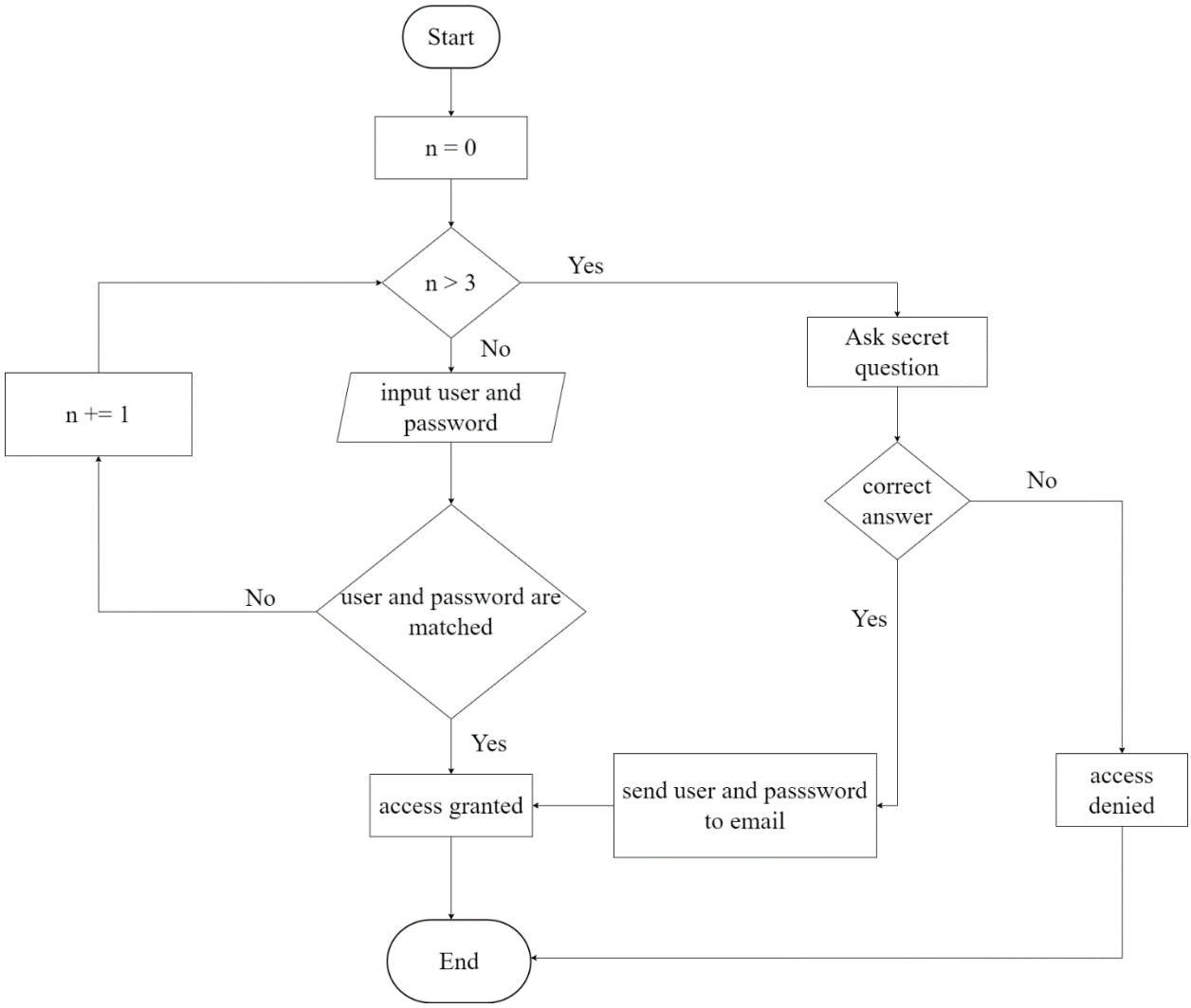
64011423 Karnpitcha Kasemsirinavin, 64011467 Natasha Greenough

Scenarios 1

- Flowchart



- Pseudocode

  SET count(n) to 0

  WHILE count(n) more than or equal to 3

  INPUT: user enter input USER and PASSWORD

  IF USER and PASSWORD are matched

  access_granted()

ELSE

Add 1 to count(n)

ENDIF

ANS = ask_secret()

IF ANS == correct_ans

access_granted()

send_email()

ELSE

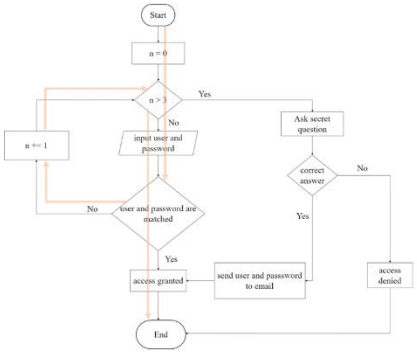
access_denied()

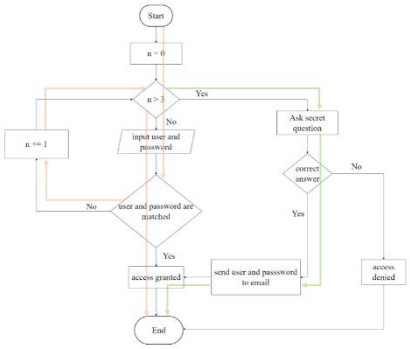
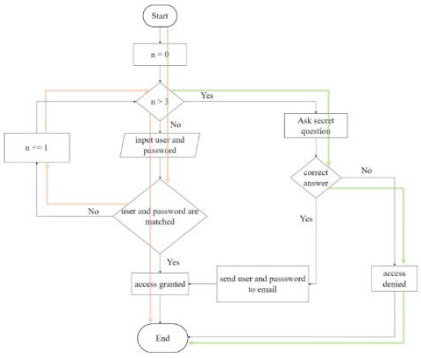
ENDIF

END

- Test case

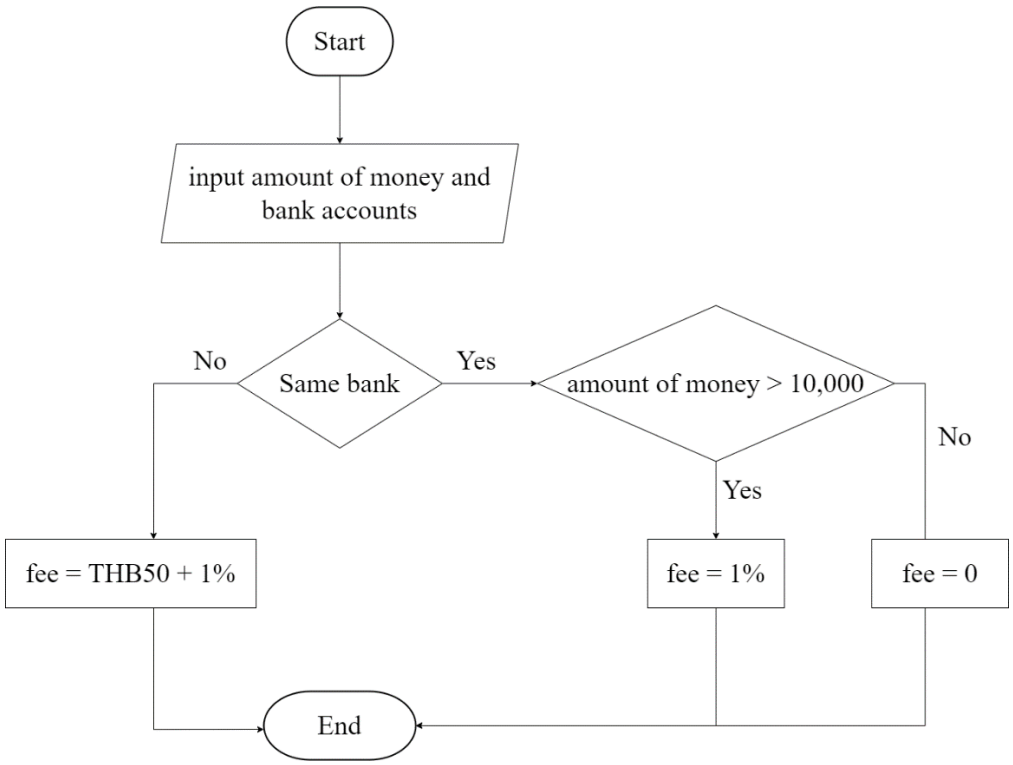
Let User = Hello, Password = 12345 and secret answer = smile

Test case	**Inputs**	**Expected result**	**Coverage**
User and password are matched within 3 times	▪ User = Hello ▪ Password = 12345	Access granted	Start → in loop n > 3 → access complete → end (Orange) 
User and password are not matched 3 times, but the answer of secret question is correct	▪ User = Hello ▪ Password = 00123 ▪ Secret answer = smile	Access granted and user's info will be shown and sent to user's email	Start → in loop for 3 times → ask secret question → access complete → receive user and password → end (Orange → Green)

			
User and password are not matched 3 times, but the answer of secret question is incorrect	▪ User = Hellp ▪ Password = 12345 ▪ Secret answer = small	Access denied	Start → in loop for 3 times → ask secret question → access denied → end (Orange → Green) 

Scenarios 2

- Flowchart



- Pseudocode

INPUT: user enter amount of money (MONEY), user bank account (BANK_ACCOUNT_1) and transferred bank account (BANK_ACCOUNT_2)

IF user bank account (BANK_ACCOUNT_1) == transferred bank account (BANK_ACCOUNT_2)

    IF amount of money (MONEY) more than 10,000

        FEE = 1% amount of money (MONEY)

    ELSE

        FEE = 0

    ENDIF

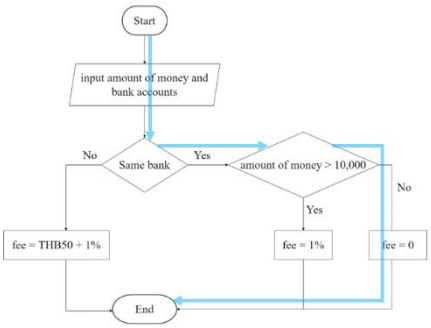
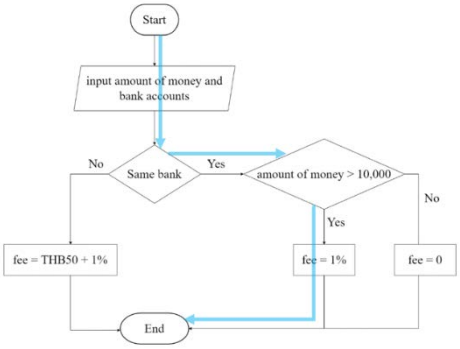
ELSE

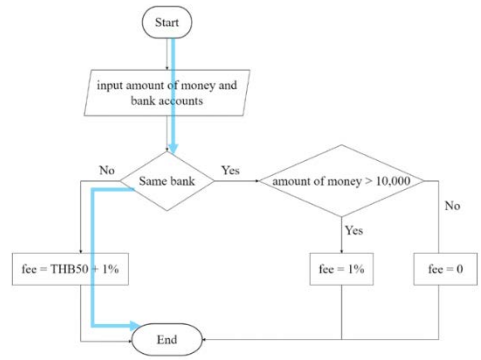
    FEE = THB50 + 1% amount of money (MONEY)

ENDIF

END

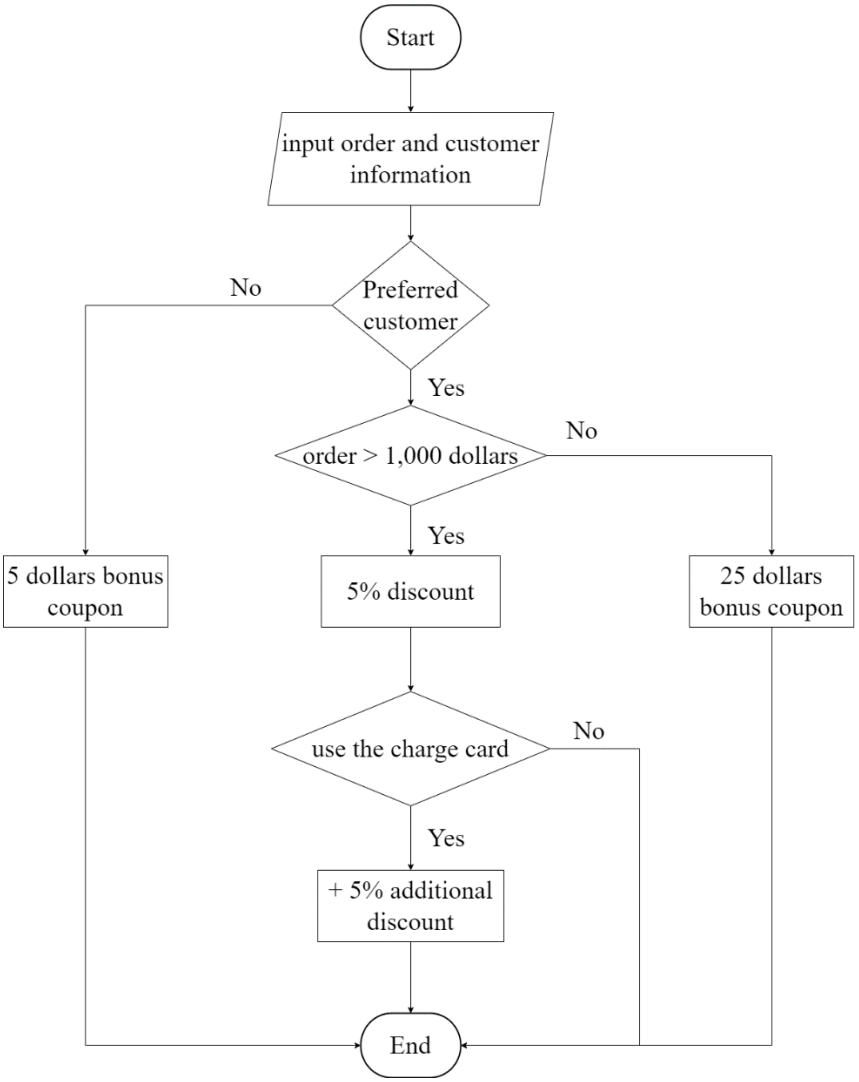
- Test case

Test case	**Inputs**	**Expected results**	**Coverage**
Same bank, but amount of money ≤ 10,000	▪ User bank account (BANK_ACCOUNT_1) = BANK A ▪ Transferred bank account (BANK_ACCOUNT_2) = BANK A ▪ Amount of money (MONEY) = 50,000	fee = 0	Start → receive inputs → check same bank → check amount of money → fee = 0 → end 
Same bank, and amount of money > 10,000	▪ User bank account (BANK_ACCOUNT_1) = BANK A ▪ Transferred bank account (BANK_ACCOUNT_2) = BANK A ▪ Amount of money (MONEY) = 8,000	fee = 1% amount of money	Start → receive inputs → check same bank → check amount of money → fee = 1% → end 
Different bank	▪ User bank account (BANK_ACCOUNT_1) = BANK A ▪ Transferred bank account (BANK_ACCOUNT_2) = BANK B ▪ Amount of money (MONEY) = 12,000	fee = THB50 + 1% amount of money	Start → receive inputs → check same bank → fee = THB50 +1% → end

Scenarios 3

- Flowchart

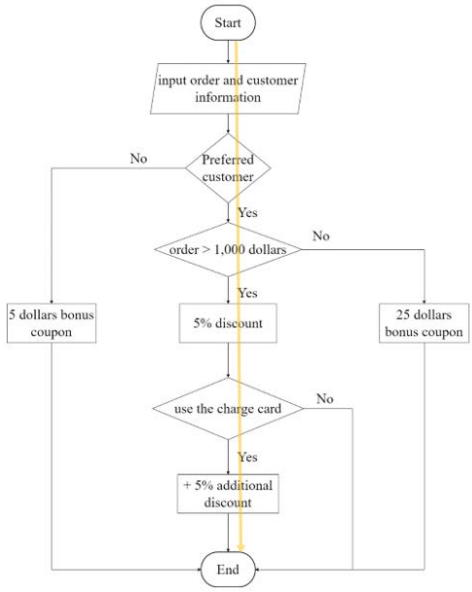
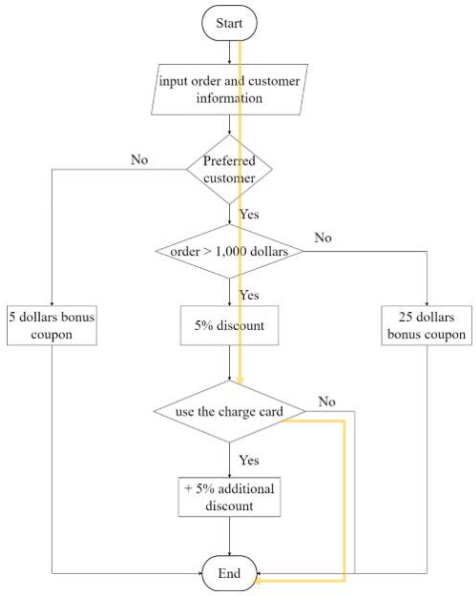


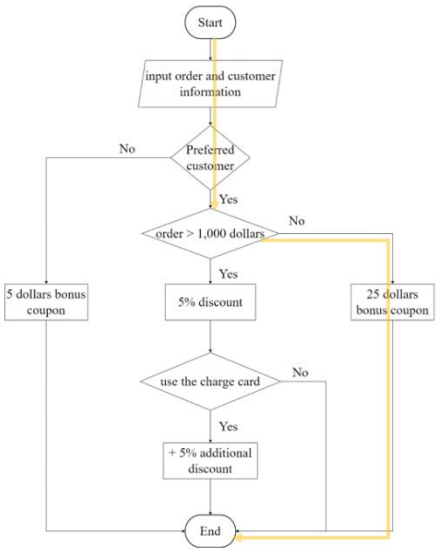
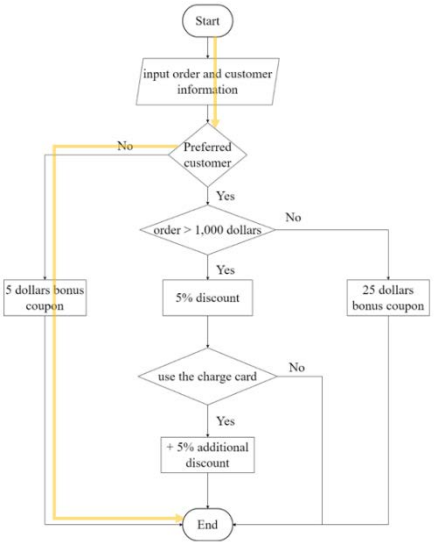
- Pseudocode

```
INPUTS: user enter order (ORDER), customer information (CUTOMER_INFO), charge card (CHARGE_CARD)

IF customer information (CUSTOMER_INFO) == preferred customer

        IF order (ORDER) more than 1,000

                DISCOUNT = 5% of order (ORDER)

                IF CHARGE_CARD == use // the customer uses charge card

                        DISCOUNT += 5% of order (ORDER)

                ENDIF

        ELSE

                25_coupon() //receive 25 dollars coupon

        ENDIF

ELSE

        5_coupon() //receive 5 dollars coupon

ENDIF

END
```

- Test case

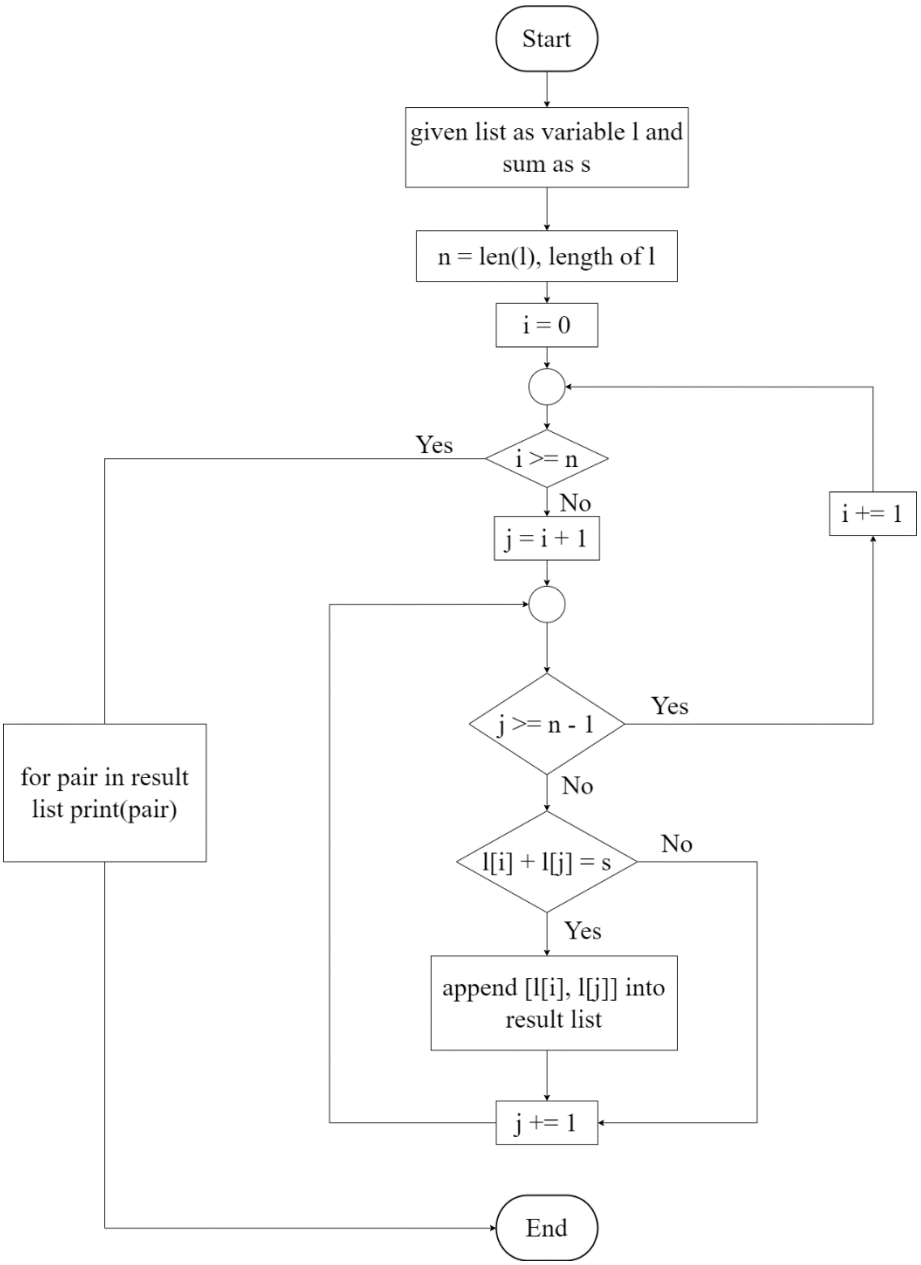
Test case	**Inputs**	**Expected results**	**Coverage**
Preferred customer, order > 1,000 dollars, and customer use charge card	▪ user enter order (ORDER) = 1,500 ▪ customer information (CUTOMER_INFO) = preferred customer ▪ charge card (CHARGE_CARD) = use	5% discount + 5% additional discount = 10% discount	Start → receive inputs → Check preferred customer → Check order amount → receive 5% discount → Check charge card → +5% additional discount → end

			
Preferred customer, order > 1,000 dollars, but customer not use charge card	▪ user enter order (ORDER) = 1,250 ▪ customer information (CUTOMER_INFO) = preferred customer ▪ charge card (CHARGE_CARD) = not use	5% discount	Start → receive inputs → Check preferred customer → Check order amount → receive 5% discount → Check charge card → end 
Preferred customer, but order ≤ 1,000 dollars	▪ customer information (CUTOMER_INFO) = preferred customer ▪ user enter order (ORDER) = 950	Receive 25 dollars bonus coupon	Start → receive inputs → Check preferred customer → receive 25 dollars bonus coupon → end

			
Other customers	▪ customer information (CUTOMER_INFO) = not preferred customer	Receive 5 dollars bonus coupon	Start → receive inputs → Check preferred customer → receive 5 dollars bonus coupon → end 

Scenarios 4

- Flowchart



- Pseudocode

  SET given list as l, given sum as s, counting position of element in list as i

  SET n = length of list

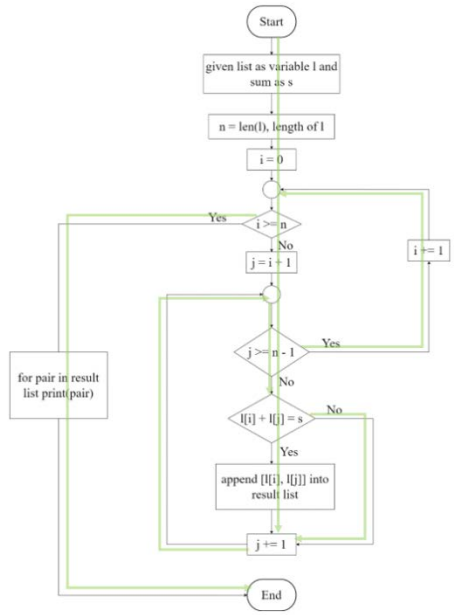
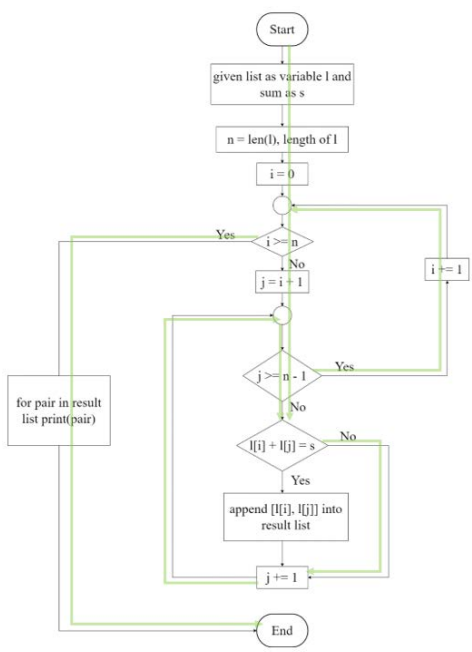
  SET i = 0

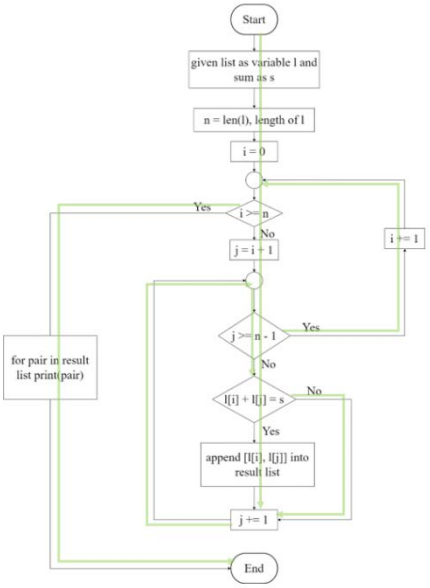
  FOR i < n

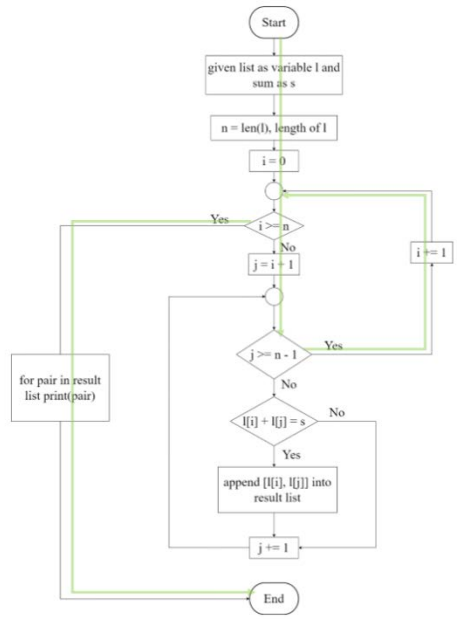
```
        j = i + 1
        FOR j < n – 1
            IF l[i] + l[j] == s
                Append [l[i], l[j]] into result list
            ENDIF
            Increase value of j by 1
        ENDFOR
        Increase value of i by 1
    ENDFOR
    PRINT pair in result list
    END
```

- Test case

Test case	**Inputs**	**Expected results**	**Coverage**
Numbers in a given list and the sum are even numbers	▪ Given list = [4,6,8,10] ▪ Sum = 10	[4,6]	Start → set variable l,s → set variable n → set variable i → Enter for loop (i < n) → set variable j → Enter another for loop (j < n – 1) → Find the sum of numbers in the given list →Append list of pair that total equal to sum → Reach the last number in list → print result → end

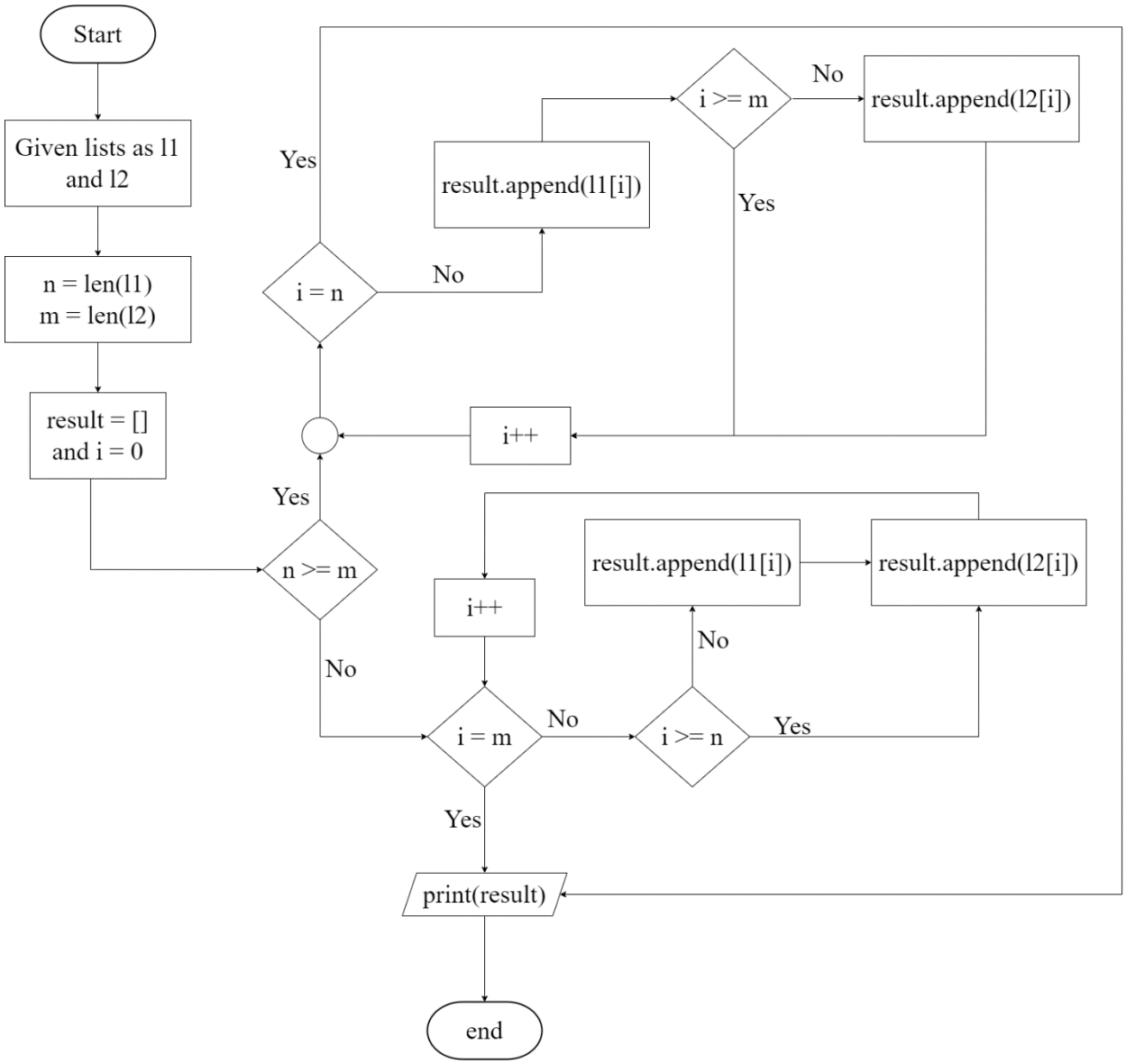
			
No pair in a given list that total equal to sum	▪ Given list = [1,3,5,7] ▪ Sum = 9	[ ]	Start → set variable l,s → set variable n → set variable i → Enter for loop (i < n) → set variable j → Enter another for loop (j < n – 1) → Find the sum of numbers in the given list → Reach the last number in list → print result → end 

Repetitive numbers in a given list	▪ Given list = [1,3,3,2] ▪ Sum = 4	[1,3], [1,3]	Start → set variable l,s → set variable n → set variable i → Enter for loop (i < n) → set variable j → Enter another for loop (j < n – 1) → Find the sum of numbers in the given list →Append list of pair that total equal to sum → Reach the last number in list → print result → end 
Only one number in a given list	▪ Given list = [1] ▪ Sum = 2	[ ]	Start → set variable l,s → set variable n → set variable i → Enter for loop (i < n) → set variable j → Enter another for loop (j < n – 1) → Reach the last number in list → print result → end

			 Start given list as variable l and sum as s n = len(l), length of l i = 0 i >= n Yes No j = i + 1 i += 1 j >= n - 1 Yes No l[i] + l[j] = s No Yes append [l[i], l[j]] into result list j += 1 for pair in result list print(pair) End

Scenarios 5

- Flowchart



- Pseudocode

SET list_1 = l1, list_2 = l2

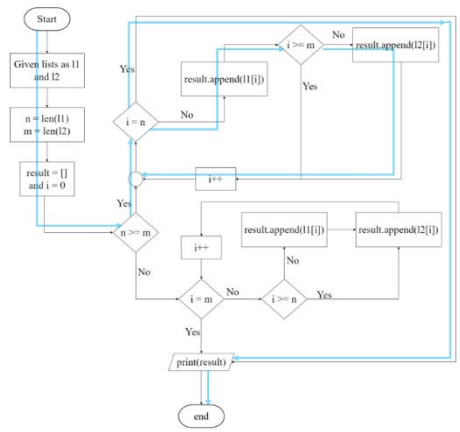
SET n = len(l1) //length of list_1

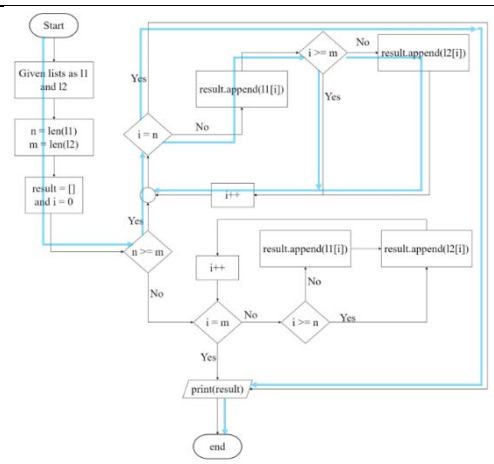
SET m = len(l2) //length of list_2

SET result = []

```
SET i = 0 //for position in list
IF length of list_1(n) >= length of list_2(m)
    FOR i < length of list_1(n)
        Append element in list_1(l1) at position i into the result
        IF i >= length of list_2(m)
            pass
        ELSE
            Append element in list_2(l2) at position i into the result
        ENDIF
        Increase value of i by 1
    ENDFOR
ELSE
    FOR i < length of list_2(m)
        IF i >= length of list_1(n)
            Append element in list_2(l2) at position i into the result
        ELSE
            Append element in list_1(l1) at position i into the result
            Append element in list_2(l2) at position i into the result
        ENDIF
        Increase value of i by 1
    ENDFOR
ENDIF
PRINT(result)
END
```

- Test case

Test case	**Inputs**	**Expected Results**	**Coverage**
The lengths of both given lists are equal	▪ Given list 1 = [1, 2, 3] ▪ Given list 2 = [b, d, e]	[1, b, 2, d, 3, e]	Start → Set lists as l1 and l2 → set n = length of list 1 and m = length of list 2 → set empty result list and i = o → Check n>= m → Enter for loop (i < n) → Append element from list 1 → Check i >= m → Append element from list 2 → increase i by 1 → Append element until reach the last element → print result list → end 
The length of list 1 > the length of list 2	▪ Given list 1 = [1, 2, 3, 4] ▪ Given list 2 = [b, d]	[1, b, 2, d, 3, 4]	Start → Set lists as l1 and l2 → set n = length of list 1 and m = length of list 2 → set empty result list and i = o → Check n>= m → Enter for loop (i < n) → Append element from list 1 → Check i >= m → Append element from list 2 → increase i by 1 → Append element until reach the last element of list 2 → Append element from list 1 only → reach last element of list 1 → print result list → end

			
The length of list 2 > the length of list 1	▪ Given list 1 = [1, 2] ▪ Given list 2 = [a, b, c, d]	[1, a, 2, b, c, d]	Start → Set lists as l1 and l2 → set n = length of list 1 and m = length of list 2 → set empty result list and i = o → Check n>= m → Enter for loop (i < m)→ Check i >= n → Append element from list 1 then list 2 → increase i by 1 → Append element until reach the last element of list 1 → Append element from list 2 only → reach last element of list 2 → print result list → end