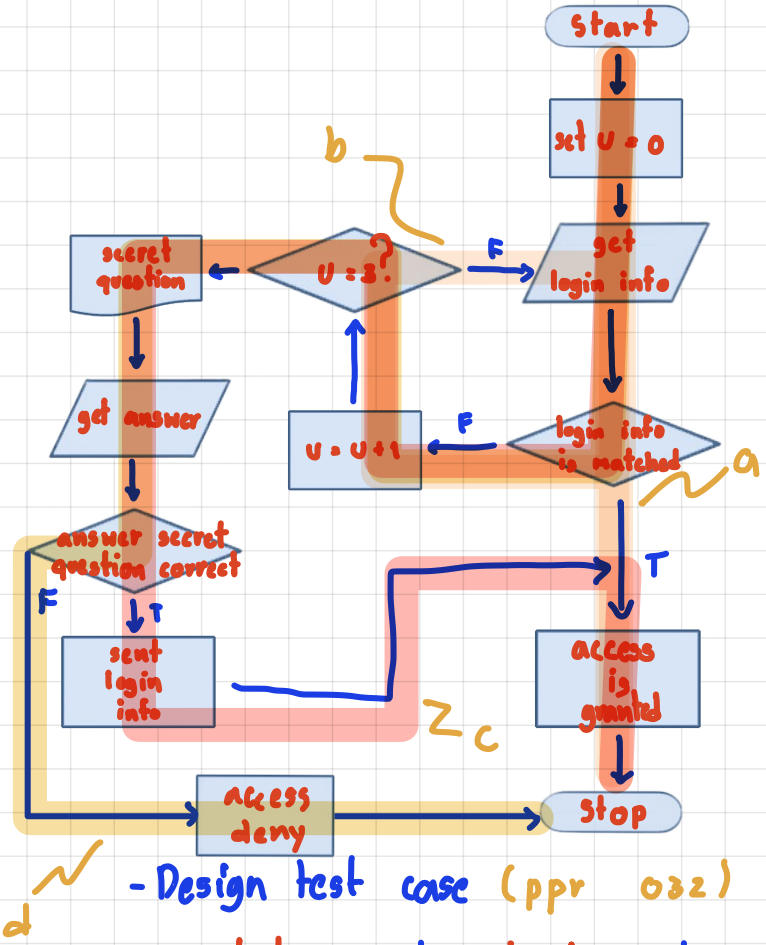
Scenarios

1. Login attempt

- Flowchart



- Pseudocode

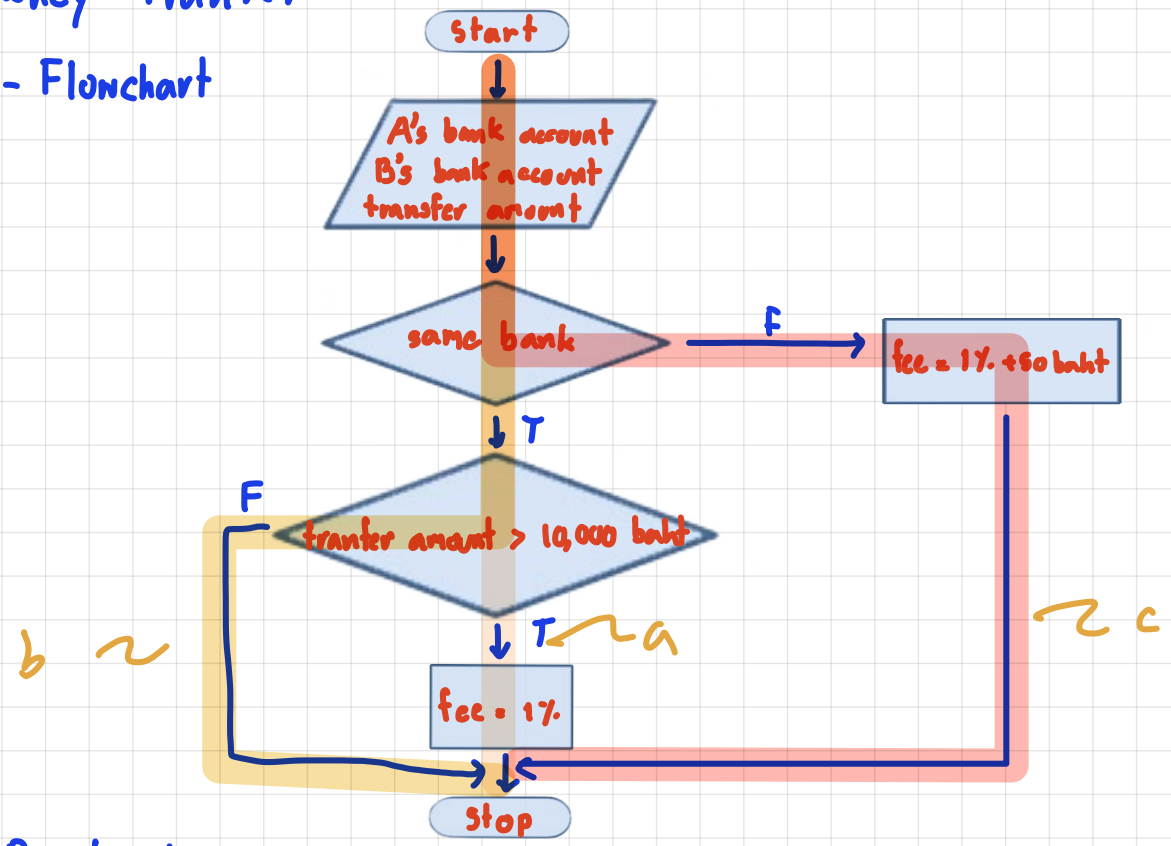
```
// Login attempt
// inputs : username , password , answer
// unsuccessful login attempt = u
Main()
1  Begin
2     set unsuccessful login attempt = 0
3     while unsuccessful login attempt ≠ 3
4        input username and password
5        if username and password are matched then
6           access is granted
7        else
8           unsuccessful login attempt + 1
9        endif
10    endwhile
11    display secret question
12    get answer secret question
13    if answer secret question correct then
14       sent login info
15       access is granted
16    endif
17 End
```

- Design test case (ppr 032)

test case	inputs	expected result	coverage
- username and password match within 3 tims	- ppr - 032	- access is granted	a b
- username and password didn't match within 3 tims and answer secret question correctly	- ppr - 000 - correct	- access is granted - sent login info to email	c
- username and password didn't match within 3 tims and answer secret question wrong	-ppr -000 -wrong	- access is deny	d

2. Money tranfer

- Flowchart



- Pseudocode

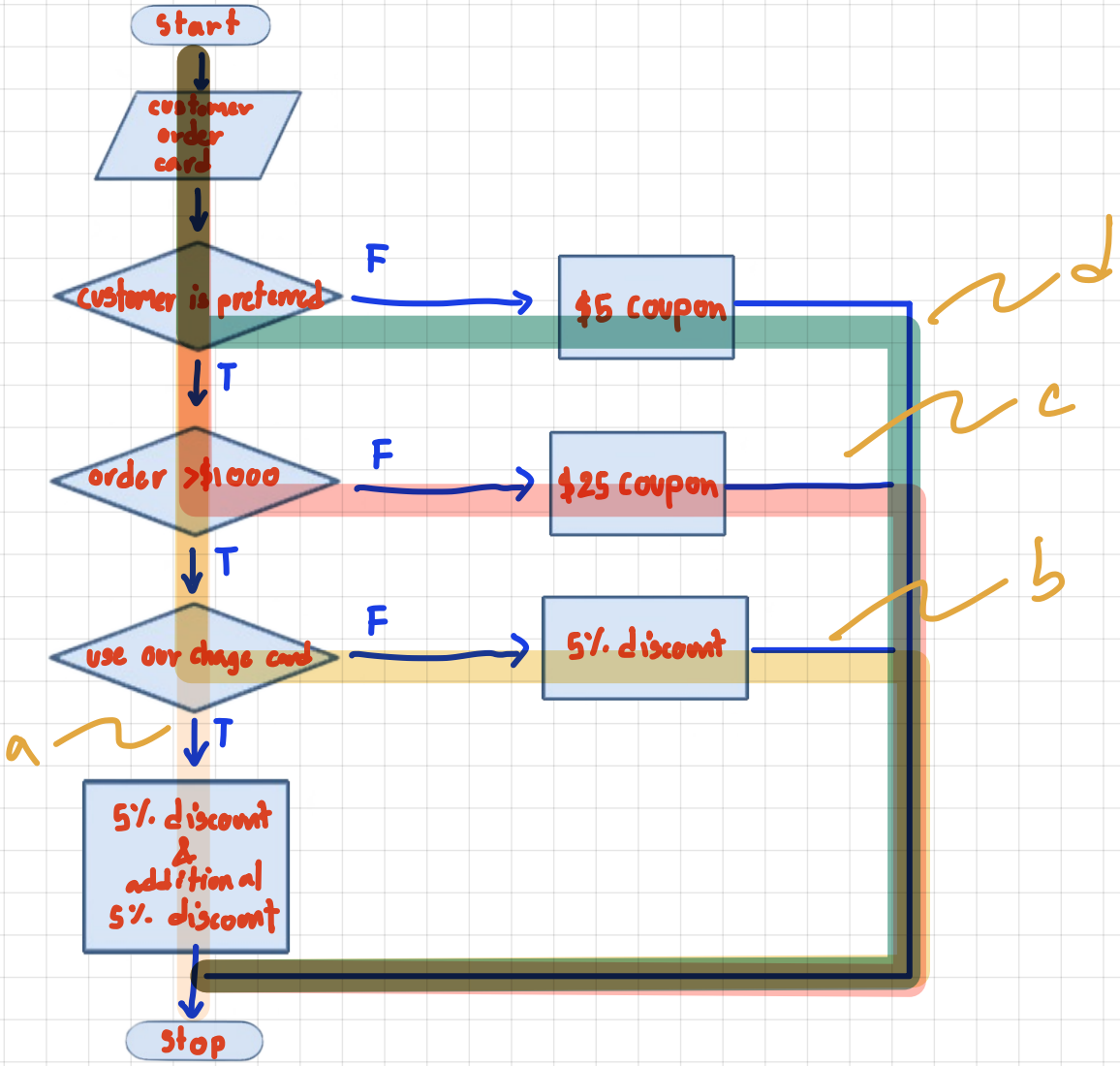
```
// Money transfer
// inputs: transfer amount , A's bank account , B's bank account
Main()
1  Begin
2     get A's bank account , B's bank account
3     input transfer amount
4     if A's bank account and B's bank account are same bank then
5        if transfer amount more than 10,000 baht then
6           fee = 1%.
7     else
8        fee = 1 % + 50 baht
9     endif
10 End
```

- Design test case

test case	inputs	expected result	coverage
- same bank > 10,000 ฿	- KBANK - KBANK - 20,000 ฿	fee = 200 ฿	a
- same bank ≤ 10,000 ฿	- SCB - SCB - 200 ฿	fee = 0 ฿	b
- different bank	- KBANK - SCB - 2000 ฿	fee = 70 ฿	c

3. Sales promotion

- Flowchart



- Pseudocode

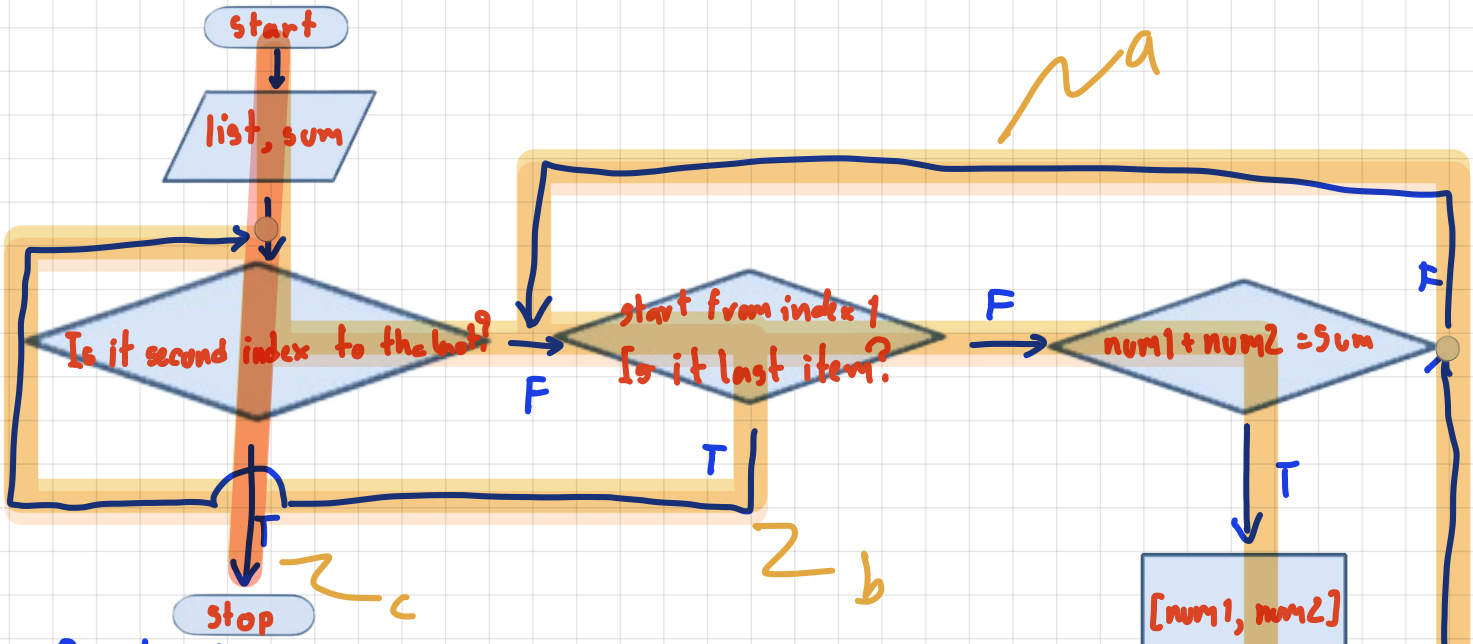
```
// Sales promotion
// inputs: customer , order, card
Main()
1  Begin
2    input customer , order, card
3    if customer is preferred customer then
4       if order more than $1000 then
5          5% discount
6          if use our charge card then
7             additional 5% discount
8          endif
9       else order not more than $1000
10            receive $25 coupon
11      end if
12   else
13      receive $5 coupon
14   endif
15 End
```

- Design test case

test case	inputs	expected result	coverage
- VIP order > 1000 our charge card	- preferred customers - $1200 - our charge card	- 10% discount	a
- VIP order > 1000 not our charge card	- preferred customers - $1200 - not our charge card	- 5% discount	b
- VIP order ≤ 1000	- preferred customers - $120	- $25 coupon	c
- non-VIP	- ordinary customers - $ 100	- $5 coupon	d

4. Find all pairs

- Flowchart



- Pseudocode

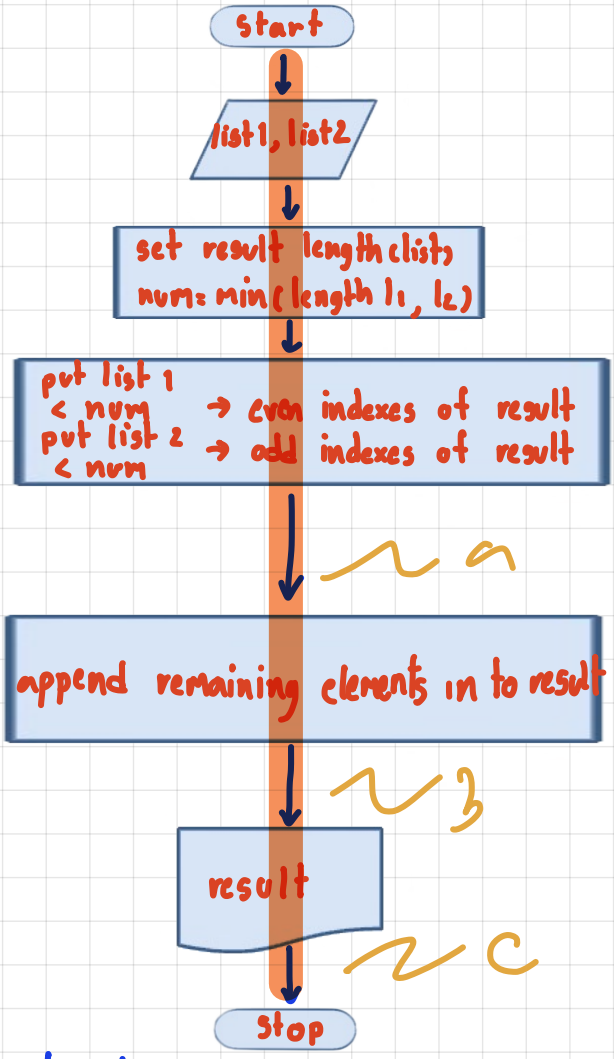
```
// Find all pair
// inputs: list, sum
Main()
1  Begin
2    input list, sum
3    for each element in list, except the last (num 1)
4      for each element in list, start from index 1 in list (num 2)
5        if sum of num1 and num2 equal to sum
6          print [num1, num2]
7          del num1 and num2
8        end if
9      end for
10   end for
11 End
```

- Design test case

test case	inputs	expected result	coverage
- found a pairs In list no duplicate number	- [1,2,3,4,5] - sum = 6	- [1,5], [2,4]	a
- found a pairs In list has duplicate numbers	- [1,2,2,4] - sum = 6	- [2,4]	b
- can't find a pair	- [1,2,3] - sum = 9	- []	c

5. Combine two lists

- Flowchart



- Pseudocode

```
// Combine two lists
// inputs: list1, list2
Main()
1  Begin
2    input list1, list2
3    set result = list that has length equal to sum of length list1 and list2
4    set num = minimum indexes
5    put the items in list1 (index <num) in to even indexes of result
6    put the items in list2 (index <num) in to odd indexes of result
7    append remaining elements in to result
8    print result
9    endif
10 End
```

- Design test case

test case	inputs	expected result	coverage
- list 1 is the same length list 2	- [1,2,3] - [a,b,c]	[1,a,2,b,3,c]	a
- list 1 shorter than list 2	- [1,2,3] - [a,b,c,d]	[a,1,b,2,c,3,d]	b
- list 1 shorter than list 2	- [1,2,3,9] - [a,b,c]	[1,a,2,b,3,c,9]	c