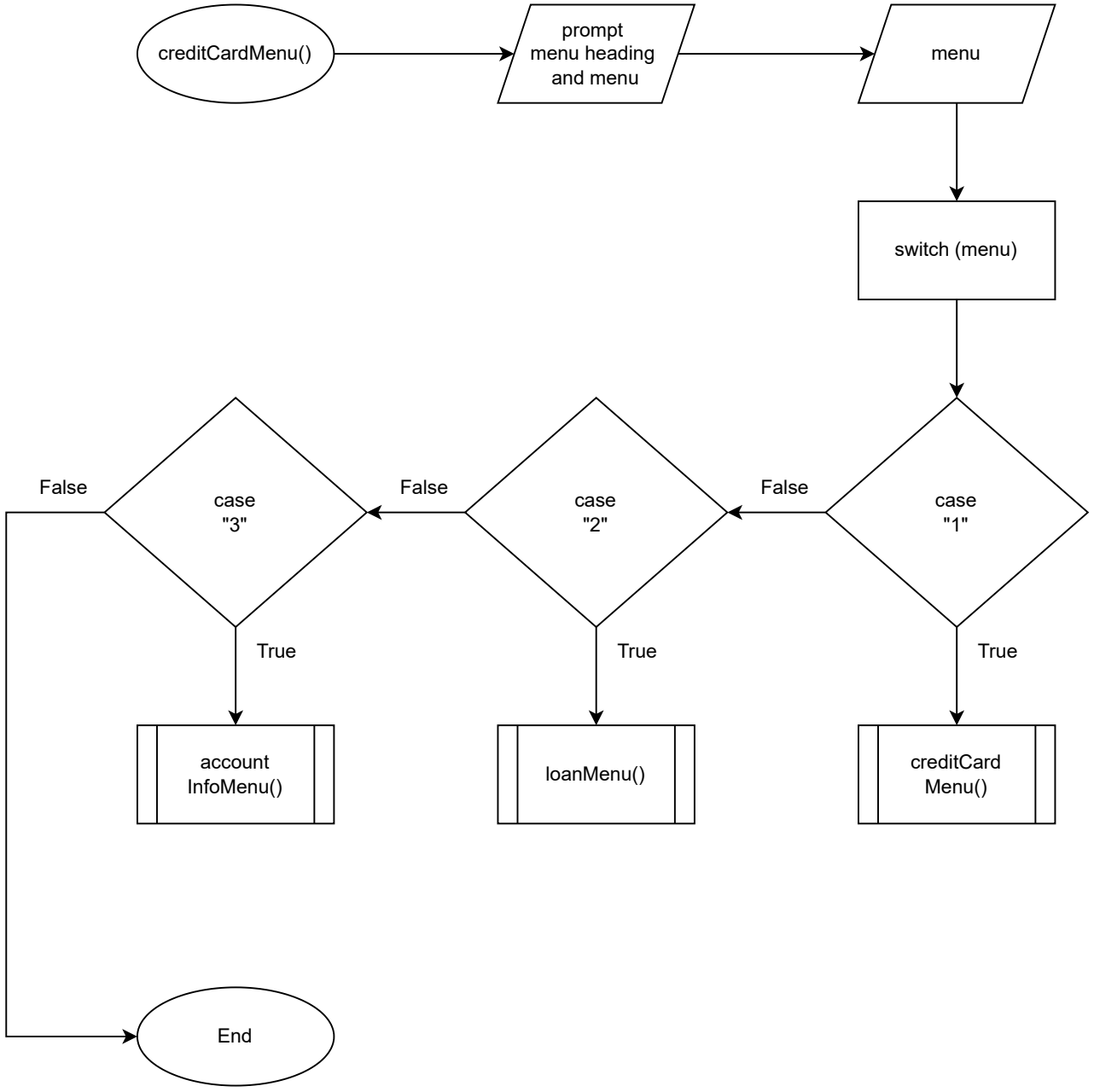

debtReadjustment(double installmentMin, double inverseReturnOfPoweredInterest, double interestInMonth, int downPayment, int downPaymentPercentage)
boolean repeat = true;
double creditLimit;
double debtMax;
int debt;
int[] out = new int[2];
creditLimit = installmentMin * (inverseReturnOfPoweredInterest / interestInMonth);
debtMax = creditLimit + downPayment;
debt
debt > debtMax
True
prompt invalid
False
repeat = false
repeat
True
False
downPayment is less than the minimum down payment
False
True
prompt the minimum down payment
repeat = true
downPayment
downPayment is less than the minimum down payment
True
prompt invalid
False
repeat = false
repeat
True
False
out[0] = debt;
out[1] = downPayment;
out = debtReadjustment(installmentMin, inverseReturnOfPoweredInterest, interestInMonth, downPayment,downPaymentPercentage)
True
downPayment is more than the minimum down payment
False

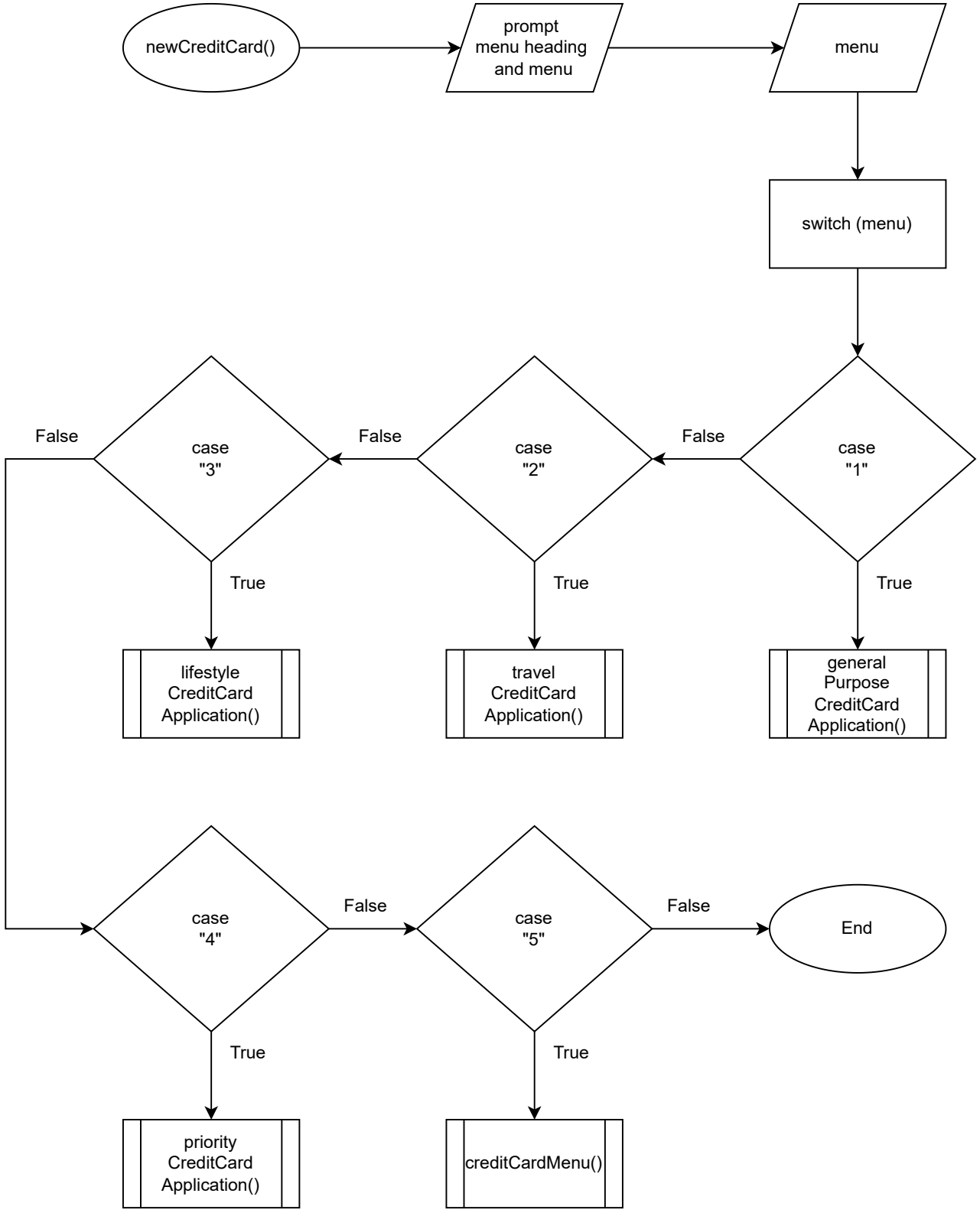

loginMenu()
prompt login
heading
and login guide
attemptLogin()
True
False
registerMenu()
True
username is
"register"
False
End
False
username is
"quit"
True
quitMenu()

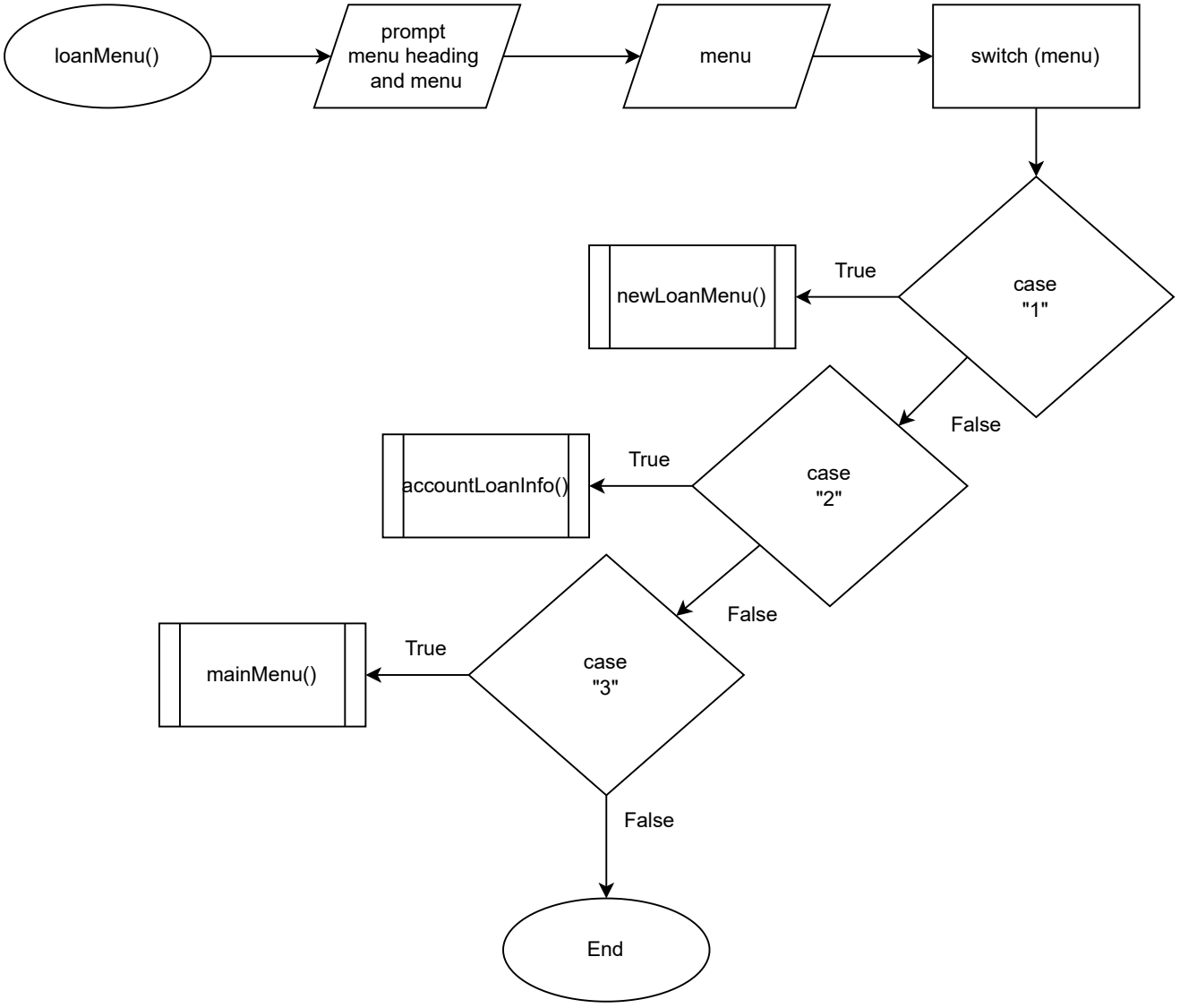

registerMenu()
prompt register heading
newStringArray(credential);
newStringArray(profile);
newDoubleArray(creditMortgage);
newString(creditMortgageDetail);
creditMortgageDetail [creditMortgageDetail.length - 1] = "";
credential [credential.length - 1][0] = promptedTextInput("Enter your username: ");
credential [credential.length - 1][1] = promptedTextInput("Enter your password: ");
loginMenu()
End

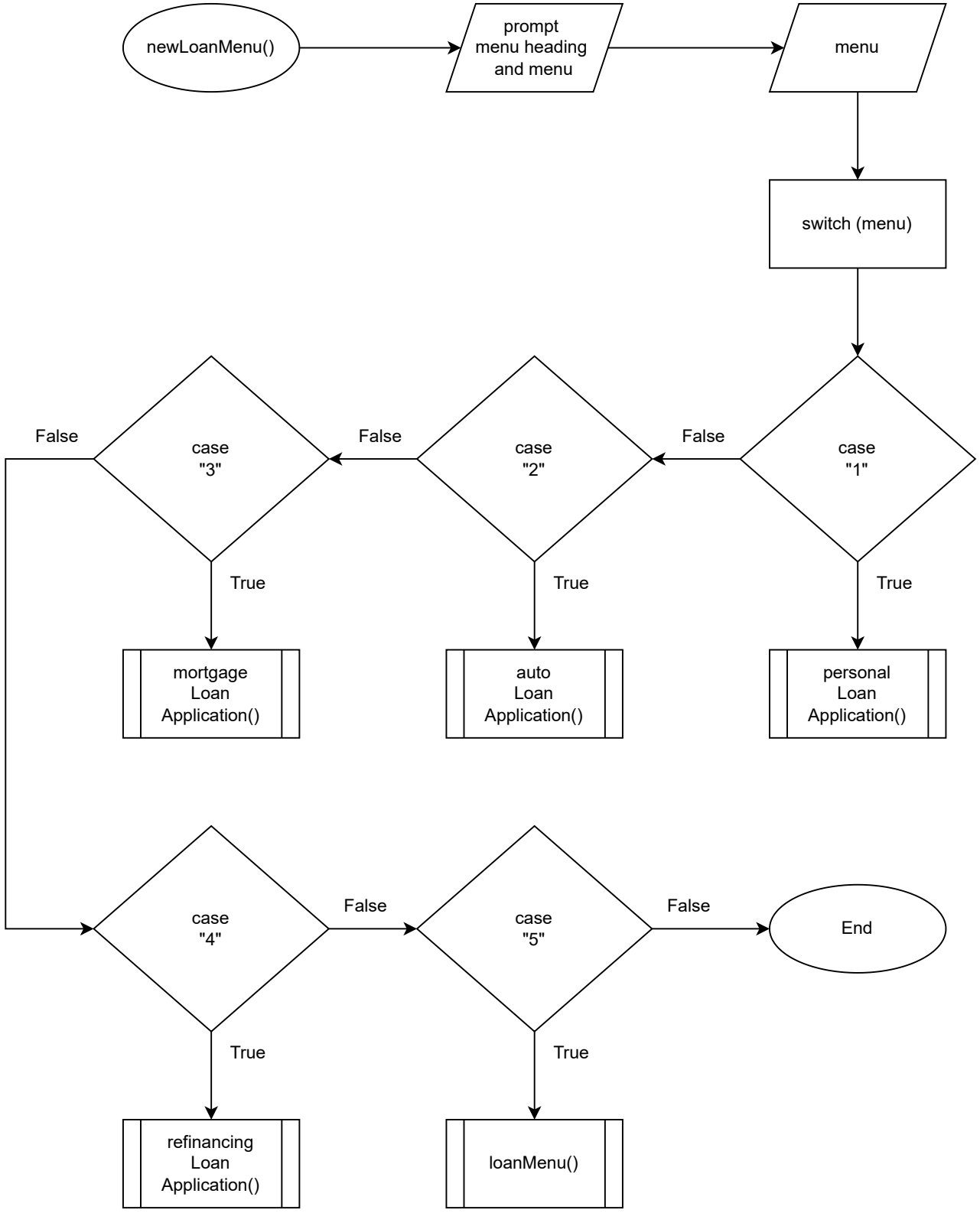

setProfile()
id = getUserID()
i = true
name
phoneNumber
validatePhoneNumber
(phoneNumber)
False
prompt invalid
True
i = false
i
True
False
prompt format
guide
i = false
phoneNumber
phoneNumber
validatePhoneNumber
(phoneNumber)
False
prompt invalid
True
i = false
i
True
False
salary
End
profile[id][0]
profile[id][1]
profile[id][2]
profile[id][3]
profile[id][0] = name
profile[id][1] = phoneNumber
profile[id][2] = IDCardNumber
profile[id][3] = salary
True
setProfile()
False
confirm()

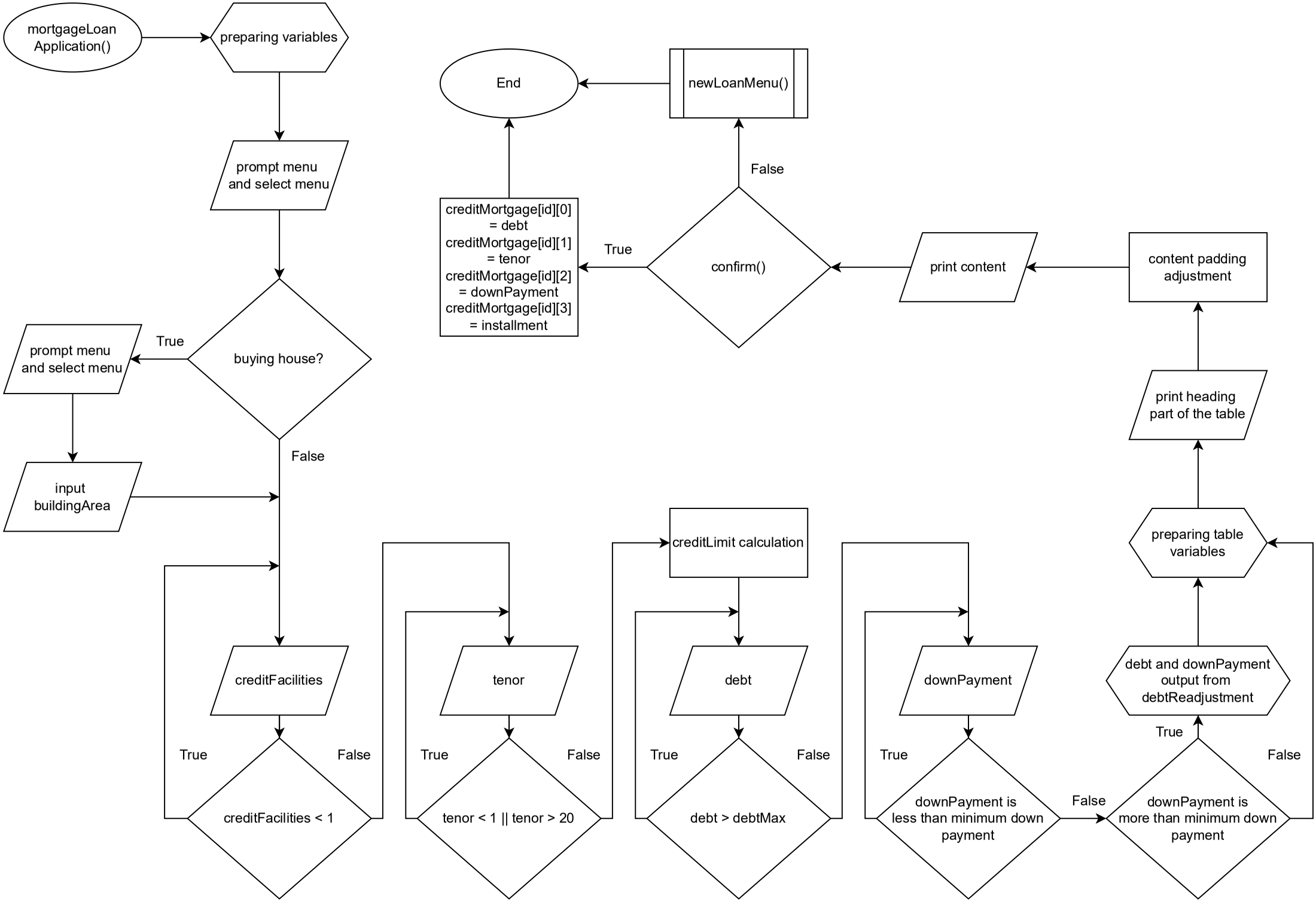

mainMenu()
prompt
menu heading
and menu
menu
switch (menu)
case
"1"
False
case
"2"
False
case
"3"
False
True
True
True
creditCard
Menu()
loanMenu()
accountInfo
Menu()
case
"4"
False
case
"5"
False
End
True
True
loginMenu()
quitMenu()


creditCardMenu()
prompt
menu heading
and menu
menu
switch (menu)
case
"1"
False
case
"2"
False
case
"3"
False
True
True
True
creditCard
Menu()
loanMenu()
account
InfoMenu()
End


newCreditCard()
prompt menu heading and menu
menu
switch (menu)
case "1"
False
case "2"
False
case "3"
False
True
general Purpose CreditCard Application()
True
travel CreditCard Application()
True
lifestyle CreditCard Application()
case "4"
False
case "5"
False
End
True
creditCardMenu()
True
priority CreditCard Application()


loanMenu()
prompt
menu heading
and menu
menu
switch (menu)
case
"1"
True
newLoanMenu()
False
case
"2"
True
accountLoanInfo()
False
case
"3"
True
mainMenu()
False
End


newLoanMenu()
prompt
menu heading
and menu
menu
switch (menu)
case
"1"
False
case
"2"
False
case
"3"
False
True
personal
Loan
Application()
True
auto
Loan
Application()
True
mortgage
Loan
Application()
case
"4"
False
case
"5"
False
End
True
loanMenu()
True
refinancing
Loan
Application()


mortgageLoan Application()
preparing variables
prompt menu and select menu
buying house?
True
prompt menu and select menu
input buildingArea
False
creditFacilities
True
False
creditFacilities < 1
tenor
True
False
tenor < 1 || tenor > 20
creditLimit calculation
debt
True
False
debt > debtMax
downPayment
True
downPayment is less than minimum down payment
False
downPayment is more than minimum down payment
True
False
debt and downPayment output from debtReadjustment
preparing table variables
print heading part of the table
content padding adjustment
print content
confirm()
True
creditMortgage[id][0] = debt
creditMortgage[id][1] = tenor
creditMortgage[id][2] = downPayment
creditMortgage[id][3] = installment
False
newLoanMenu()
End