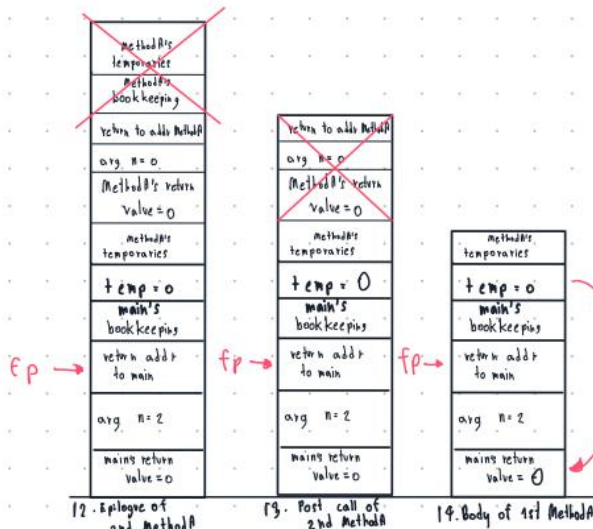
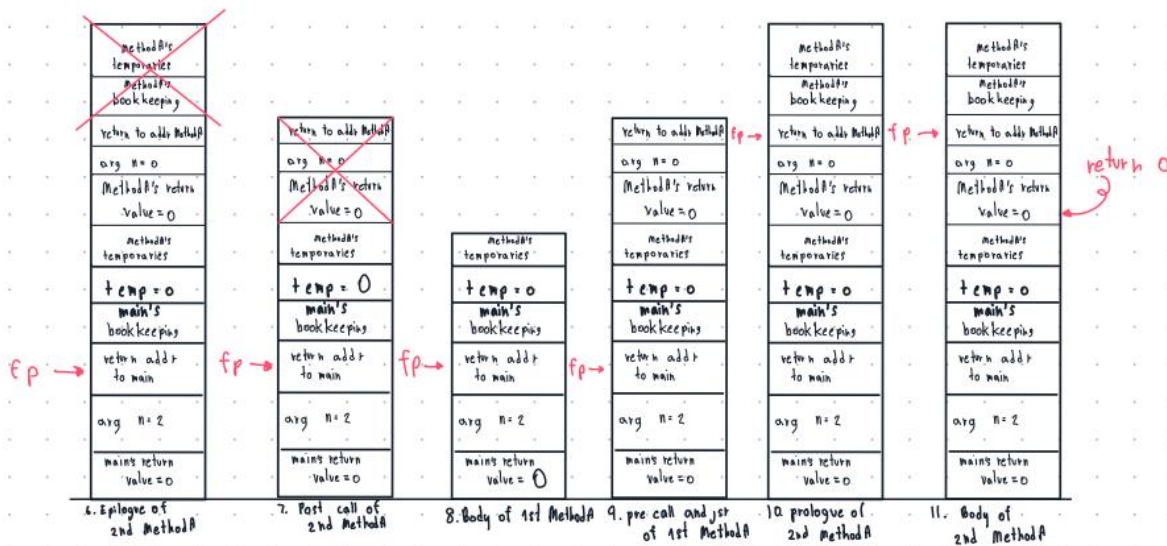
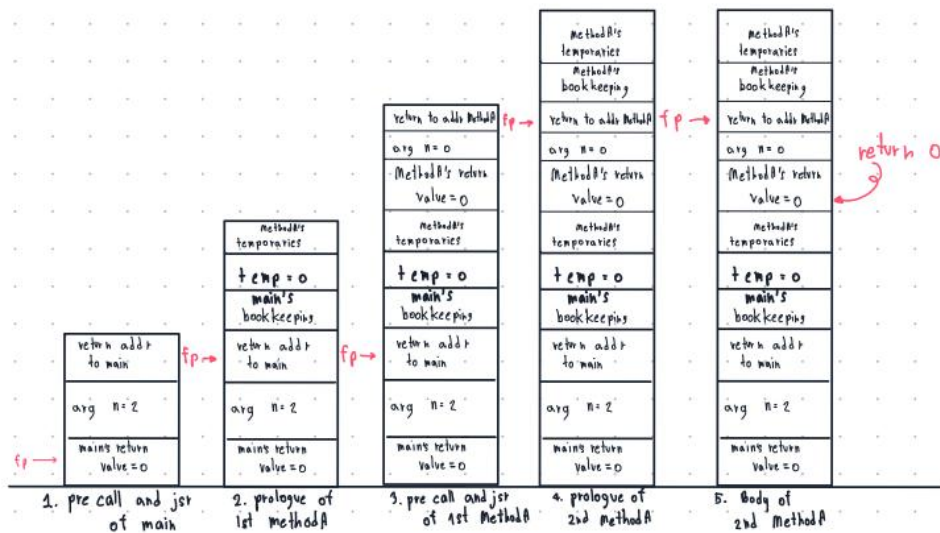


Homework 02

1. (10 คะแนน) มีโปรแกรมหนึ่งมีโค้ดดังนี้

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

จงวาดสแตคในหน่วยความจำแสดงสิ่งที่เกิดจากการเรียกเมธอดแต่ละครั้ง (ไม่ต้องแสดงเฟรมของ main) จนถึงขณะที่เมธอดสร้าง methodA(0) เฟรมที่สองเสร็จแล้วกำลังจะรีเทิร์น วาดรายละเอียดให้ได้มากที่สุดเท่าที่จะวาดได้



2. (3 คะแนน) มีโค้ด assembly อย่างง่ายดังนี้:

```
0: push 2 //push value 2 onto stack
1: popTo r1 //pop value in stack to register 1
2: push 0
3: popTo r2
4: push r1 //push value from r1 onto stack
5: if top == 1 pop , goto 28 //top is the top of stack, pop stack and "goto" jump to label 28
6: if top == 2 pop, goto 35
7: pop, goto 42

28: push r2
29: push r1
30: iadd //pop arguments from stack and add them, push result on stack
31: popTo r2
32: goto 45
35: push r2
36: push r1
37: imul //pop arguments from stack and multiply them, push result on stack
38: popTo r2
39: goto 45
42: iinc r2, 1 //increment value in register by 1
45: return
```

ถามว่า control flow ของโค้ดนี้ มีความรวดเร็วในฐานะ selection statement ดีเพียงพอหรือไม่ ปรับปรุงอะไรได้อีกหรือไม่ จงอธิบาย

คำตอบ: ตัว control flow ดังกล่าวนั้นมีความรวดเร็วดีเพียงพอ ซึ่งเทียบเคียงได้กับ if-else statement ใน ภาษา high level แต่ยังสามารถปรับปรุงตัว logic เพื่อลดจำนวน instruction ที่ใช้ได้ โดยที่ชัดเจนที่สุดคือ ในช่วงบรรทัด 28-32 กับ 35-39 นั้นคล้ายคลึงกันมาก โดยเปลี่ยนแค่ iadd เป็น imul เราจึงสามารถปรับเปลี่ยนโครงสร้างหรือใช้ instruction อื่นเล็กน้อยก็จะสามารถลดความซ้ำซ้อนตรงนี้ได้

3. มีโค้ดของภาษาที่เมธอด nest กันได้ ดังนี้:

```
public class XY{
    public static void main(String[] args){
        int x =1;
        int y =1;
        public void method01(int a){
            public void method2(int y){
                public void method3(int m){
                    m += x+y+a;
                    System.out.println(x + "," + y + "," +m); //line1
                }
                int x = y+1;
                int m = x+y;
                method3(m);
                System.out.println(x + "," + y + "," +m); //line2
            }
            x = x+ a - y;
            method2(x);
        }
        method1(x+1);
        System.out.println(x + "," +y); //line3
    }
}
```

ให้ถือว่าพารามิเตอร์ของเมธอด เป็นการ declare ตัวแปรเพื่อเตรียมใช้งานเลย

(8 คะแนน) ถ้าใช้ dynamic scope, line1 ถึง line 3 จะพิมพ์อะไรออกมา

คำตอบ:

line1: 3,2,12

line2: 3,2,5

line3: 2,1