

LABORATORY 1 : Flowchart / Pseudocode / Test cases

OBJECTIVES

- to understand how to express thoughts/algorithms in flowchart and pseudocode
- to be able to write test cases for simple programs

BACKGROUND

1. Flowchart

- <https://dyclassroom.com/flowchart/introduction>
- <https://youtu.be/vOEN65nm4YU>

2. Pseudocode

- <https://blog.usejournal.com/how-to-write-pseudocode-a-beginners-guide-29956242698>
- <https://youtu.be/r1BpraNa2Zc>

3. Whitebox testing – logic coverage

- <https://www.softwaretestinghelp.com/white-box-testing-techniques-with-example/>
- <https://www.guru99.com/code-coverage.html>
- The Art of Software Testing (book) by Glenford J. Myers (chapter 4)

LABORATORY 1: Pre-lab, In-lab, Post-lab

Work in pair

For each scenario,

1. write a flowchart
2. write a pseudocode
3. design test cases

indicate type(or types) of coverage (statement, branch, condition or path)
given by your test cases

test case	inputs	expected results	coverage
1. test case description	all inputs to the test case	expected results	line numbers in your pseudocode or path IDs in your flowchart
2.

Scenarios

1. Login attempt
 - username and password are required to login
 - only when username and password are matched, an access is granted
 - secret question is asked after the 3rd unsuccessful login attempts
 - if the answer to the secret question is correct, an access is granted and the login info (username and password) is sent to user's email.
2. Money transfer
 - transfer money from account A to account B
 - fee is charged according to the following rules
 - same bank : transferred amount > THB10,000, fee 1%
 - different bank : fee THB50 + 1%
3. Sales promotion

SAMPLE OF A SALES PROMOTION POLICY

- Preferred customers who order more than \$1,000 are entitled to a 5% discount, and an additional 5% discount if they used our charge card.
- Preferred customers who do not order more than \$1,000 receive a \$25 bonus coupon.
- All other customers receive a \$5 bonus coupon.

4. Find all pairs of numbers in a given list that sum to a given value

Example:

[1, 2, 3, 4, 5] sum = 6
result : [1, 5] , [2, 4]

5. Combine two lists by alternatingly taking elements

Example:

List 1 : [1, 2, 3]
List 2 : [a, b, c]
result : [1, a, 2, b, 3, c]

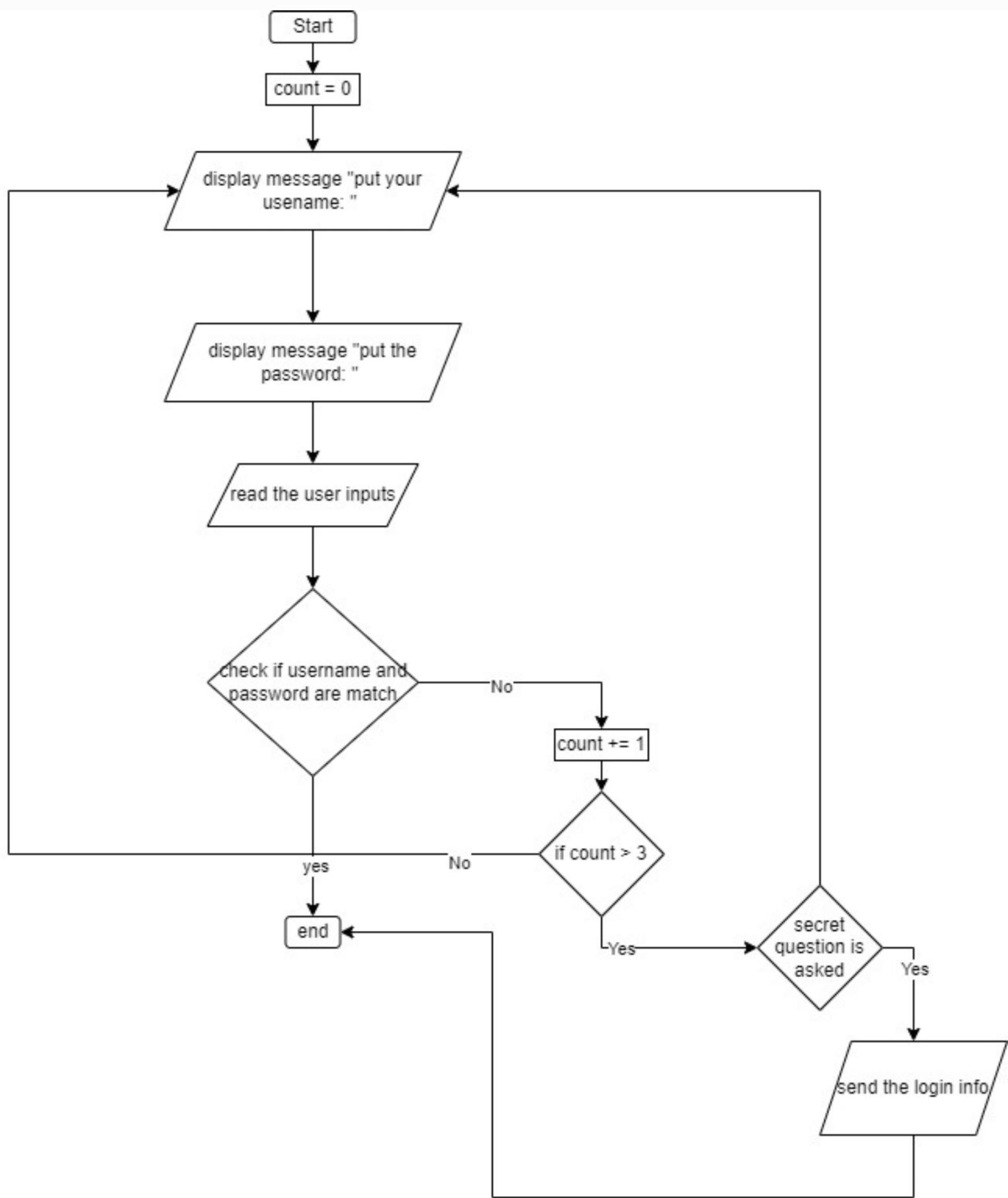
Note that list lengths may differ

Submission:

via Canvas

details are posted in Canvas.

You are to review your work with the TAs during lab session.



① Pseudocode

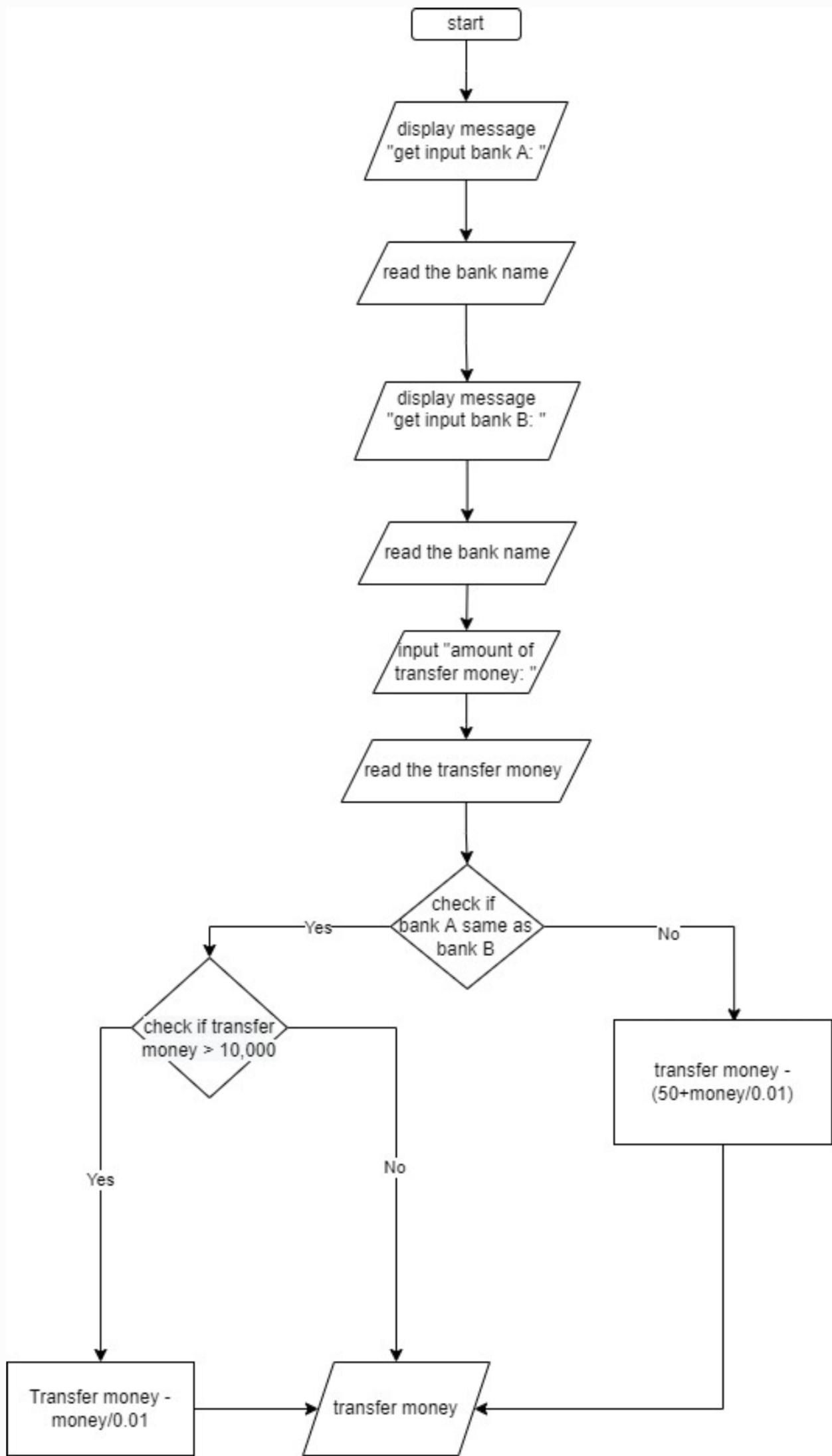
```

1 userInput = read( Input )
2 passInput = read( Input )
3 set count = 0
4 while userInput and passInput are not equal to username and password
5   count++
6   if count < 3:
7     ask secret question
8     if answer is correct:
9       send login info to user email
10    End while
11  Ask for userInput
12  Ask for passInput
13 access granted

```

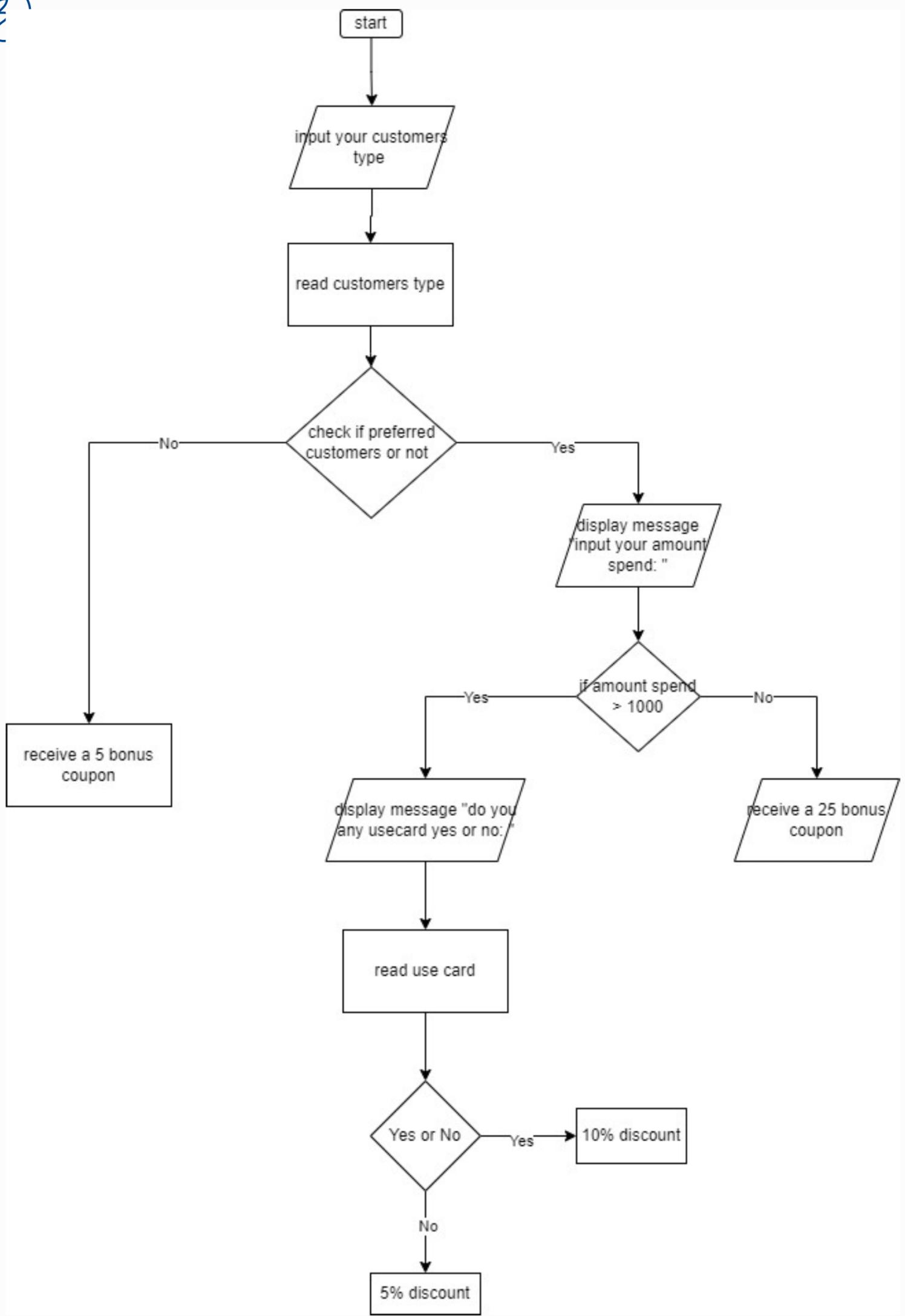
① Test Cases

Test Case	Inputs	Expected results	coverage
login with correct username and password 1st try	Correct username Correct password	access granted	1-3, 13
login with correct username and password 4th try	Correct username Wrong password 3 times Correct username Correct password	Need to answer secret question access granted	1-13
log in incorrect userinput but correct password	Wrong username Correct password	denied access ask for new inputs	4-12
log in with incorrect username and password but correct secret question	Wrong username Wrong password 3 times Correct secret question	Access granted Send user info to email	1-13



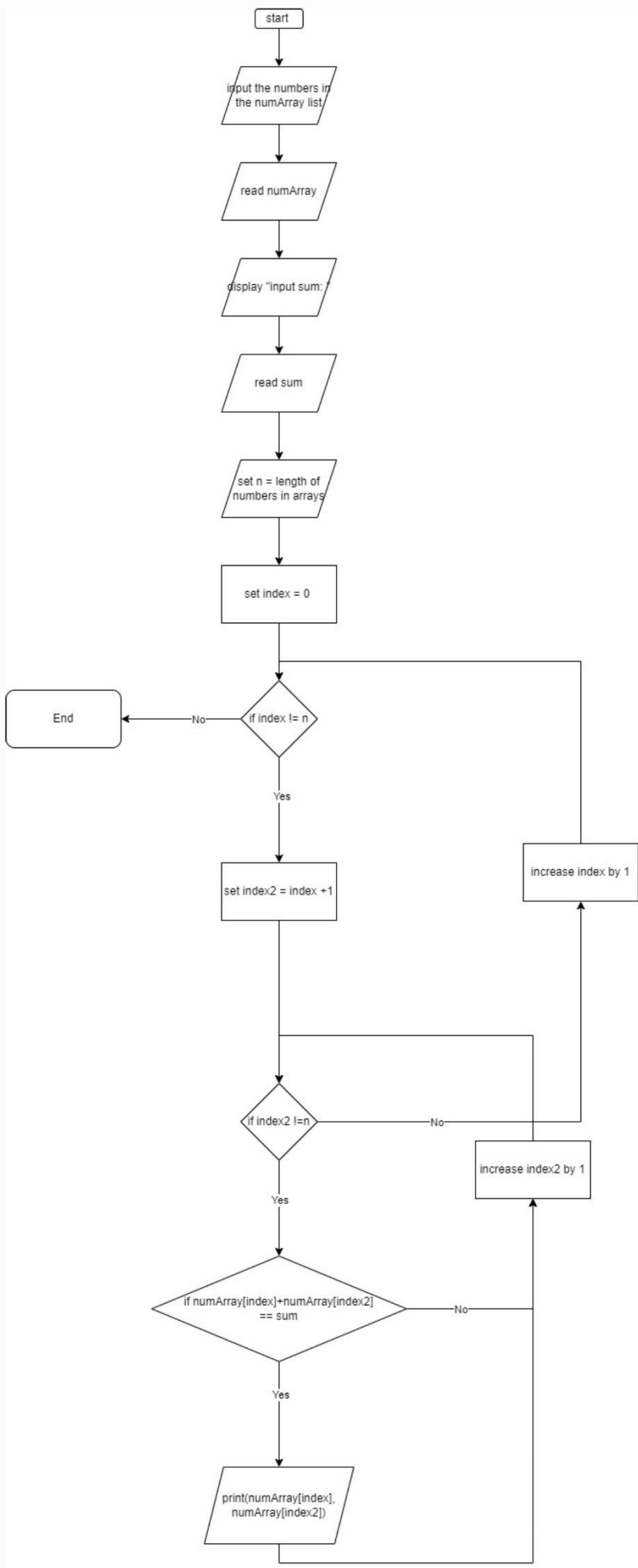
②
 1 Read bank A
 2 Read bank B
 3 Read money
 4 If bank A is same as bank B
 5 if money > 10,000
 6 transfer money - (money times 0.01)
 7 transfer money
 8 else
 9 transfer money - (50 + (money times 0.01))

Test cases	Inputs	Expected results	Coverage
Same bank with 11,000	Same bank 11,000 baht	Transfer amount 10,890 baht	1-6
Different bank with 11,000	Different bank 11,000 baht	Transfer amount 10,840 baht	1-3, 8-9
Same bank with 5000	Same bank 5,000 baht	Transfer amount 5000 baht	1-3, 7
Different bank with 5000	Different bank 5000 baht	Transfer amount 4900 baht	1-3, 8-9



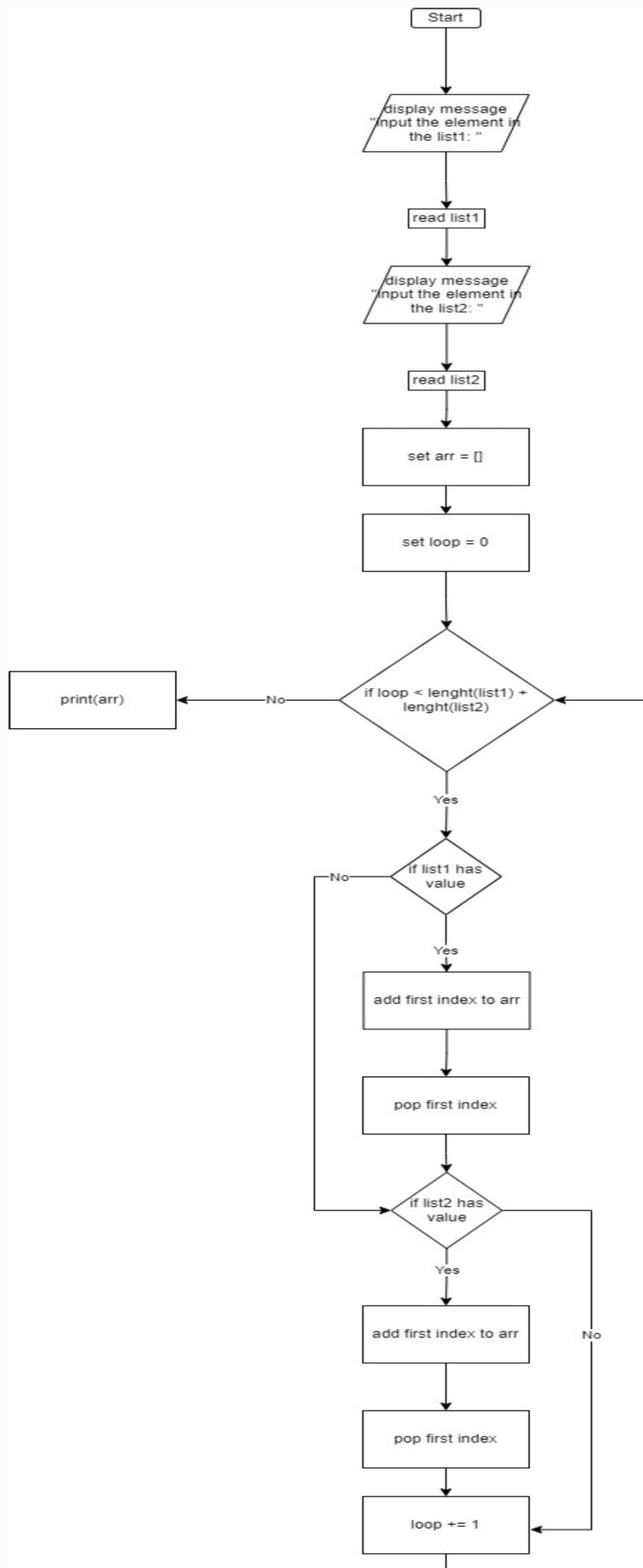
- ③ 1 read customerType
 2 if customerType is preferred:
 3 read amountSpend
 4 if amountSpend > 1000:
 5 read useCard
 6 if useCard is true:
 7 return with 10% discount
 8 return with 5% discount
 9 return with \$25 bonus coupon
 10 return with \$5 bonus coupon

Test Cases	Inputs	Expect results	Coverage
Preferred customer with order more than \$1000 but no card	Preferred \$1000 No card	5% Discount	1-5, 8
Preferred customer with order more than \$1000 but with card	Preferred \$1000 with card	10% Discount	1-7
Preferred customer with order less than \$1000	Preferred \$500	\$25 bonus coupon	1-3, 9
Not preferred customer	Not preferred	\$5 coupon	1, 10
Preferred customer with order less than \$1000 but with card	Preferred \$500	\$25 bonus coupon	1-3, 9



④ 1 read numArray
 2 read sum
 3 set n equal length of numArray
 4 for index in range(0, n):
 5 for index2 in range(index+1, n):
 6 if numArray[index] + numArray[index2] == sum:
 7 print(numArray[index], numArray[index2])

Test case	Input	Expect outputs	Coverage
Array has number that can be equal to sum	[1, 2, 3, 4, 5] 6	[1, 5], [2, 4]	1-7
Array has number that doesn't equal to sum	[1, 2, 3, 4, 5] 70	None	1-7



⑤
 1 read list 1
 2 read list 2
 3 set arr
 4 for loop in range(length(list1)+length(list2)):
 5 if list1 has value:
 6 add first index to arr
 7 pop first index
 8 if list2 has value:
 9 add first index to arr
 10 pop first index
 11 print(arr)

Test case	Inputs	Expect results	Coverage
list 1 and list 2 have same length	list 1 = [1, 2, 3] list 2 : [a, b, c]	[1, a, 2, b, 3, c]	1-11
list 1 length is longer than list 2	list 1 = [1, 2, 3, 4, 5] list 2 : [a, b, c]	[1, a, 2, b, 3, c, 4, 5]	1-11
list 2 length is longer than list 1	list 1 : [1, 2, 3] list 2: [a, b, c, d, e]	[1, a, 2, b, 3, d, e]	1-17