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;
}
```

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

1.Main's pre-call and jsr

Return address to main
n = 2
Main's return value from methodA(2) <-fp

2.Prologue of methodA(2)

methodA(2)'s temporaries
temp
Main's bookkeeping
Return address to main <-fp
n = 2
Main's return value from methodA(2)

3.Pre-call and jsr of methodA(2)

Return address to methodA(2)
n = 0
methodA(2)'s return value from methodA(0)
methodA(2)'s temporaries
temp
Main's bookkeeping
Return address to main <-fp
n = 2
Main's return value from methodA(2)

4.Prologue of methodA(0)

methodA(0)'s temporaries
temp
methodA(2)'s bookkeeping
Return address to methodA(2) <-fp
n = 0
methodA(2)'s return value from methodA(0)
methodA(2)'s temporaries
temp
Main's bookkeeping
Return address to main
n = 2
Main's return value from methodA(2)

5.body of methodA(0)

methodA(0)'s temporaries
temp
methodA(2)'s bookkeeping
Return address to methodA(2) <-fp
n = 0
methodA(2)'s return value from methodA(0) = 0
methodA(2)'s temporaries
temp
Main's bookkeeping
Return address to main
n = 2
Main's return value from methodA(2)

6.epilogue of methodA(0)

Return address to methodA(2)
n = 0
methodA(2)'s return value from methodA(0) = 0
methodA(2)'s temporaries
temp
Main's bookkeeping
Return address to main <-fp
n = 2
Main's return value from methodA(2)

7. methodA(2) post call

methodA(2)'s temporaries
temp = 0 +
Main's bookkeeping
Return address to main <-fp
n = 2
Main's return value from methodA(2)

8.Pre-call and jsr of methodA(2)

Return address to methodA(2)
n = 0
methodA(2)'s return value from methodA(0)
methodA(2)'s temporaries
temp = 0 +
Main's bookkeeping
Return address to main <-fp
n = 2
Main's return value from methodA(2)

9.prologue of methodA(0)

methodA(0)'s temporaries
temp
methodA(2)'s bookkeeping
Return address to methodA(2) <-fp
n = 0
methodA(2)'s return value from methodA(0)
methodA(2)'s temporaries
temp = 0 +
Main's bookkeeping
Return address to main
n = 2
Main's return value from methodA(2)

10.body of methodA(0)

methodA(0)’s temporaries
temp
methodA(2)’s bookkeeping
Return address to methodA(2) <-fp
n = 0
methodA(2)’s return value from methodA(0) = 0
methodA(2)’s temporaries
temp = 0 +
Main’s bookkeeping
Return address to main
n = 2
Main’s return value from methodA(2)

11. epilogue of methodA(0)

Return address to methodA(2)
n = 0
methodA(2)’s return value from methodA(0) = 0
methodA(2)’s temporaries
temp = 0 +
Main’s bookkeeping
Return address to main <-fp
n = 2
Main’s return value from methodA(2)

12. post call of methodA(2)

methodA(2)’s temporaries
temp = 0 + 0
Main’s bookkeeping
Return address to main <-fp
n = 2
Main’s return value from methodA(2)

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

ดีเพียงพอแล้ว เพราะว่ามีการแบ่งกรณีที่ชัดเจนคือ top == 1 หรือ top == 2 หรือ กรณีอื่น ๆ แล้วในแต่ละกรณีมีการแบ่งชัดเจน เมื่อจบแต่ละกรณีก็จะไปรันที line 45 แล้วโค้ดในแต่ละส่วนมีการรันที่ไม่ซ้ำซ้อนกัน ทำให้รันได้อย่างมีคุณภาพเพียงพอแล้ว

```
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

3,2,12 //line 1 3,2,5 //line 2 2,1 //line3