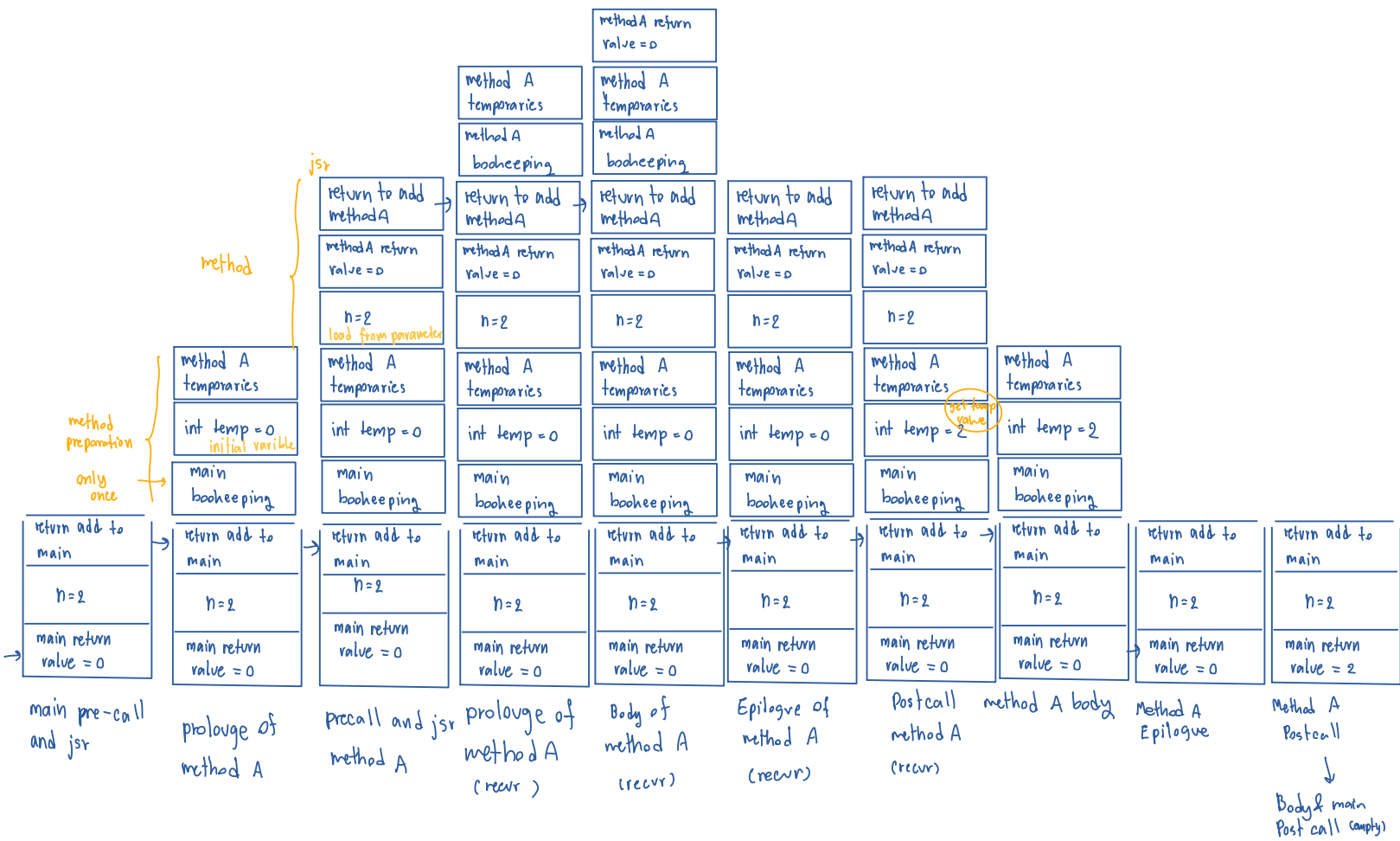


1. (10 คะแนน) มีโปรแกรมหนึ่งมีโค้ดดังนี้

```
int main(){
    return methodA(2);
}

int methodA(int n){
    if(n <= 0){
        return 0;
    }
    int temp;
    temp = n + methodA(n-2);
    return temp;
}
```

จงวาดสแตคในหน่วยความจำแสดงสิ่งที่เกิดจากการเรียกเมธอดแต่ละครั้ง จนถึงขณะที่เมธอดรัน **methodA(0)** เสร็จแล้วกำลังจะรีเทิร์น วาดรายละเอียดให้ได้มากที่สุดเท่าที่จะวาดได้



2. (10 คะแนน) มีโค้ดเมธอดของภาษาที่มี short-circuit Boolean evaluation ดังนี้

```
int f(int a, int b, int c, int d, int e, int f) {  
    int result = 0;  
    if( (a>b || c>d) && b != c) {  
        for(int i=1; i<=f; i++)  
            result = result + f;  
    }  
    return result;  
}
```

ถ้าภาษานี้ไม่มี short-circuit แต่เราต้องการให้การเช็คและรันเหมือนกับภาษาที่ใช้ short-circuit จงเขียนเมธอดนี้ใหม่

```
int f(int a, int b, int c, int d, int e, int f) {  
    int result = 0;  
    if (a > b) {  
        if (b != c) {  
            goto l1;  
        }  
    } else {  
        if (c > d) {  
            if (b != c) {  
                goto l1;  
            }  
        }  
    }  
    l2: return result;  
    l1: for (int i = 1; i <= f; i++) result = result + f;  
    goto l2;  
}
```

3. (4 คะแนน) มีนิยามฟังก์ชันที่รับค่า `int` ที่ต้องเขียนดังนี้

$$f(i) = i, \text{ if } 1 \leq i \leq 100$$

$$= 2i, \text{ if } 101 \leq i \leq 550$$

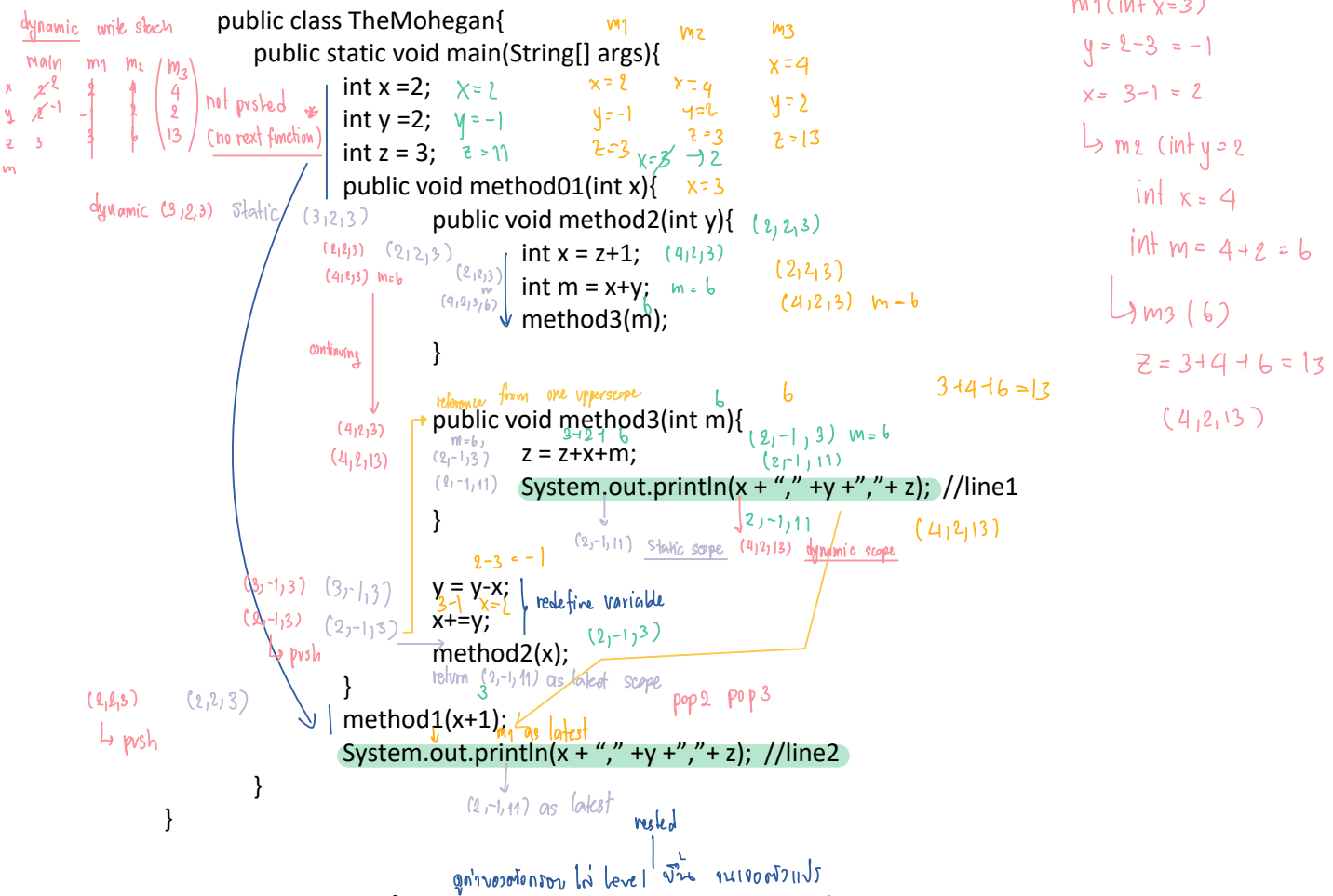
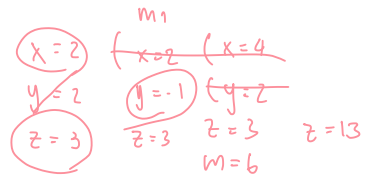
$$= 3i, \text{ if } 551 \leq i \leq 1000$$

$$= 0, \text{ otherwise}$$

สมมุติว่า `switch` สามารถเขียนโดยระบุ `range` ได้ เช่น 1 ถึง 10 ก็เขียนได้เลยว่า 1..10 ถ้าว่า นิสิตจะเลือกเขียนเมธอดนี้โดยใช้ `if else` หรือ `switch statement` จงบอกเหตุผลที่เลือก

ใช้ `if else statement` เพราะสะดวกต่อสนใจน้อยกว่า เทียบกับ `switch` ที่ต้องดูตัว 1000 กรณี
ซึ่งประสิทธิภาพการขวนนั้รวดเร็วจะไม่ดีกว่า

4. มีโค้ดของภาษาที่เมธอด nest กันได้ ดังนี้:



a. (3 คะแนน) ถ้าใช้ **static scope**, line1 กับ line 2 จะพิมพ์อะไรออกมา

2,-1,11
2,-1,11

b. (3 คะแนน) ถ้าใช้ **dynamic scope**, line1 กับ line 2 จะพิมพ์อะไรออกมา

4,2,13
2,-1,3