1. Scope checking

กำหนดโค้ดต่อไปนี้

```
1.int b = 5
2.
3.Function foo(){
4.        int a = b+5
5.        return a;
6.}
7.
8..Function bar(){
9.        int b = 2;
10.       return foo();
11.}
12.Main(){
13.       print foo();
14.       print '\t';
15.       print bar();
16.       return 0;
17.}
```

กำหนดให้ compiler ทำงานแบบ dynamic scoping

1.1. จงเขียน Symbol table แบบ Explicit เมื่อเรียกคำสั่งในบรรทัดที่ 5 เป็นครั้งแรก และ แสดงผลลัพธ์หลังเรียกฟังก์ชั่นบรรทัดที่ 13 (2 คะแนน)

1.2. จงเขียน Symbol table แบบ Explicit เมื่อเรียกคำสั่งในบรรทัดที่ 5 เป็นครั้งที่สอง และ แสดงผลลัพธ์หลังเรียกฟังก์ชั่นบรรทัดที่ 15 (2 คะแนน)

1.3. จงเขียน Spaghetti stack ของโปรแกรมนี้ (2 คะแนน)

2. Type Checking

Rule 1: X is an identifier X in scope S and has Type T ——————— S ⊢ X : T	Rule 2: f is an identifier f is a function in scope S f has type(T1 , … , Tn) -> U f ⊢ Ei : Ti for 1 ≤ I ≤ n ——————— S ⊢ f(E1,…,En) : U	Rule 3: S ⊢ E1 : T[] S ⊢ E2 : int ——————— S ⊢ E1[E2] : T
Rule 4: S ⊢ E1 : T S ⊢ E2 : T ——————— S ⊢ E1=E2 : T	Rule 5: X is an integer constant ——————— S ⊢ X : int	Initial fact: X is ID, f is ID f is a function in scope S f has type (float) -> float X has type float in scope S a has type float[] in scope S 10 is integer constant

Code:
```
float f(float);
float x, a[10];
x = f(a[1]);
```

จงทำการตรวจสอบแบบ type checking ของโค้ดต่อไปนี้โดยระบุถึง Rule number ที่ใช้, เงื่อนไข, และเขียน fact ที่เกิดขึ้นใหม่ด้วย (6 คะแนน)

No.	Rule Number	Condition	Fact
1			
2			
3			
4			
5			

3. TAC

Code

```
total = capital + interest;
balance = total * total;
gain = 0;
temp = capital + interest;
asset = temp * temp;
growth = gain + 3;
net = asset + growth;
```

กำหนดตัวแปร asset และ net ยังอยู่จนถึงการทำงานที่บรรทัดสุดท้าย ให้นักศึกษาทำการ Optimization ด้วยวิธีการต่างๆตามที่กำหนด โดยให้เขียนโค้ดที่ผ่านการ optimization ตามขั้นตอนทั้งหมด ระบุตำแหน่งอย่างชัดเจน (ว่าแก้อะไรไปบ้าง ที่ไหน) และให้นำอินพุทจากข้อปัจจุบันไปใช้ในข้อถัดไปด้วย (ตัวอย่างเช่น โค้ดข้อ 2 จะต้องถูกนำไปใช้ในข้อ 3)

i. Common sub-expression elimination (1 คะแนน)

ii. Copy propagation (1 คะแนน)

iii. Constant folding (1 คะแนน)

iv. Dead code(1 คะแนน)

v. ให้ทำกระบวนการตามข้อ i. ii. iii. iv. อีกครั้งโดยเขียนเฉพาะ Source code สุดท้าย (2 คะแนน)

4. Register allocation

code	a	b	c	d	e	f	g	h
a = b+c;								
_L0:								
d = 100 < a;								
ifZ d Goto _L1								
a = b + e + c;								
g = a < b;								
ifZ g Goto _L2								
b = b – h;								
Goto _L3;								
_L2:								
b = b+f;								
_L3:								
Goto _L0;								
_L1:								
a = b+c+e+f+h;								

4.1. จงเขียนตาราง live interval ของแต่ละตัวแปร (โดยใช้ X เพื่อแทนการระบายสีได้)(4 คะแนน)

4.2. จงใช้ chaitin's algorithm โดยกำหนดให้มี Register ทั้งหมด 7 ตัว จงหาว่ามีตัวแปรใดบ้างที่ถูก spilled ถ้ามี โปรดระบุ ? (2 คะแนน)

5. Garbage Collection

```
code
class Node{
    Node next;
}
main (){
    Node previous, root, current;
    root = new Node;
    current = root;
    for(i=0;i<5;i++){
            current.next = new Node;
            previous = current;
            current = current.next;
    }
    prev.next = root;
    return 0;
}
```

5.1. จงวาดแผนผังการทำงานของ Garbage collection ด้วยวิธี Mark-and-Sweep ที่ขั้นตอนสุดท้ายของโปรแกรม (3 คะแนน)

5.2. จงระบุตำแหน่งในแผนผังว่า object ใด ถูกหน่วยความจำเรียกคืน (1 คะแนน)

5.3. จำนวน pointer ของ work-list ที่ใช้ในโปรแกรม (2 คะแนน)