Question 1 Answer　　　　　　　　　　　**Student I.D. No.:**　　　　　1155116317

(1) ①_____Budget_____; ②_____project_____; ③_____quality_____.

(2) ④_____what_____; ⑤_____how_____.

(3) ⑥_____ user-friendliness _____; ⑦_____ understandability _____; ⑧_____ visibility _____;

(4) ⑨_____external_____; ⑩_____internal_____.

(5) ⑪_____corrective_____; ⑫_____adaptive_____; ⑬_____perfective_____.

(6) ⑭___separation of concerns_____; ⑮_____incrementality_____.

(7) ⑯_____ separation of concerns_____; ⑰___generality_____;⑱___Abstraction_____.

(8) ⑲_____cohesion_____; ⑳_____coupling_____.

Question 2 Answer　　　　　　　　　　**Student I.D. No.:** ____1155116317____

(1) ①____informal_____; ②____semi-formal_____; ③_____formal_____.

(2) ④____top-down_____; ⑤_____incrementyality_____.

(3) ⑥_____; ⑦_____.

(4) ⑧_____inherutance_____; ⑨___association_____;
⑩____aggregation_____.

(5) ⑪_____; ⑫_____; ⑬_____.

(6) ⑭___C1_____; ⑮_____C1,C5_____; ⑯_____true_____.

(7) ⑰____11.65hour_____; ⑱_____97.08%_____.

Question 3 Answer   **Student I.D. No.:**   1155116317

(1)

P1 : Buyer_arrived

T1

P2 : 1_USD_Inserted

T2

P3 : Dispensing_Cola

T3

P4 : Remain_Can_checked

T6   T7

T4:still_have_stock   T5: all_sold

P5: refilled_machines   P6: wait_machine_maintainer   P6: refilled_6cans
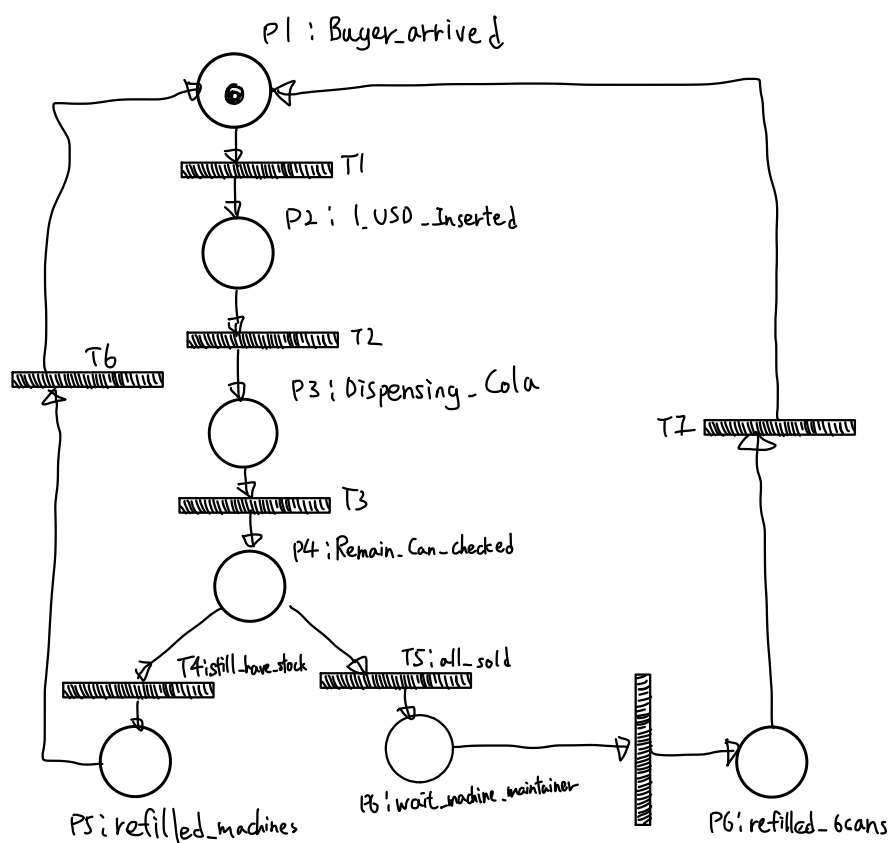
Question 3 Answer　　　　　　　　　**Student I.D. No.:**　　　　1155116317

(2)

Question 4 Answer   **Student I.D. No.:** _____1155116317_____

(1) ①_____i1>i2_____; ②_____i1_____;

③_____1_____; ④_____i_____;

⑤_____a[j]_____; ⑥_____ leftMax_____;

⑦_____ i_____; ⑧_____n_____;

⑨_____ a[k]_____; ⑩_____ rightMax_____.

⑪_____ min(leftMax, rightMax) - a[i]_____.

(2) ①_____l < r_____; ②_____a[l]_____;

③_____leftMax - a[l]_____; ④_____l+1_____;

⑤_____a[r]_____; ⑥_____rightMax - a[r]_____;

⑦_____r-1_____;

Question 5 Answer   **Student I.D. No.:**   1155116317

① _____expression[i]_____;

② _____expression[i]_____;

③ _____right_____;

④ _____left_____;

⑤ _____op_____;

⑥ _____operand_stack_____;

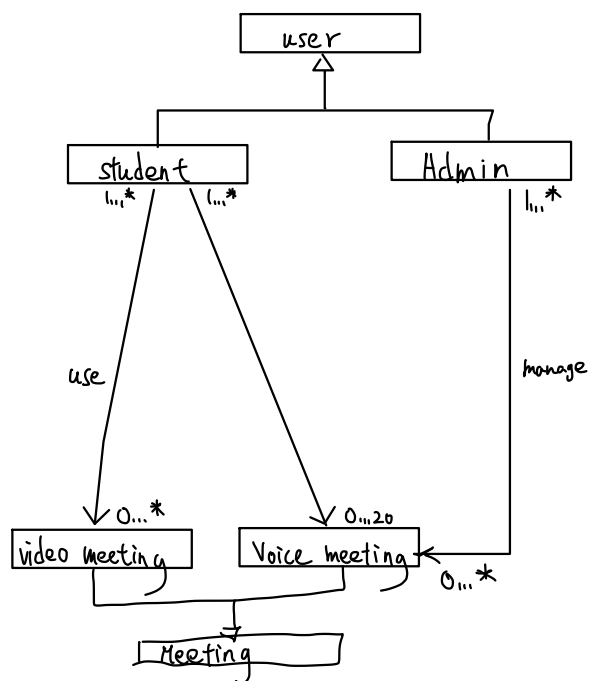⑦ _____operate(left, right, op)_____;

⑧ _____operand_stack.top()_____.

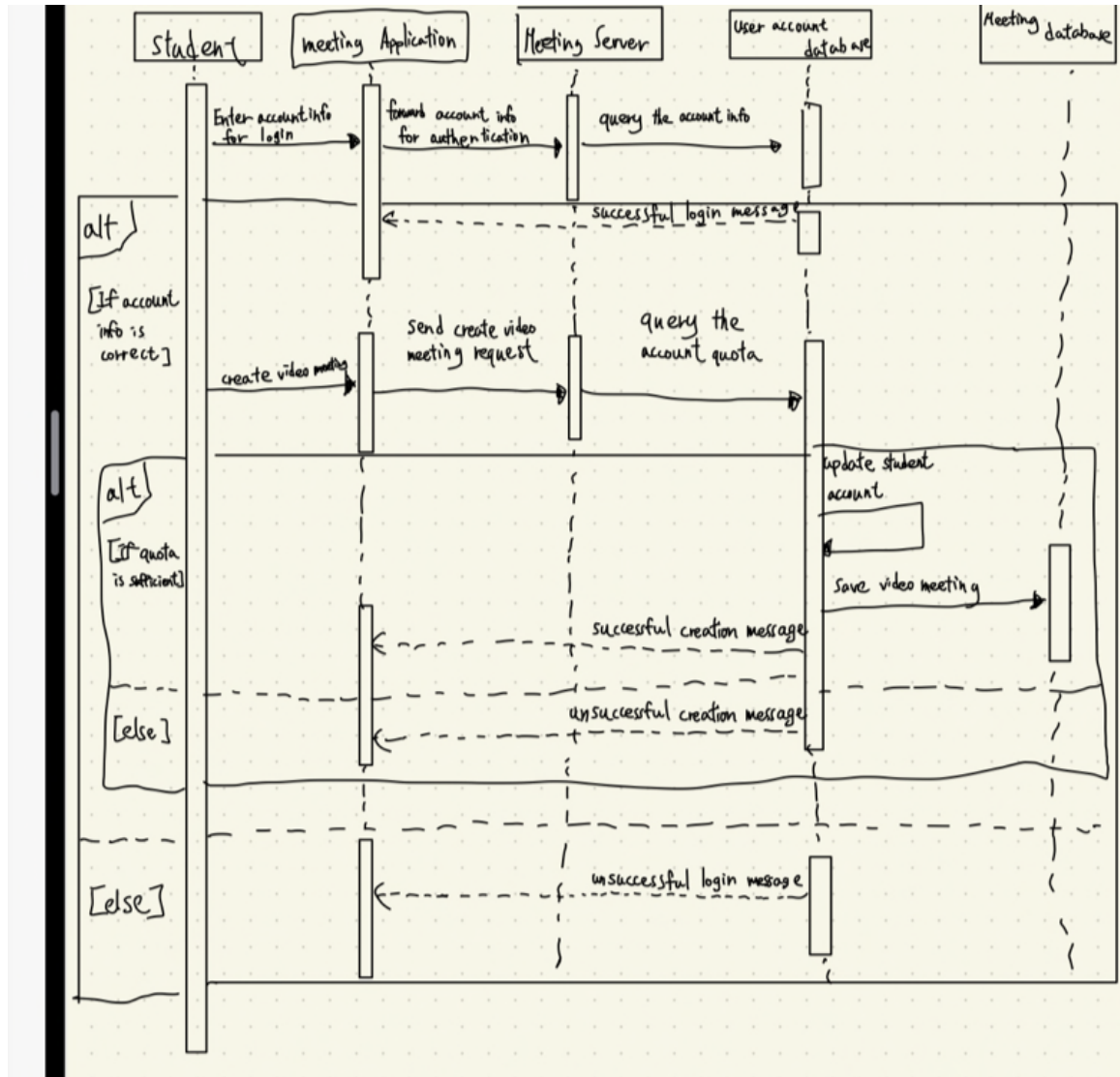**Question 6 Answer**　　　　　　　　　**Student I.D. No.:**　　　　1155116317

(1)

Question 6 Answer   **Student I.D. No.:   1155116317_____**

(2)

Question 7 Answer   **Student I.D. No.:**   1155116317

(1)

COUPON = 0.

BASE_COUPON_PERCENT=1.

DISTANCE_COUPON_PERCENT=0

IF FINISH = TRUE

    DISTANCE_COUPON_PERCENT= DISTANCE_COUPON_PERCENT + 0.1.

IF OBESE = TRUE

    DISTANCE_COUPON_PERCENT= DISTANCE_COUPON_PERCENT + 0.05.

    BASE_COUPON_PERCENT= BASE_COUPON_PERCENT + 0.1

IF AGE >=10

    COUPON = COUPON + 100.

    DISTANCE_COUPON_PERCENT= DISTANCE_COUPON_PERCENT+0.1.

    ELSE

    GO TO CALUCATION

IF AGE >=20

    COUPON = COUPON + 100.

    DISTANCE_COUPON_PERCENT= DISTANCE_COUPON_PERCENT+0.05

    ELSE

    GO TO CALUCATION

IF AGE >=30

    COUPON = COUPON + 100.

    DISTANCE_COUPON_PERCENT= DISTANCE_COUPON_PERCENT+0.05

    ELSE

    GO TO CALUCATION

IF AGE >=40

    COUPON = COUPON + 100.

    DISTANCE_COUPON_PERCENT= DISTANCE_COUPON_PERCENT+0.05

    ELSE

    GO TO CALUCATION

IF AGE >=50

    COUPON = COUPON + 100.

    DISTANCE_COUPON_PERCENT= DISTANCE_COUPON_PERCENT+0.05

ELSE

GO TO CALUCATION

CALUCATION: COUPON = COUPON * BASE_COUPON_PERCENT + ACH_DISTANCE * DISTANCE_COUPON_PERCENT
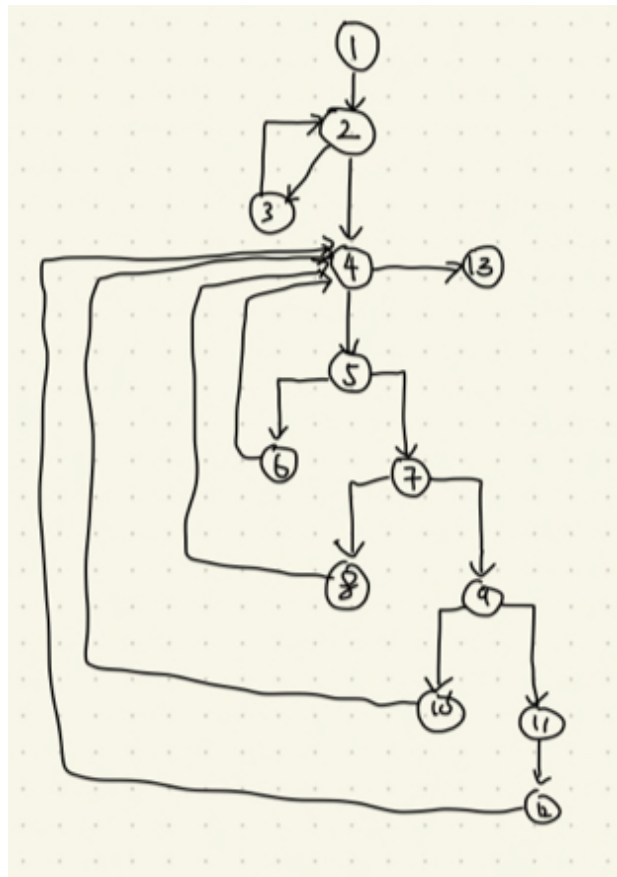
END

(2)

My program is write like stepwise statement which is relatively easy to write and can do top down testing because each part of code belong to one policy to the marathon.

Question 8 Answer　　　　　　　　**Student I.D. No.:**　　　　1155116317

(1)

(2)

(2) Predicate Node : 5

regions : 5

$V[G] = E - N + 2 = P + 1 = 6$

Question 8 Answer　　　　　　　　**Student I.D. No.:**　　**1155116317_____**

(3)

(3) 1, 2, 4, 5, 6, 4, 13

1, 2, 3, 4, 5, 6, 4, 13

1 2 4 5 7 8 4, 13

1 2 3 4 5 7 8 4　13

1 2 4 5 7 9 10　4　13

1 2 3 4 5 7 9 10　4 13

1 2 4 5 7 9 11 12 4 13

1 2 3 4 5 7 9 11 12 4 13

(4)

(4) test set

(36 , 8)

(-10 , 8)

(5) (36, 8)

(-10, 8)

(5)

Question 9 Answer   **Student I.D. No.:   1155116317_____**

strategy	
arithmetic expression with integer	1+2, 3-4, 5*6
arithmetic expression with float	1.1+2.2, 3.3-4.4,5.5*6.6
arithmetic expression with integer and float	1+2.2, 3.3-4, 5.5*6
arithmetic expression with priority (/ +- vs */)	1/2+3*4
arithmetic expression with division (output exist)	1/2, 3/4
arithmetic expression with division (output non exist)	1/0
arithmetic expression with parenthesized unitary operators	(+1)+(-2.2)
arithmetic expression with multiple different blanket	{([1+2]*3)/4}
arithmetic expression with priority (blanket + operator)	1/(2+3)*4