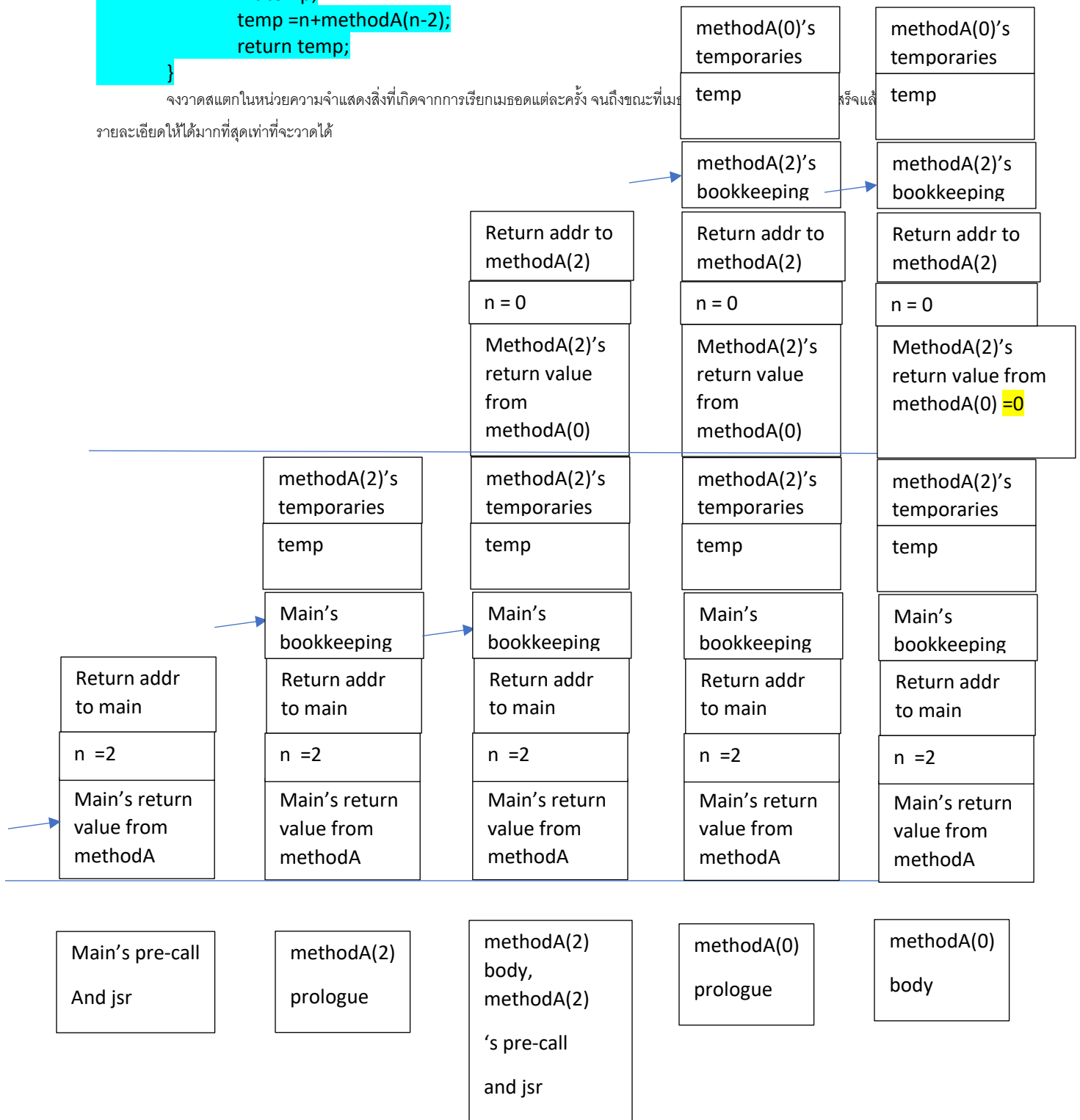
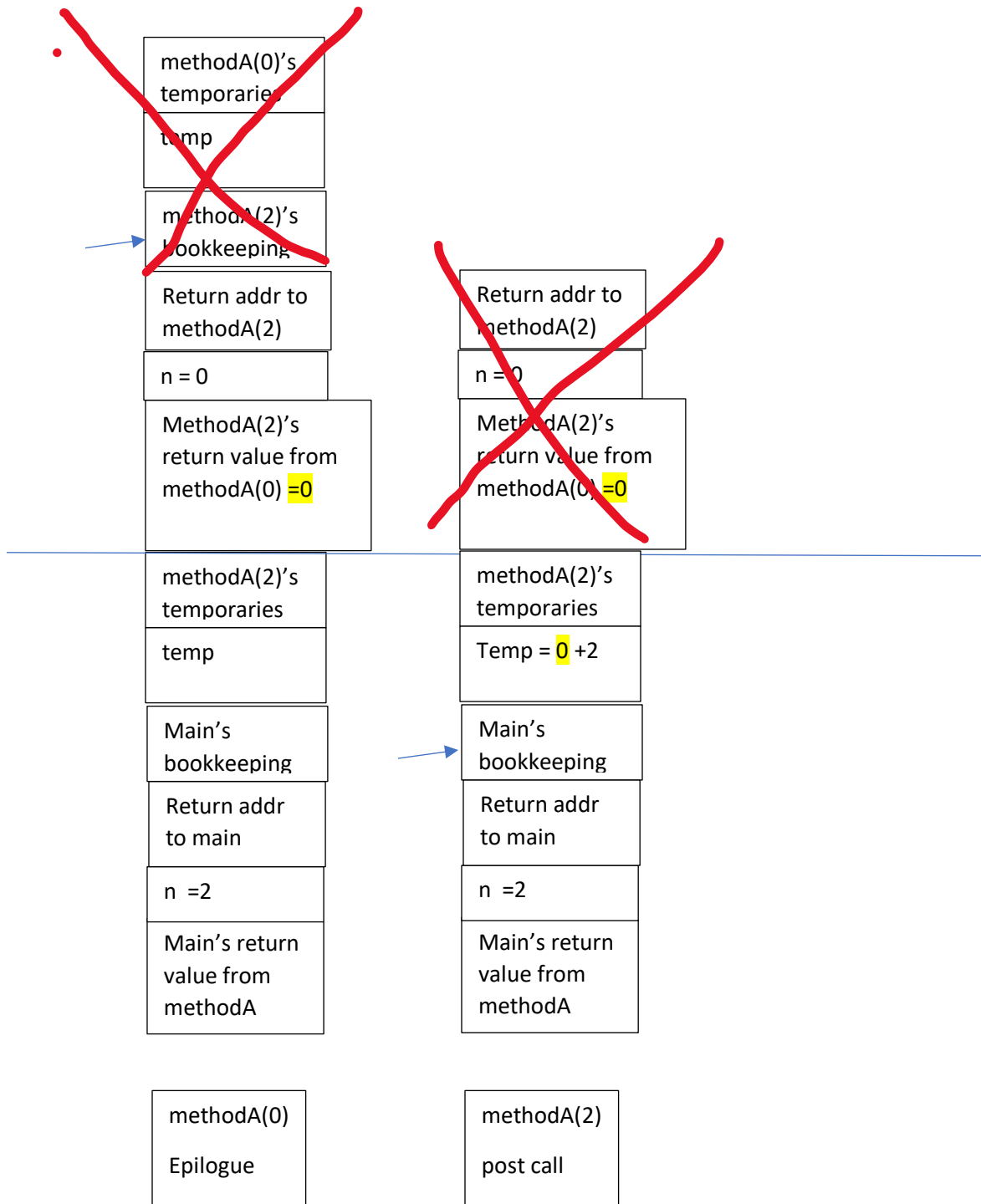


1. (10 คะแนน) มีโปรแกรมหนึ่งมีโค้ดดังนี้

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

จงวาดสแตคในหน่วยความจำแสดงสิ่งที่เกิดจากการเรียกเมธอดแต่ละครั้ง จนถึงขณะที่เมธอดจะเอยต์ให้ได้มากที่สุดเท่าที่จะวาดได้





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){  
            for(int i=1; i<=f; i++)  
                result = result + f;  
        }  
    } else{  
        if(c>d){  
            if(b!=c){  
                for(int i=1; i<=f; i++)  
                    result = result + f;  
            }  
        }  
    }  
    return result;  
}
```

3. (4 คะแนน) มีนิยามฟังก์ชันที่รับค่า `int` ที่ต้องเขียนดังนี้

$$\begin{aligned} f(i) &= i, \text{ if } 1 \leq i \leq 100 \\ &= 2i, \text{ if } 101 \leq i \leq 550 \\ &= 3i, \text{ if } 551 \leq i \leq 1000 \\ &= 0, \text{ otherwise} \end{aligned}$$

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

ถ้าเลือก `switch`

- (2 คะแนน) ต้องบอกได้ว่า `switch` สร้างโค้ดที่เร็วกว่า ไม่ต้องมากรอง `condition` เป็นขั้นๆ ตั้งแต่ต้นจนจบเหมือน `if-else`
- (2 คะแนน) มองว่า `jump table` ขนาดใหญ่ ไม่ทำให้เดือดร้อน `resource` สำหรับเครื่องคอมพิวเตอร์ที่มีหน่วยความจำมากพอ
ดังนั้นจึงเลือก `switch`

ถ้าเลือก `if else`

- (2 คะแนน) อธิบายว่า `switch` สร้างโค้ดที่เร็วกว่า ไม่ต้องมากรอง `condition` เป็นขั้นๆ ตั้งแต่ต้นจนจบเหมือน `if-else`
- (2 คะแนน) แต่ว่า กรณีข้างบน `jump table` ที่สร้างมาสำหรับ `switch` อาจใหญ่เกินไป กินที่สำหรับ `source code` ที่จะ `execute` ดังนั้นจึงตัดสินใจเลือก `if-else` แทน

4. มีโค้ดของภาษาที่เมธอด nest กันได้ ดังนี้:

```
public class TheMohegan{
    public static void main(String[] args){
        int x=2;
        int y=2;  $= 2-3 = -1$ 
        int z=3;  $= 3+4+6 = 13$ 
        public void method01(int x){
            public void method2(int y){
                int x = z+1;  $\rightarrow 3+1=4$ 
                int m = x+y;  $\rightarrow 4+2=6$ 
                method3(m);  $\rightarrow 6$ 
            }

            public void method3(int m){
                z = z+x+m;
                System.out.println(x + "," + y + "," + z); //line1
            }

            y = y-x;
            x+=y;
            method2(x);  $\rightarrow 2$ 
        }

        method1(x+1);  $\rightarrow 3$ 
        System.out.println(x + "," + y + "," + z); //line2
    }
}
```

\downarrow \downarrow \downarrow
2 -1 13

static

```
public static void main(String[] args){
    int x=2;
    int y=2;  $= 2-3 = -1$ 
    int z=3;  $= 3+4+6 = 13$ 
    public void method01(int x){
        public void method2(int y){
            int x = z+1;  $\rightarrow 3+1=4$ 
            int m = x+y;  $\rightarrow 4+2=6$ 
            method3(m);  $\rightarrow 6$ 
        }

        public void method3(int m){
            z = z+x+m;
            System.out.println(x + "," + y + "," + z); //line1
        }

        y = y-x;
        x+=y;
        method2(x);  $\rightarrow 2$ 
    }

    method1(x+1);  $\rightarrow 3$ 
    System.out.println(x + "," + y + "," + z); //line2
}
```

\downarrow \downarrow \downarrow
2 -1 13

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

2, -1, 13
2, -1, 13

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

4, 2, 13
2, -1, 13