**งานชิ้นที่ 3 วิชา 01076262 Compiler Construction**

**“Expression Evaluator”**

# **คำสั่ง**

ให้นักศึกษาจับกลุ่ม 3 คน เพื่อทำโปรแกรม Expression Evaluator ด้วย GNU Flex และ GNU Bison ในการคำนวณนิพจน์ สำหรับคิดคำนวณจำนวนเต็มบวก ขนาด 32 บิต คิดเครื่องหมาย แบบ Infix ทีละบรรทัด โดยรองรับความสามารถขั้นต่ำดังนี้

1. ตัวกระทำ (operator)
   1. กระทำบิต AND, OR, NOT
   2. ติดลบ –
   3. บวก +, ลบ -, คูณ \*, หารเอาส่วน /, หารเอาเศษ \
   4. วงเล็บ ()
2. ตัวถูกกระทำ (operand)
   1. ค่าคงที่
      1. เลขฐาน 10
      2. เลขฐาน 16 (*????***h**)
   2. รีจีสเตอร์และพื้นที่จัดเก็บแบบไดนามิก
      1. สำหรับเก็บค่า **$rA** - **$rZ**
      2. สำหรับเก็บผลลัพธ์ที่กระทำล่าสุดเสมอ **$acc**
      3. สั่งเก็บลง/ดึงจากสแต็ก **PUSH** *reg*, **POP** *reg* (ควรจัดทำแบบลิงก์ลิสต์)
      4. สำหรับอ้างถึงค่าที่จัดเก็บบนสุดของสแต็ก (READONLY!) **$top**
      5. สำหรับอ้างถึงจำนวนข้อมูลที่จัดเก็บในสแต็ก (READONLY!, ≥ 0) **$size**
      6. ทั้งหมดนี้สามารถสั่งแสดงค่าที่เก็บด้วยคำสั่ง **SHOW** *reg*
      7. ทั้งหมดนี้สามารถสั่งถ่ายสำเนาค่าด้วยคำสั่ง **LOAD** *reg1* *reg2*

# **การส่ง**

รวบรวมแฟ้มรายงานชื่อ nnnnNNNN.PDF ส่วนโค้ดและอื่นๆ ใส่แฟ้มบีบอัดชื่อ nnnnNNNN.ZIP (โดย nnnn คือรหัส นศ. สี่ตัวท้ายที่น้อยสุด NNNN คือรหัส นศ. สี่ตัวท้ายที่มากสุด) ของทุกคนบันทึกลงในแผ่นซีดี/ดีวีดีเพียงแผ่นเดียวส่งในตู้รับเอกสารของอาจารย์ผู้สอนภายในวันศุกร์ที่ 2 มีนาคม 2561

# **คะแนน**

ชิ้นงานนี้คิดเป็น 10% ของคะแนนทั้งหมด (อาจนัดนำเสนอชิ้นงานนี้แบบสุ่มภายหลัง เพื่อสอบทานความเข้าใจ)

# **หมายเหตุ**

* Hint: "Bison Infix Notation Calculator"
* หากลอกหรือคล้ายคลึงกัน หรือผิดเงื่อนไขข้างต้น จะไม่ได้คะแนนเลย
* รูปสกรีนชอต (screenshot) ไม่ควรเป็นพื้นสีดำหรือสีเข้ม
* ควรมี error handling แต่ไม่จำเป็นต้องซับซ้อน
* รายงานต้องประกอบไปด้วย
* แนวคิดและรายละเอียดวิธีการดำเนินงานสร้างเชิงเทคนิค
* คำอธิบายโค้ดในไฟล์ flex และ bison
* ผลการรันกับตัวอย่างหลากหลายรูปแบบ ทั้งรูปแบบที่ถูกต้อง (20++) และไม่ถูกต้อง (10++)