

Programming for NLP 2025

PA 1: Karel

อ่านนี่ก่อนนะ

Programming Assignment มีเทคนิคและกฎ กติกา มารยาทดังนี้

- โจทย์ทุกข้อจะดูเหมือนว่ามันยาก แต่พอนั่ง decompose ดีๆ และเขียน pseudocode บนกระดาษ ก่อนมันจะง่ายลงเยอะมาก
- ถ้าเป็นไปได้อย่า print document นี้ออกมา เพราะทีมงานอาจจะเข้ามาเติมรายละเอียดต่าง ๆ ถ้าเกิดมีอะไรไม่ค่อยชัดเจน
- ส่ง 3 ไฟล์เท่านั้นบน course ville: fix_road_karel.py mak_rook_karel.py vote_karel.py
- ห้ามเปลี่ยนชื่อไฟล์เอง ให้ส่งด้วยชื่อไฟล์เดิมที่ให้ไป และห้าม zip file เพื่อให้โค้ดรันได้โดยสคริปตรวจ
- สำหรับ PA นี้ เพราะฉะนั้นคุยกับเพื่อนได้ แต่ต้องเขียน solution ของเราเอง และแต่ละคนมักจะไม่เหมือนกันเลย
- ชิ้นงานนี้มีการเก็บคะแนน 10 คะแนน
- ระยะเวลาการทำ 14 วัน (15 สิงหาคม - 28 สิงหาคม 23:59 น.) กรณีส่งสาย ชิ้นงานจะถูกตัดคะแนน 20% ทันที
- หากมีข้อสงสัยหรือปัญหาระหว่างการทำ assignment สามารถสอบถามทาง Discord ได้

Problem 1: fix_road_karel.py

Karel ต้องการจะซ่อมถนนที่เป็นหลุมเป็นบ่อ โดยการเอา Beeper ไปอุดรู โดยที่ assume ว่า

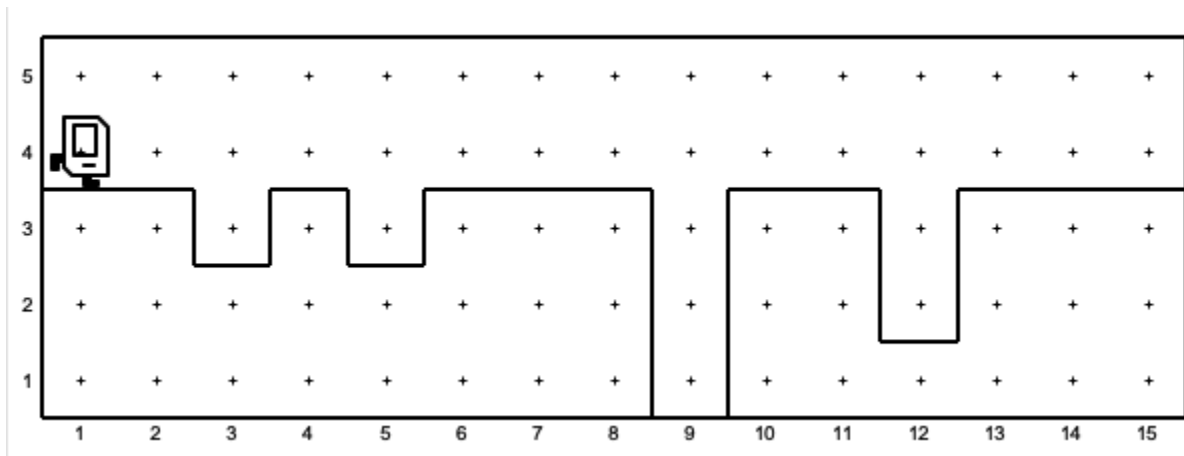
- หลุมความลึกไม่เท่ากัน
- หลุมมีความกว้าง 1 ช่อง เท่ากันทุกหลุม
- หลุมทุกหลุมมีกั้นหลุม
- ส่วนของถนนที่ไม่มีหลุมมีระดับเท่ากันหมด
- ไม่รู้หลุมอยู่ตรงไหนบ้าง

- ไม่รู้ว่าถนนมีความยาวเท่าไร
- Karel จะต้องจบที่จุดถนน

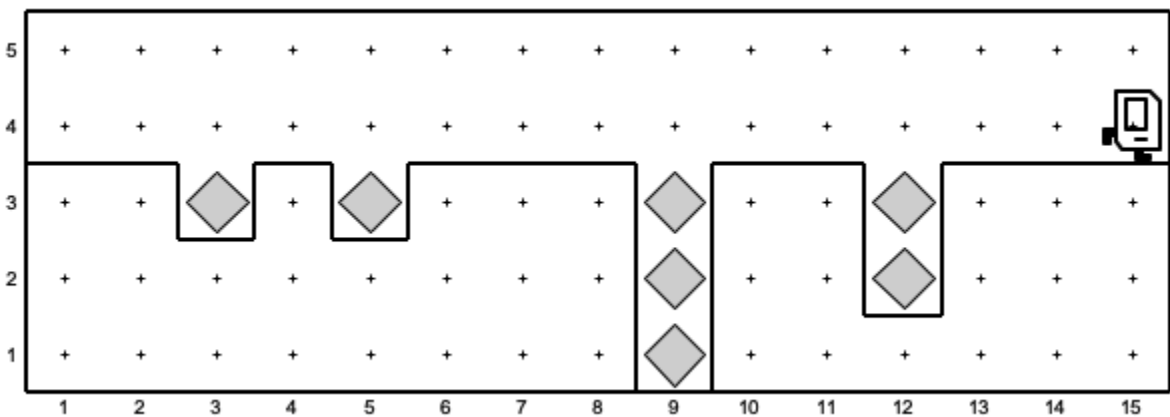
โปรแกรมจะต้องทำงานได้ถูกต้องทั้ง fix_road_karel1.w และ fix_road_karel2.w

ตัวอย่าง

ก่อน



หลัง



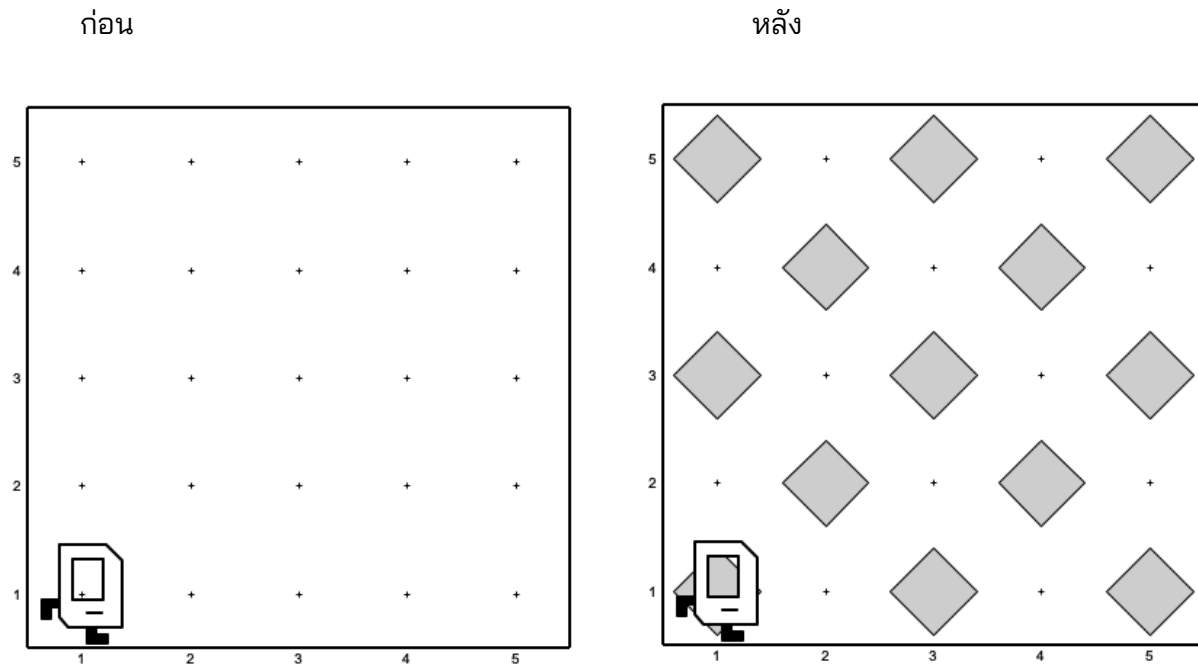
Problem 2: mak_rook_karel.py

Karel ต้องการวางตารางหมากรุก (หรือหมากรุก) ก็ได้ คือการวาง Beeper ช่องเว้นช่อง และแถวเว้นแถว โดยที่ assume ว่า

- world มีขนาดอย่างน้อย 2x2
- ผลสำเร็จจะต้องมี beeper ที่มุมซ้ายล่าง

โปรแกรมจะต้องทำงานได้ถูกต้องในทุก world ที่โล่งๆ ไม่มี beeper และไม่มีกำแพงขวางภายใน เช่น 3x5.w 8x1.w 8x8.w เป็นต้น

ตัวอย่าง การรันโปรแกรมบน 5x5.w



Problem 3: vote_karel1.py

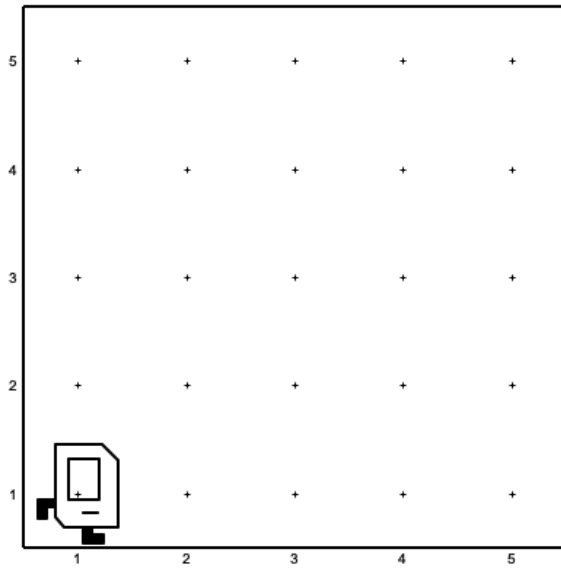
Karel ต้องการขีดกากบาทลงใน world โดยที่ assume ว่า

- world ของเราใหญ่อย่างน้อย 3x3
- world เป็นสี่เหลี่ยมจัตุรัส
- Karel จบตรงไหนก็ได้

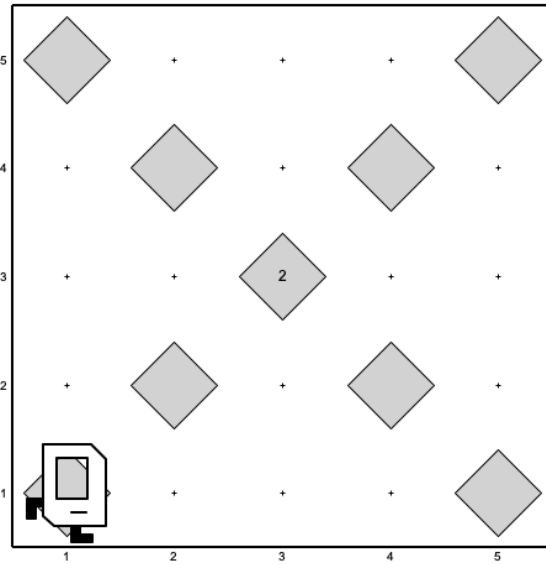
โปรแกรมจะต้องทำงานได้ถูกต้องในทุก world ที่โล่งๆ ไม่มี beeper และไม่มีกำแพงขวางภายใน เช่น 5x5.w 8x8.w เป็นต้น

ตัวอย่าง การรันโปรแกรมบน 5x5.w

ก่อน



หลัง



Karel Quick Reference

<https://compedu.stanford.edu/karel-reader/docs/python/en/reference.html>

Base Karel commnds: <code>move()</code> <code>turn_left()</code> <code>put_beeper()</code> <code>pick_beeper()</code>	Conditions: <code>if condition:</code> <code>code run if condition passes</code> <code>if condition:</code> <code>code block for "yes"</code> <code>else:</code> <code>code block for "no"</code>
Karel program structures: <code># Comments can be included in any part</code> <code># of a program. They start with a #</code> <code># and include the rest of the line.</code> <code>def main() :</code> <code>code to execute</code> <i>declarations of other functions</i>	Loops: <code>for i in range(count):</code> <code>code to repeat</code> <code>while condition:</code> <code>code to repeat</code>
Names of the conditions: <code>front_is_clear()</code> <code>front_is_blocked()</code> <code>beepers_present()</code> <code>no_beepers_present()</code> <code>beepers_in_bag()</code> <code>no_beepers_in_bag()</code> <code>left_is_clear()</code> <code>left_is_blocked()</code> <code>right_is_clear()</code> <code>right_is_blocked()</code> <code>facing_north()</code> <code>not_facing_north()</code> <code>facing_south()</code> <code>not_facing_south()</code> <code>facing_east()</code> <code>not_facing_east()</code> <code>facing_west()</code> <code>not_facing_west()</code>	Function Declaration: <code>def name():</code> <code>code in the body of the function.</code> Extra Karel Commands: <code>paint_corner(COLOR_NAME)</code> <code>corner_color_is(COLOR_NAME)</code>