ข้อสอบวัดประสิทธิภาพของการโปรแกรมของนักศึกษาระดับปริญญาตรี (Exit Exam)

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ ครั้งที่ 1 ประจำปีการศึกษา 2566

วันอาทิตย์ที่ 10 ตุลาคม พ.ศ. 2566

เวลาสอบ 9.30 – 12.30 น

ข้อที่ 1

อาจารย์โมคิดว่าระบบจัดการฐานข้อมูลสำหรับระบบ PEARLS มันใช้งานยากและมีประสิทธิภาพที่ต่ำเกินไป อาจารย์โมเลยต้องการสร้าง Query Engine (ซึ่งเป็นส่วนที่ใช้ในการประมวลผลของ Query) ขึ้นมา ชื่อว่า “Vemox” โดยมีความต้องการดังนี้

ความต้องการของโจทย์ (Requirements)

1) เขียนโปรแกรมโดยใช้แนวคิด MVC (Model-View-Controller) Design Pattern หรือแนวคิดอื่น ๆ ที่คล้ายคลึง โดยใช้ภาษาโปรแกรมใดก็ได้
2) Query Engine “Vemox” เป็น Query Engine อย่างง่ายที่ทำงานได้แค่การประมวลผล SQL Query ได้เท่านั้น เบื้องต้น อาจารย์โมต้องการประมวลผลคำสั่ง SQL อย่างง่าย ได้แก่
   - SELECT **[จำนวนเต็ม]**; ซึ่งจะคืนค่า **[จำนวนเต็ม]** ออกมา เช่น SELECT 312; จะคืนค่า 312 ออกมา
   - SELECT **[จำนวนเต็ม] + [จำนวนเต็ม]**; ซึ่งจะคืนค่าเป็นผลบวกของนิพจน์ดังกล่าวออกมา เช่น SELECT 1 + 1; จะคืนค่า 2 ออกมา

   โดยที่นักศึกษาจะต้องสร้าง Query Engine ดังกล่าวขึ้นมารองรับการประมวลผลคำสั่ง SQL ทั้งหมด 3 รูปแบบข้างต้น ซึ่ง Query Engine สามารถประมวลผลคำสั่ง SQL ได้ครั้งละ 1 คำสั่งเท่านั้น **รวมทั้ง Query Engine จะต้องสามารถจับเวลาการประมวลผลคำสั่งดังกล่าวเป็นมิลลิวินาทีได้**
3) แต่นักศึกษาที่พัฒนาระบบซึ่งเชื่อมต่อกับระบบฐานจัดการข้อมูลไม่ชอบเขียน SQL ดังนั้นอาจารย์โมจึงพัฒนาอีกภาษานึงชื่อว่า MQL โดยมีคำสั่งดังนี้
   - **print([จำนวนเต็ม])**; ซึ่งจะคืนค่า **[จำนวนเต็ม]** ออกมา เช่น print(312); จะคืนค่า 312 ออกมา
   - **print([จำนวนเต็ม] + [จำนวนเต็ม]);** ซึ่งจะคืนค่าเป็นผลบวกของนิพจน์ดังกล่าวออกมา เช่น print(1 + 1); จะคืนค่า 2 ออกมา

โดย Query Engine ที่นักศึกษาสร้างจะต้องรองรับ Query ภาษา MQL ได้เช่นกัน โดยที่นักศึกษาอาจจะให้ผู้ใช้ระบุได้ว่า Query ที่ผู้ใช้ป้อนเป็นภาษา SQL หรือ MQL (เช่น แยกปุ่มออกเป็น 2 ปุ่มในการส่ง)

4) การแสดงผลของ Query Engine “Vemox” ที่อาจารย์โมต้องการในแต่ละ Query มีลักษณะดังนี้
   - หาก Query นั้นเขียนไม่ถูกต้องตามรูปแบบที่กำหนด (รวมถึงช่องว่างที่ไม่ตรง) ให้แสดงผลเป็นข้อความ Wrong Syntax!
   - หาก Query นั้นเขียนถูกต้องแล้ว (ซึ่งสามารถประมวลผลได้) ให้แสดงผลลัพธ์เป็นข้อความ 3 บรรทัด คือ
     - บรรทัดที่ 1 เป็น Query: [คำสั่ง Query ที่ถูกนำไปประมวลผล]
     - บรรทัดที่ 2 เป็น Result: [ผลลัพธ์ของ Query ที่ถูกนำไปประมวลผล]
     - บรรทัดที่ 3 เป็น Query Execution Time: [เวลาที่ใช้ในการประมวลผล Query เป็นมิลลิวินาที] ms
5) แต่นักศึกษาที่พัฒนาระบบ PEARLS ต้องการการแสดงผลของ Query ที่สามารถนำไปใช้ได้ง่ายกว่านี้ โดยมีลักษณะดังนี้
   - หาก Query นั้นเขียนไม่ถูกต้องตามรูปแบบที่กำหนด (รวมถึงช่องว่างที่ไม่ตรง) ให้แสดงผลเป็นข้อความ { "status": "false" }
   - หาก Query นั้นเขียนถูกต้องแล้ว (ซึ่งสามารถประมวลผลได้) ให้แสดงผลลัพธ์เป็นข้อความ { "status": "true", "query": "[คำสั่ง Query ที่ถูกนำไปประมวลผล]", "result": "[ผลลัพธ์ของ Query ที่ถูกนำไปประมวลผล]", "execution_time": [เวลาที่ใช้ในการประมวลผล Query เป็นมิลลิวินาที] }

โดยที่นักศึกษาต้องสามารถรองรับการแสดงผลได้ทั้งสองรูปแบบ โดยให้สามารถเลือกการแสดงผลได้ในหน้าจอ

ตัวอย่างการทำงาน

ตัวอย่างการใช้งานของผู้ใช้	คำตอบตามข้อ 4)	คำตอบตามข้อ 5)
ใช้ SQL SELECT 1	Wrong Syntax!	{ "status": "false" }
ใช้ SQL SELECT 1;	Query: SELECT 1; Result: 1 Query Execution Time: 2 ms	{ "status": "true", "query": "SELECT 1;", "result": "1", "execution_time": 2 }
ใช้ MQL print(2 + 1);	Query: print(2 + 1); Result: 3 Query Execution Time: 5 ms	{ "status": "true", "query": "print(2 + 1);", "result": "3", "execution_time": 5 }

การส่งคำตอบสำหรับข้อนี้

1) ส่ง Source Code ของโปรแกรมดังกล่าวทั้งหมด
2) ส่ง Script สำหรับการสร้างฐานข้อมูล (หากมี)
3) ข้อความอธิบายการทำข้อสอบแต่ละส่วน ในคอมเมนต์ของ Google Classroom ซึ่งประกอบด้วย ไฟล์ที่ส่งมาไฟล์ใดทำหน้าที่อะไรใน MVC และทำงานร่วมกันอย่างไร และทำข้อสอบข้อที่เท่าไหร่

ข้อที่ 2

สืบเนื่องจากโจทย์ของเมื่อวานที่อาจารย์โมต้องการคิดภาษาโปรแกรมใหม่ชื่อว่า **CSGO** (Computer Science's Go) ซึ่งเป็นภาษารูปแบบที่ใช้ Compiler ในการแปลเป็นภาษาเครื่อง

Compiler นอกจากการแปลเป็นภาษาเครื่องแล้ว Compiler ยังสามารถทำให้ Source Code มีประสิทธิภาพที่มากขึ้นได้ โดยเราเรียกส่วนของ Compiler ที่ใช้ในการปรับปรุง Source Code ว่า Optimizer

กระบวนการทำงานของ Optimizer จะเริ่มจากการวิเคราะห์ Source Code และปรับปรุงตามกฎต่าง ๆ ซึ่งกฎในการปรับปรุง Code อย่างง่าย เรียกว่า Local Optimization ซึ่งทำได้อย่างน้อย 3 วิธี คือ

1. Constant Folding
2. Algebraic Identity
3. Strength Reduction

อาจารย์โมต้องการให้นักศึกษาสร้าง Optimizer ที่ทำ Local Optimization ทั้งสามวิธีนี้ โดยมีรายละเอียดตามความต้องการด้างล่าง **ซึ่งโปรแกรมต้องอนุญาตให้ผู้ใช้งาน Optimizer เลือกได้ว่าจะทำ Local Optimization วิธีไหนบ้าง (สามารถทำได้มากกว่า 1 วิธีในเวลาเดียวกัน)**

ความต้องการของโจทย์ (Requirements)

1) เขียนโปรแกรมโดยใช้แนวคิด MVC (Model-View-Controller) Design Pattern หรือแนวคิดอื่น ๆ ที่คล้ายคลึง โดยใช้ภาษาโปรแกรมใดก็ได้เขียนโปรแกรมโดยใช้แนวคิด MVC (Model-View-Controller) Design Pattern หรือแนวคิดอื่น ๆ ที่คล้ายคลึง โดยใช้ภาษาโปรแกรมใดก็ได้
2) โปรแกรมนี้จะต้องรับไฟล์ Source Code ของโปรแกรมที่เขียนด้วยภาษา CSGO โดยมีลักษณะดังนี้
   - ภาษา CSGO มีแค่ 3 คำสั่งเท่านั้น คือ
     - บวกจำนวนเต็มสองตัวเข้าด้วยกันและเก็บลงไปในตัวแปร โดยการพิมพ์ว่า **[ชื่อตัวแปร] = [จำนวนเต็ม] + [จำนวนเต็ม]** อย่างเช่น total = 3 + 5
     - คูณตัวแปรเข้ากับจำนวนเต็มและเก็บลงไปในตัวแปรอีกตัวนึง โดยการพิมพ์ว่า **[ชื่อตัวแปร] = [ชื่อตัวแปร] * [จำนวนเต็ม]** อย่างเช่น total = age * 2
   - แต่ละคำสั่งในภาษา CSGO จะต้องเขียนแยกบรรทัดกันเท่านั้น ไม่จำเป็นต้องมีเครื่องหมายเพื่อปิดบรรทัด

- นักศึกษาจะต้องแสดง Source Code หลังจากการปรับปรุงของ Optimizer ให้กับผู้ใช้งาน **ซึ่งโปรแกรมจะต้องสามารถให้ผู้ใช้เลือกได้ว่า จะปรับปรุง Code โดยใช้กฎตามข้อใดบ้าง**

3) โปรแกรมนี้จะต้องสามารถทำ Constant Folding ซึ่งเป็นหนึ่งในการปรับปรุง Code ในรูปแบบ Local Optimization ได้ เมื่อเจอบรรทัดที่มีการบวกจำนวนเต็มสองจำนวนเข้าด้วยกัน ซึ่งมีหลักการดังนี้
   - แทนที่จะให้คอมพิวเตอร์คำนวณผลบวกหลังจากที่รันโปรแกรมไปแล้ว Compiler สามารถคำนวณผลบวกให้ได้ เช่น การปรับปรุงจาก age = 3 + 5 เป็น age = 8
4) โปรแกรมนี้จะต้องสามารถทำ Algebraic Identity ซึ่งเป็นหนึ่งในการปรับปรุง Code ในรูปแบบ Local Optimization ได้ เมื่อเจอบรรทัดที่มีการคูณตัวแปรเข้ากับจำนวนเต็ม ซึ่งมีหลักการดังนี้
   - หากพบคำสั่งในลักษณะ y = x * 0 เราสามารถใช้สมบัติทางคณิตศาสตร์ที่เมื่อจำนวนใด ๆ คูณกับ 0 แล้วจะได้ 0 เพื่อลดจำนวนการคูณได้ เช่น การปรับปรุงจาก total = age * 0 เป็น total = 0
   - หากพบคำสั่งในลักษณะ y = x * 1 เราสามารถใช้สมบัติทางคณิตศาสตร์ที่เมื่อจำนวนใด ๆ คูณกับ 1 แล้วผลคูณจะเท่ากับจำนวนใด ๆ เพื่อลดจำนวนการคูณได้ เช่น การปรับปรุงจาก total = age * 1 เป็น total = age
5) โปรแกรมนี้จะต้องสามารถทำ Strength Reduction ซึ่งเป็นหนึ่งในการปรับปรุง Code ในรูปแบบ Local Optimization ได้ เมื่อเจอบรรทัดที่มีการคูณตัวแปรเข้ากับจำนวนเต็มบางจำนวน ซึ่งมีหลักการดังนี้
   - หากพบคำสั่งในลักษณะ y = x * 2 เราสามารถเปลี่ยนจากรูปการคูณให้เป็นการบวกได้ เช่น การปรับปรุงจาก total = age * 2 เป็น total = age + age
   - หากพบคำสั่งในลักษณะ y = x * 4 เราสามารถเลือกใช้คำสั่งที่คอมพิวเตอร์สามารถคำนวณได้เร็วกว่าได้อย่างเช่นการ Shift Right แทนได้ เช่น การปรับปรุงจาก total = age * 4 เป็น total = age >> 2

ตัวอย่างการทำงาน

ตัวอย่าง Source Code ภาษา CSGO	คำตอบของโปรแกรม (หากปรับปรุงโดยใช้ทุกวิธี)
aa = 3 + 5 bb = aa * 0 cc = bb * 1 dd = cc * 2 ee = dd * 4	aa = 8 bb = 0 cc = bb dd = cc + cc ee = dd >> 2

การส่งคำตอบสำหรับข้อนี้

1) ส่ง Source Code ของโปรแกรมดังกล่าวทั้งหมด
2) ส่ง Script สำหรับการสร้างฐานข้อมูล (หากมี)
3) ข้อความอธิบายการทำข้อสอบแต่ละส่วน ในคอมเมนต์ของ Google Classroom ซึ่งประกอบด้วย ไฟล์ที่ส่งมาไฟล์ใดทำหน้าที่อะไรใน MVC และทำงานร่วมกันอย่างไร และทำข้อสอบข้อที่เท่าไหร่