

1. (3 คะแนน) มีโปรแกรม ที่เขียนแบบ recursive ในภาษาหนึ่งดังนี้

```
def factorial(x: Int, acc: Int): Int = {  
  if (x == 0) return acc  
  return factorial(x-1, x*acc)  
}  
  
def main(args: Array[String]): Unit = {  
  factorial(3, 1)  
}
```

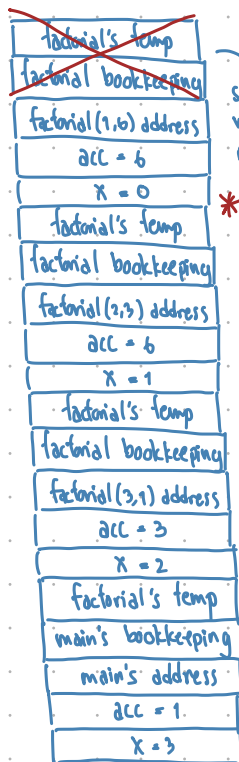
- (1.5 คะแนน) จงวาด stack frame จากการ รันตามขั้นตอน (pre-call, post-call, etc) ตามที่เรียน ณ เมื่อการรันกำลังจะ execute main body ของ factorial(2,3) แยกแต่ละเฟรมให้ชัดเจน วาดบนคอมพิวเตอร์เท่านั้น (ใช้ powerpoint หรือโปรแกรมอะไรก็ได้ วาดเสร็จจัดภาพมาแปะได้ แต่ต้องวาดบนคอมพิวเตอร์ให้บันทึกไว้ในวิดีโอได้)

หน้าต่อไป

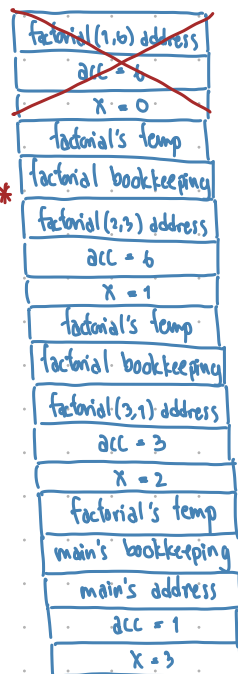
- (1.5) ลักษณะการใช้สแตกแบบนี้ สามารถปรับปรุงให้ใช้เนื้อที่สแตกเฟรมได้น้อยลง
  - ได้หรือไม่
  - ถ้าได้ ทำไม่ถึงได้
  - แล้วต้องทำอะไร (ให้นิสิตเสนอไอเดียมา)

ได้

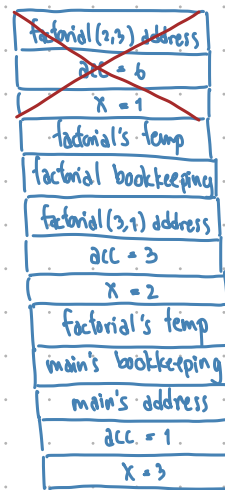
เปลี่ยน base condition เป็น if (n == 0) return acc เป็น if (n == 1) return acc



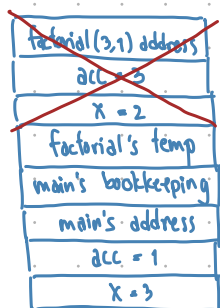
- \* return value can be returned via stack/register \*




<del>factorial's temp</del>
<del>factorial bookkeeping</del>
factorial(2,3) address
acc = 6
X = 1
factorial's temp
factorial bookkeeping
factorial(3,1) address
acc = 3
X = 2
factorial's temp
main's bookkeeping
main's address
acc = 1
X = 3



<del>factorial's temp</del>
<del>factorial bookkeeping</del>
<del>factorial(3,1) address</del>
acc = 3
X = 2
factorial's temp
main's bookkeeping
main's address
acc = 1
X = 3



<del>factorial's temp</del>
<del>main's bookkeeping</del>
main's address
acc = 1.
X = 3



main's address  
dec = 1  
X = 3

\* / = แก่กว่า

// = นอก scope \*

## 2. (2 คะแนน) สมมุติว่ามีภาษาแบบจาวา ที่มี nested เมธอดได้ ดังนี้

```
public static void main(String[] args){
    int x = 5;
    int y = 10;
    public void lloyd(int x){
        public void yor(int y){
            x = y + 1;
            anya(y-2);
        }

        public void anya(int m){
            int z = y + m;
            y = x + z;
            bond(y);
            public void bond(int m){
                y = m - x;
            }
            System.out.println(x + "," + y); //line1 5, -1
        }

        int m = y - x;
        yor(m);
    }
    lloyd(x+1); // 5, 10
    System.out.println(x + "," + y); //line2
}
```

- ให้ถือว่าพารามิเตอร์ของเมธอด เป็นการ **declare** ตัวแปร
- ให้การรันเป็นแบบใช้ไดนามิกสโคป ตัวแปรจะถือว่าไม่มีถ้าสโคปนั้นๆ รันเสร็จไปแล้ว
- ถามว่า line 1 กับ line 2 แต่ละ line ปริ้นอะไรบ้าง เติมคำตอบมา

Line1: 5, -1

Line2: 5, 10

### 3. (3 คะแนน) มี switch statement ดังนี้

```
switch (... /*tested expression */) {  
    case 51: clauseC;  
        break;  
    case 52:  
    case 55: clauseA;  
        break;  
    case 53: clauseB;  
        break;  
    default: clauseD;  
        break;  
}
```

จงเติมโค้ดด้านล่างที่เกิดจากการสร้าง assembly จาก switch statement นี้ให้สมบูรณ์

goto L5	0	T: &L3	--tested expression = 51
L1: <u>clauseA</u>	1	&L1	
goto L6	2	&L2	
L2: <u>clauseB</u>	3	&L4	
goto L6	4	& <u>L1</u>	--tested expression = 55
L3: clauseC		L5: r1 = tested expression	
goto L6		if r1 < <u>51</u>	goto L4
L4: clauseD		if r1 > <u>55</u>	goto L4
goto L6		r1 -= <u>51</u>	
		r1 = T[r1]	
		goto *r1	
		L6:	

4. (2 คะแนน) จง ☒ เพื่อระบุว่าเกิดอะไรขึ้นกับโค้ด Java ด้านล่าง (สามารถเลือกได้มากกว่า 1 อย่างในแต่ละบรรทัด แต่ถ้าเลือกเกินจะถูกหักคะแนน) การคิดบรรทัดหนึ่ง ให้สมมุติว่าบรรทัดก่อนหน้านี้ที่มี error นั้นโดนคอมไพล์ไว้

**\*\*ข้อนี้ให้อ่านโจทย์แล้วพิจารณาเอง ห้ามรันโปรแกรม ใครรันโปรแกรมจะให้ 0 \*\***

```
class Robot {...}
class RealRobot extends Robot {...}
class FantasyRobot extends Robot {...}
...
Robot a1, a2;
RealRobot r1, r2;
FantasyRobot f1, f2;
Robot[] rs = new Robot[3];
```

```
a1 = new RealRobot();
a2 = new FantasyRobot();
f1 = (FantasyRobot)a1; ☐ compiles OK ☐ compile time error ☐ runs OK ☐ runtime error
a1 = a2; ☐ compiles OK ☐ compile time error ☐ runs OK ☐ runtime error
f2 = (FantasyRobot)a1; ☐ compiles OK ☐ compile time error ☐ runs OK ☐ runtime error
r1 = new RealRobot();
rs[0] = r1;
rs[1] = a1;
r2 = (RealRobot)rs[1]; ☐ compiles OK ☐ compile time error ☐ runs OK ☐ runtime error
```