

**Project Report**

**เรื่อง**

Rat in a Maize

**เสนอ**

รศ.ดร.รังสิพรรณ มฤคทัต

**จัดทำโดย**

นายธนกฤต ชุติวงศ์ธนะพัฒน์ 6513112

นายภูรินท์ พงษ์พานิช 6513135

นายจารุภัทร โชติสิตานันท์ 6513161

นางสาวชลิษา บัวทอง 6513163

**รายงานนี้เป็นส่วนหนึ่งของรายวิชา Data Structure and Algorithm**

**ภาคเรียนที่ 2 ปีการศึกษา 2566**

**คำนำ**

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา Data Structure and Algorithm โดยจัดทำขึ้นเพื่อใช้ประกอบการอธิบายการทำงานของโปรแกรม Rat in a maze ซึ่งประกอบไปด้วยส่วนของคู่มือการใช้งานโปรแกรมเบื้องต้น และอธิบายในส่วนของ Code และ Algorithm รวมถึงข้อจำกัดต่างๆในการใช้งานโปรแกรม

ทางคณะผู้จัดทำขอขอบพระคุณ รศ.ดร.รังสิพรรณ มฤคทัต ผู้ให้ทั้งความรู้ และแนวทางการศึกษา สุดท้ายนี้ทางคณะผู้จัดทำหวังว่ารายงานฉบับนี้จะสามารถเป็นประโยชน์ไม่มากก็น้อยแก่ผู้อ่านทุกท่าน หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้และขออภัยมา ณ ที่นี้ด้วย

ขอขอบพระคุณ

คณะผู้จัดทำ

**สารบัญ**

**หัวข้อ**

**คำนำ**

**สารบัญ**

**เกี่ยวกับโปรแกรม**

**คู่มือการใช้งาน(User Manual)**

**โครงสร้างข้อมูล (Data Structure)**

**อัลกอริทึมที่ใช้ในการแก้ปัญหา**

**Algorithm Flow Chart**

**อธิบายการทํางานของโปรแกรม**

**ตัวอย่างแสดงการประมวลผล**

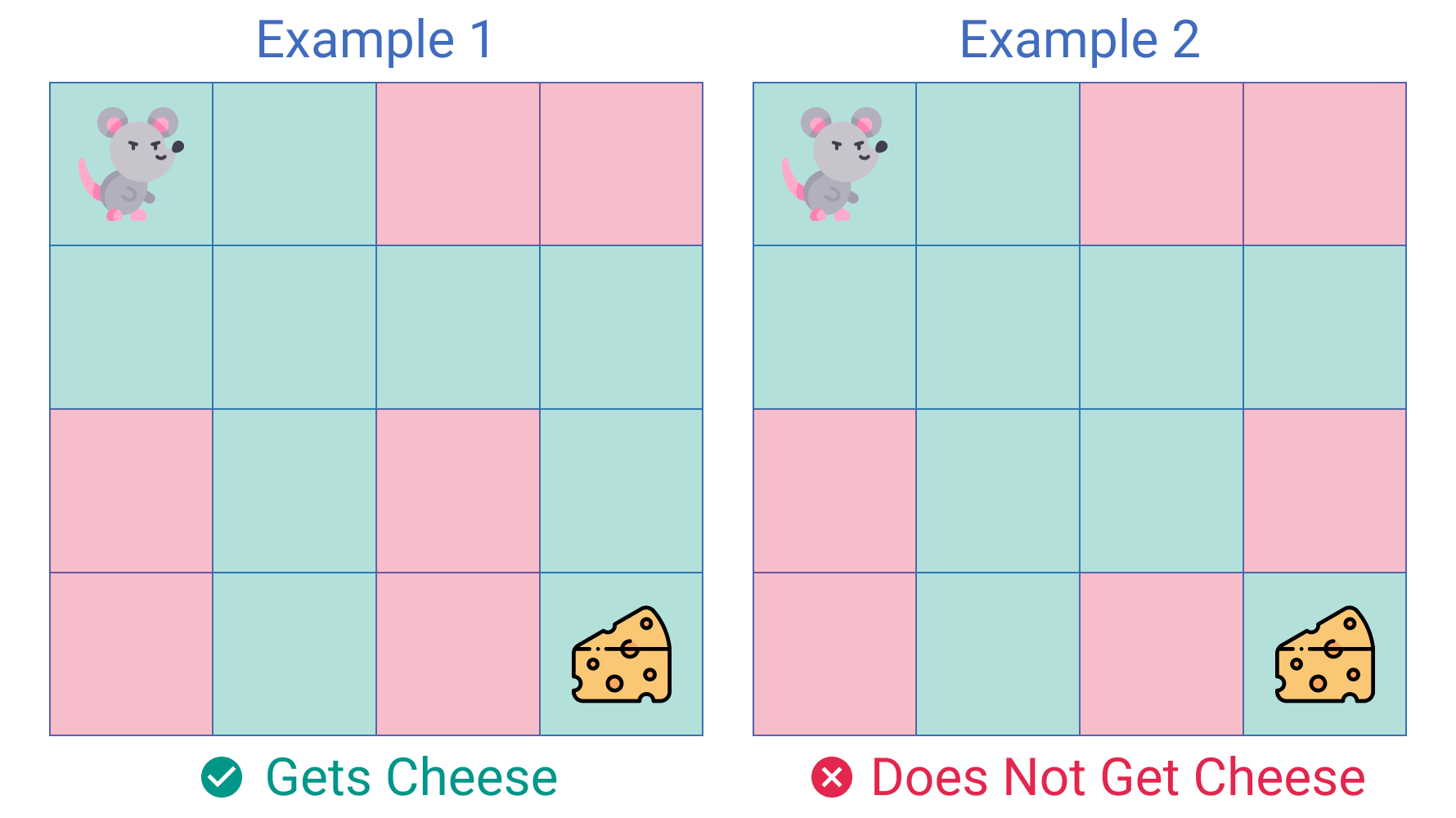
**Runtime ของโปรแกรม**

**ข้อจํากัดของโปรแกรม**

**แหล่งอ้างอิง**

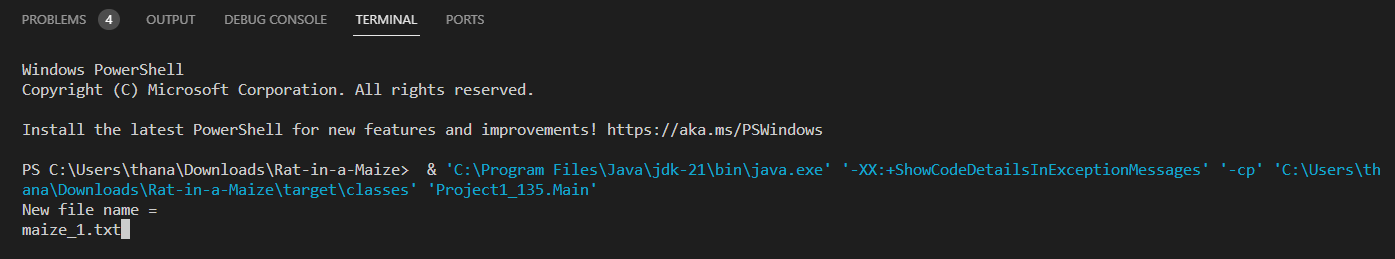
เกี่ยวกับโปรแกรม

โปรแกรม "Rat in a Maize" เป็นเกมที่ให้หนูเดินทางในแผนที่เพื่อหาอาหาร โดยใส่ทิศทางที่หนูจะเคลื่อนที่ไป และมีการใช้งานอัลกอริทึม DFS (Depth-First Search) เพื่อหาเส้นทางที่เหมาะสมในการไปยังอาหารในแผนที่ดังตัวอย่างต่อไปนี้



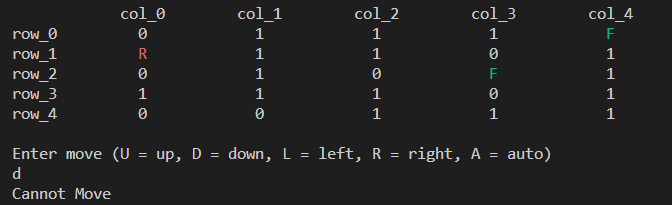
**คู่มือการใช้งาน(User Manual)**

1. ใส่ชื่อไฟล์เขาวงกตที่ต้องการเล่นแล้วกด Enter(หากใส่คำว่า “exit” จะทำการออกจากโปรแกรม)



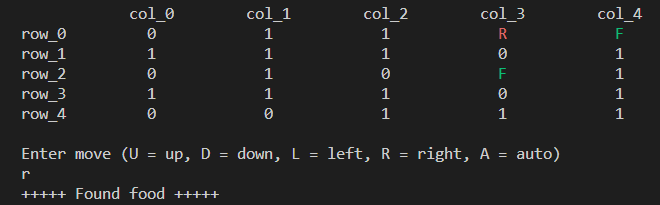
1. พิมพ์ทิศทางที่ต้องการให้หนูขยับ

U = ขยับขึ้น ,D = ขยับลง, L = ขยับไปทางซ้าย, R = ขยับไปทางขวา

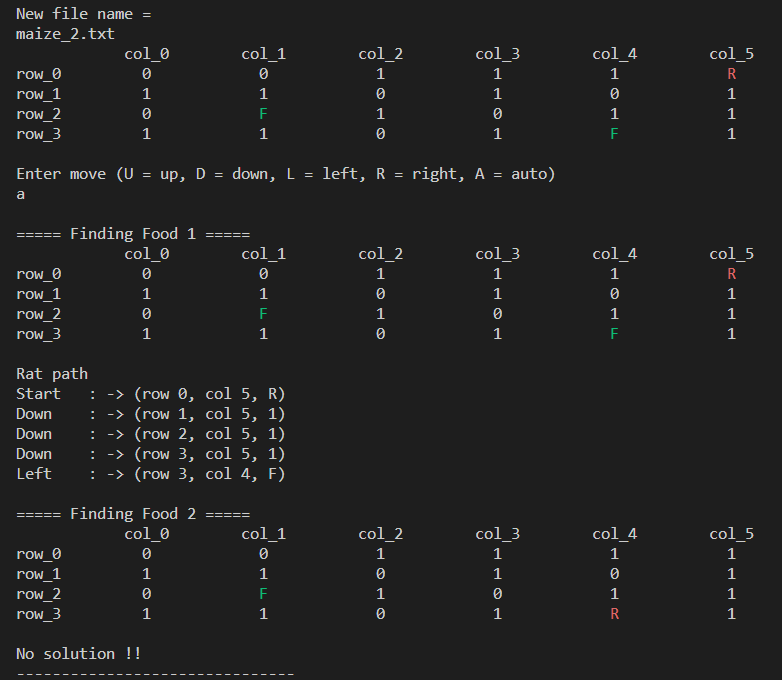
โดยหนูจะขยับไปได้แค่ช่อง 1(ground) หรือ F(ground with food) เท่านั้นและไม่สามารถขยับไปนอกmap หรือในช่อง 0 (wall)ได้ โปรแกรมจะบอก “Cannot Move”

1. เมื่อหนูขยับจนมาเจออาหารโปรแกรมจะขึ้นว่า

+++++Found food+++++



1. เมื่อกด A หนูจะหาอาหารเองโดยอัตโนมัติ โดยจะเขียนเส้นทางการเดินตามในรูปภาพ แต่ถ้าหนูหาอาหารไม่เจอ โปรแกรมจะขึ้นว่า No solution !!



**Data Structure**





ในไฟล์ Command.java, การ import คลาส ArrayDeque, ArrayList, และ Collections ใช้เพื่อให้สามารถใช้งานคลาสและเมทอดที่เกี่ยวข้องในโค้ดได้โดยไม่ต้องระบุชื่อคลาสทั้งหมด File ใช้ในการระบุตำแหน่งและชื่อไฟล์, Scanner ใช้ในการอ่านข้อมูลจากไฟล์ที่ถูกเลือกในโปรแกรมนี้

* ArrayDeque (java.util.ArrayDeque):

-เหตุผลในการใช้: ใช้เพื่อเก็บเส้นทางที่หนูได้ค้นหาในแบบของเขตข้อมูล (queue) ArrayDeque มีการจัดการขนาดได้ดีและมีประสิทธิภาพสูงในการเพิ่มและลบจากทั้งสองข้างของ queue

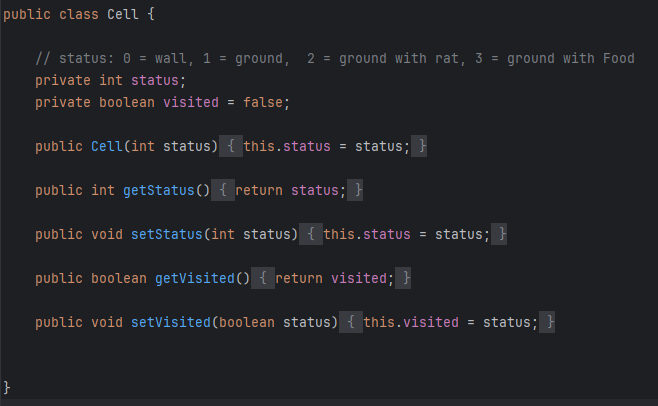
* ArrayList (java.util.ArrayList):

-เหตุผลในการใช้: ใช้สำหรับเก็บข้อมูลของแผนที่และเส้นทางที่หนูได้ทำหารค้นห. ArrayList เป็นโครงสร้างข้อมูลที่สามารถขยายขนาดได้ตามต้องการและมีการเข้าถึงข้อมูลที่รวดเร็ว

* Collections (java.util.Collections):

-เหตุผลในการใช้: ใช้สำหรับการ shuffle ทิศทางการเคลื่อนที่ที่ของหนู

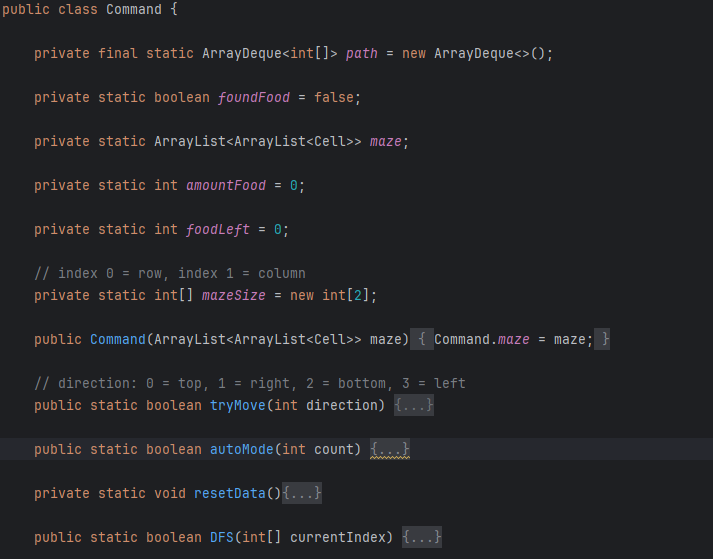
โครงสร้างข้อมูลใน Class Cell

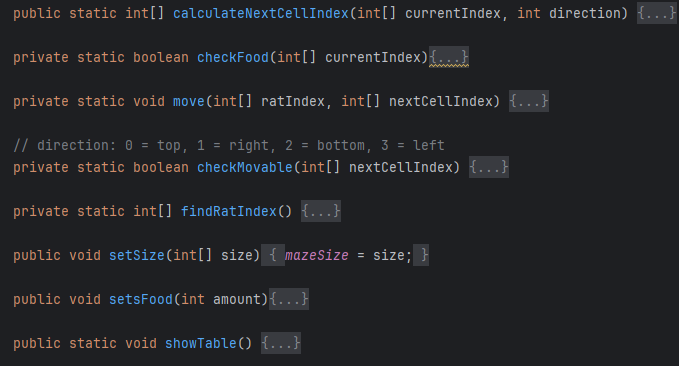


Class Cell เป็น class ที่ทำการเก็บข้อมูลของเซลล์นั้นๆ ซึ่งเก็บค่า status กับ visited ค่า status บ่งบอกถึง ว่า cell นั้นๆเป็นอะไรเช่น 0 คือwall, 1 คือ ground, 2 คือ ground with rat และ 3 คือ ground with food โดย class cell มี 4 method ดังนี้

1. getStatus() :เป็นฟังก์ชันที่ทำการเซตค่า status จาก parameter
2. setStatus() :เป็นฟังก์ชันที่คืนค่าข้อมูล status
3. getVisited() :เป็นฟังก์ชันที่คืนค่าข้อมูล visited
4. setVisited() : เป็นฟังก์ชันที่ทำการเซตค่า visited จาก parameter

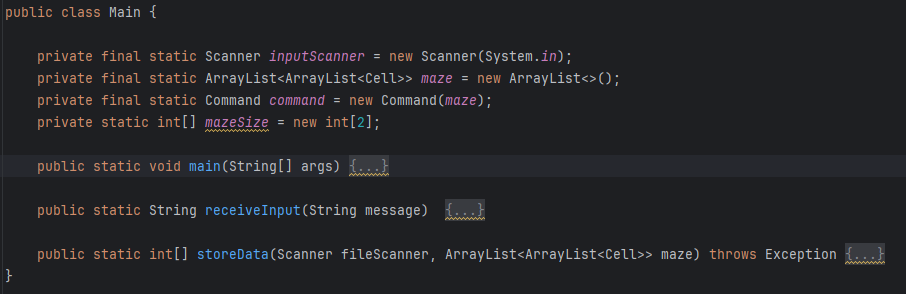
โครงสร้างข้อมูลใน Class Command





* 1. tryMove(): เป็นฟังชันที่ใช้ตรวจสอบความเป็นไปได้ของการเคลื่อนที่และ ทำการเปลี่ยนสถานะของเซลล์ที่หนูอยู่
* 2. autoMode():เป็นฟังชันที่ใช้อัลกอริทึม DFS ในการค้นหาเส้นทางไปยัง อาหาร และทำการบันทึกเส้นทางที่หนูเคลื่อนที่
* 3. DFS():เป็นฟังก์ชันที่ใช้อัลกอริทึม DFS ในการค้นหาเส้นทางไปยังอาหาร, และทำการเพิ่ม/ลบเซลล์ลงใน path ที่พบ
* 4. calculateNextCellIndex(int[ ] currentIndex, int direction):เป็นฟังก์ชันที่ใช้คำนวณและคืนค่าดัชนีของเซลล์ถัดไปขึ้นอยู่กับทิศทางที่กำหนด.
* 5. checkFood():เป็นฟังก์ชันที่ใช้ตรวจสอบว่าเซลล์ปัจจุบัน มีอาหารหรือไม่
* 6. move():เป็นฟังก์ชันที่เคลื่อนที่หนูจากเซลล์ปัจจุบันไปยังเซลล์ถัดไป
* 7. checkMovable(int[ ] nextCellIndex):เป็นฟังก์ชันที่คืนค่า true ถ้าหนู สามารถเคลื่อนที่ไปยังเซลล์ถัดไปได้, คืนค่า false ถ้าไม่สามารถเคลื่อนที่ได้
* 8. findRatIndex():เป็นฟังก์ชันที่ค้นหาตำแหน่งของเซลล์ที่มีสถานะเป็น "ground with rat"
* 9. resetData():เป็นฟังก์ชันที่รีเซ็ตข้อมูลหลังจากการค้นหาเส้นทาง
* 10. setSize(): เป็นฟังก์ชันที่กำหนดขนาดของแผนที่
* 11. setsFood(): เป็นฟังก์ชันที่เซตค่า จำนวนอาหารที่มีในแผนที่
* 12. showTable(): เป็นฟังก์ชันที่จะแสดงตาราง

โครงสร้างข้อมูลใน Class Main



Class main คือ class หลักของโปรแกรมจำทำการรับ input จาก user ละมีการเก็บ object ของ file อื่นๆ โดน class main มี 2 method

1. receiveInput () : เป็นฟังก์ชันที่รับค่าจาก user
2. storeData(): เป็นฟังก์ชันที่นำค่าจากการอ่านไฟล์ไปเก็บ

**อัลกอริทึมที่ใช้ในการแก้ปัญหา**

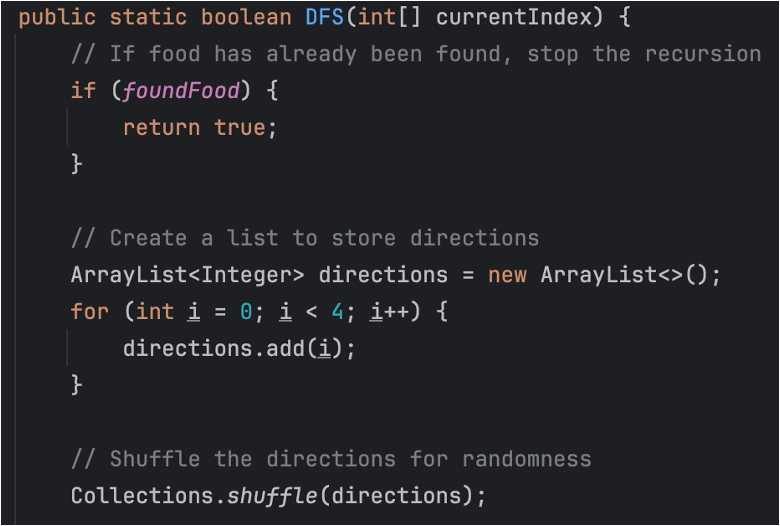
**วิธีการเช็คว่าหนูจะเดินไปในทิศทางไหนได้คือ ?**

ทำการใช้ Depth-First Search (DFS) เพื่อค้นหาเส้นทางที่หนูสามารถที่จะไปหาอาหารได้ โดยเราเขียนฟังก์ชันโค้ดให้มีการรับ Array ชื่อ ‘currentIndex’นั่นก็คือตำแหน่งปัจจุบันของหนูที่อยู่ในเขาวงกต โดยหลักการทำงานภายในฟังก์ชันจะทำงานดังนี้

1. ทำการตรวจสอบว่าเจออาหารไหม (‘foundFood’ is a gobal boolean variable)ถ้าเจอจะหยุดการ Recursion แล้วคืนค่าเป็น True (สั่งให้ทุก recursion ยกเลิกการทำงานเพราะเจออาหารแล้ว Backtracking)

2. ทำการสร้าง ArrayList ที่มีชื่อว่า directions ที่เก็บค่า directions

( 0 = top, 1 = right, 2 = bottom, 3 = left) หลังจากนั้นทำการสลับค่าที่เก็บใน ArrayList เสมือนเป็นการสุ่มทิศทางการเดินของหนู



3. เซ็ตค่าไว้ว่าตำแหน่งที่หนูอยู่ ณ ปัจจุบันมีการ Visited แล้ว

4.สำรวจเส้นทางที่หนูสามารถเคลื่อนที่ไปได้โดยใช้ฟังก์ชัน“calculateNextCellIndex” โดยรับค่าเป็นตำแหน่งปัจจุบันของหนู (currentIndex) กับทิศทางที่หนูจะเดินไป (dir)

5. เช็คว่าหนูสามารถเดินไปทิศทางที่สุ่มได้ไหมโดยใช้ฟังก์ชัน“checkMovable”

ถ้าสามารถเคลื่อนที่ไปได้และหากยังไม่ได้ Visited จะเพิ่ม currentIndex เข้าไปในpath ที่เป็น ArrayDeque แล้วจึงทำการเคลื่อนที่

6. ถ้า Cell ที่เคลื่อนที่เข้าไปมีอาหารจากการเช็คโดยใช้ฟังก์ชัน“checkFood” จะทำการเพิ่ม Cell นั้นเข้าไปที่ path และเซ็ตตัวแปร “foundFood” เป็นค่า True และหยุดการ Recursion

(สั่งให้ทุก recursion ยกเลิกการทำงานเพราะเจออาหารแล้ว Backtracking)

7. ทำการเรียก function ตัวเองเพื่อทำการ Recursive ไปเรื่อยๆ

8.หาก Recursive แล้วไม่เจออาหารเลย จะทำการลบ cell ตัวเองออกจาก path และส่งค่า False ให้ Recursive ชั้นก่อนหน้า



### หลักการทำงานของ DFS

### Forwarding Steps:

* 1.การกำหนดทิศทาง:

-สุ่มลำดับของทิศทาง (up, right, down, left) ที่หนูจะต้องทำการค้นหา

-ทิศทางถูกเก็บใน directions เพื่อให้การค้นหาเป็นไปตามลำดับที่สุ่ม

* 2.การตรวจสอบทิศทาง:

-ทำการตรวจสอบทุกทิศทางที่ถูกสุ่ม, หากทิศทางนั้นเป็นเส้นทางที่เคลื่อนที่ได้ (movable) และยังไม่เคยเจอเซลนี้มาก่อน ให้ทำการเลือกทิศทางนั้น

* 3.การเคลื่อนที่:

-ทำการเคลื่อนที่ไปยังเซลถัดไปตามทิศทางที่เลือก

-หากเจออาหาร, ทำการบันทึกเซลที่มีอาหารลงใน path และตั้งค่า foundFood เป็น true

* 4.การเรียก DFS แบบ loop:

-เรียกตัวเอง (DFS) โดยให้เซลปัจจุบันเป็นเซลที่เคลื่อนที่ไปแล้ว

### Backtracking Steps:

* 1.การ Backtracking:

-หากหนูเจอเส้นทางที่ไม่สามารถก้าวไปข้างหน้าได้, หรือเจอเซลที่ทางนี้ไม่สามารถเคลื่อนที่ได้, ให้ทำการถอยกลับ (backtrack) ไปยังเซลก่อนหน้าโดยการ return false(กรณีไม่เจออาหาร)

* 2.การลบเส้นทาง:

- ทำการลบเซลล่าสุดที่ถูกเพิ่มเข้าไปใน path

### Finding Next Food:

* การค้นหาเส้นทางไปยังอาหารถัดไป:

-หลังจากหนูเคลื่อนที่ไปยังอาหารแล้ว ทำการลบเซลที่มีอาหารออกจาก path

* การตั้งค่าพารามิเตอร์ใหม่:

-ทำการตั้งค่า foundFood เป็น false เพื่อให้หนูเริ่มการค้นหาใหม่ได้

### How the Algorithm Concludes No Solution:

หาก DFS algorithm ได้ทำการตรวจสอบทุกเซลล์แล้วไม่มีทิศทางใดที่เป็นไปได้จะทำการ return false ทุกๆชั้นจนชั้นแรก return false และบอกโปรแกรมว่าไม่พบอาหาร

### Recursive Calls:

* การเรียกตัวเองซ้ำ (DFS):

-เมื่อมีทิศทางที่เป็นไปได้ ให้ทำการเรียกตัวเอง (DFS) ไปยังเซลถัดไป

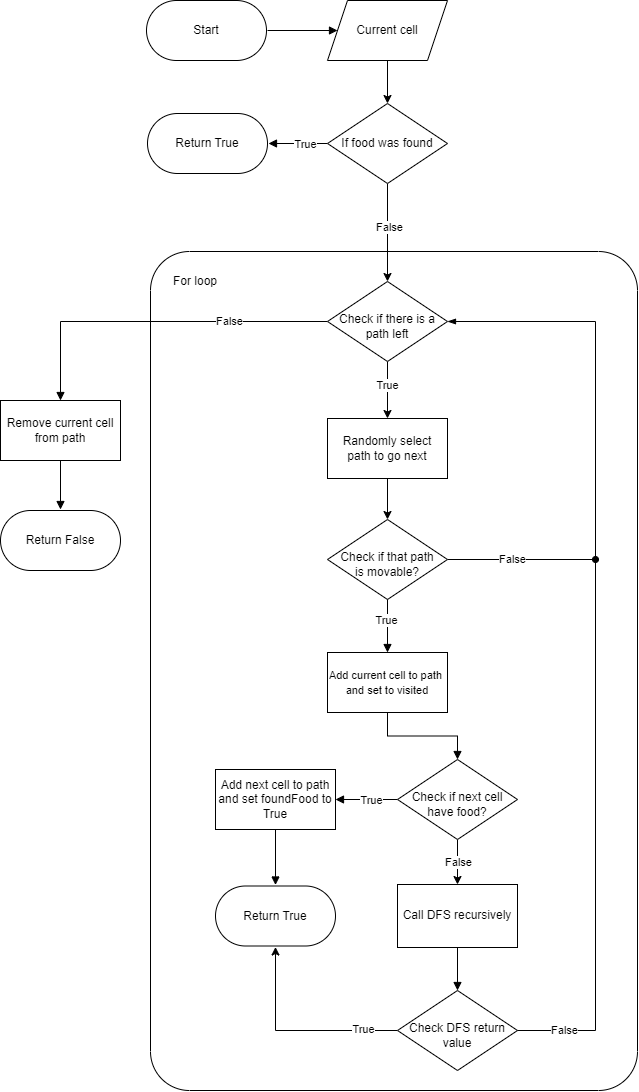
* การจัดการสถานของ cell:

-ในการเรียกแต่ละครั้ง จะทำการเพิ่มตำแหน่ง cell ที่ผ่านไปลงใน path

-หากเจอเซลที่มีอาหาร ใหเทำการบันทึกเซลที่มีอาหารลงใน path และตั้งค่า foundFood เป็น true

นอกจากนี้, มีการใช้ ArrayDeque เพื่อเก็บทางที่ผ่านไป path

Algorithm Flowchart

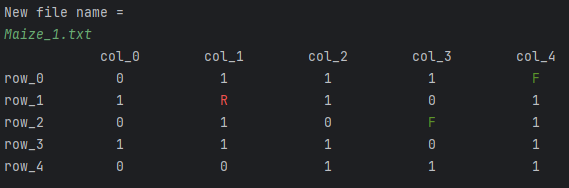


**DEMO**

ตัวอย่างการแสดงผล

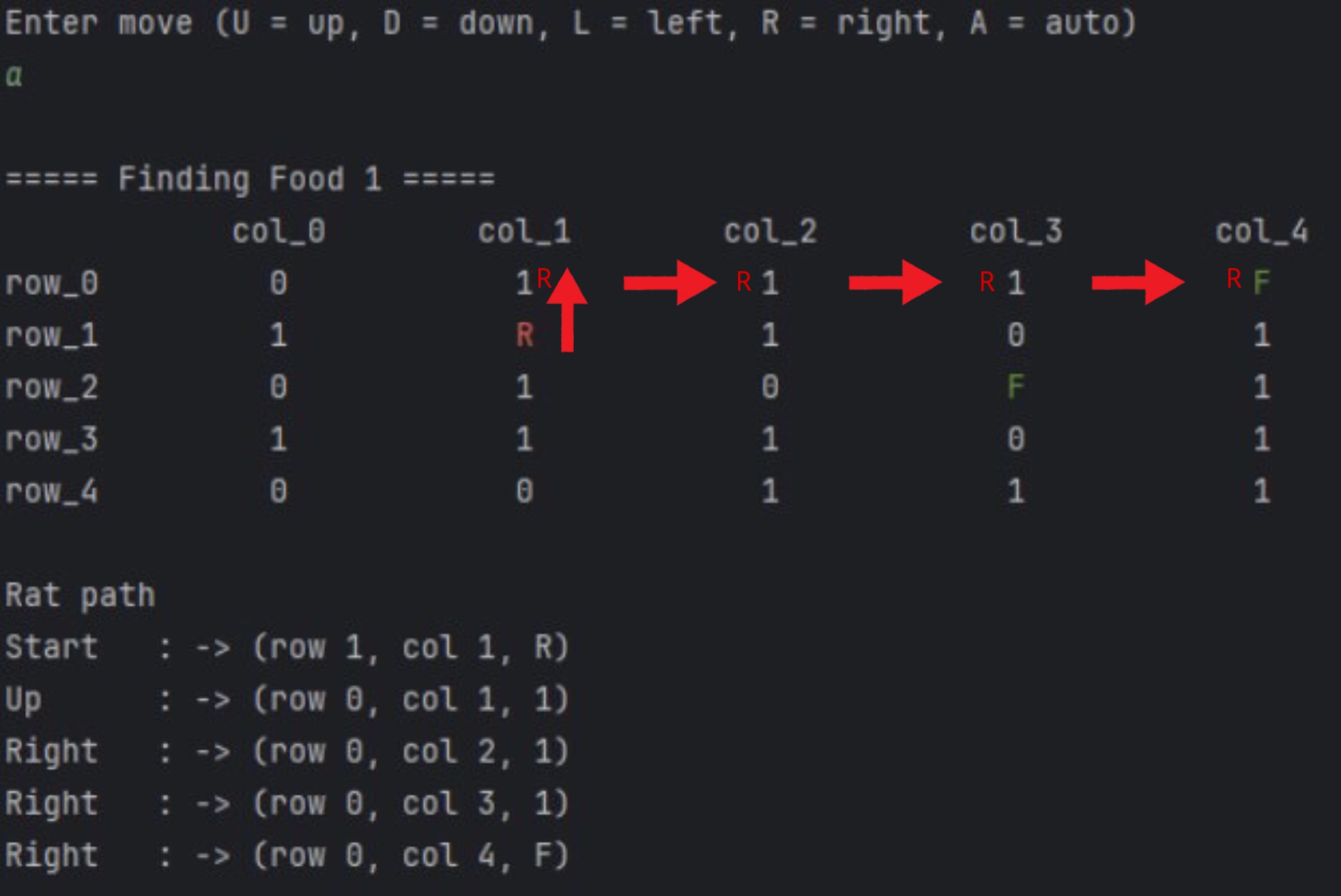
1. ผู้ใช้สามารถเริ่มทำงานโปรแกรมได้โดยการพิมพ์ชื่อไฟล์เพื่อเลือก maize ที่ต้องการ
2. โปรมแกรมจะแสดงตำแหน่งหนูโดยแทนด้วย (R สีแดง) แสดงตำแหน่งอาหารด้วย F(สีเขียว) เลข 1 และเลข 0 คือ พื้นดิน และกำแพง ตามลำดับ

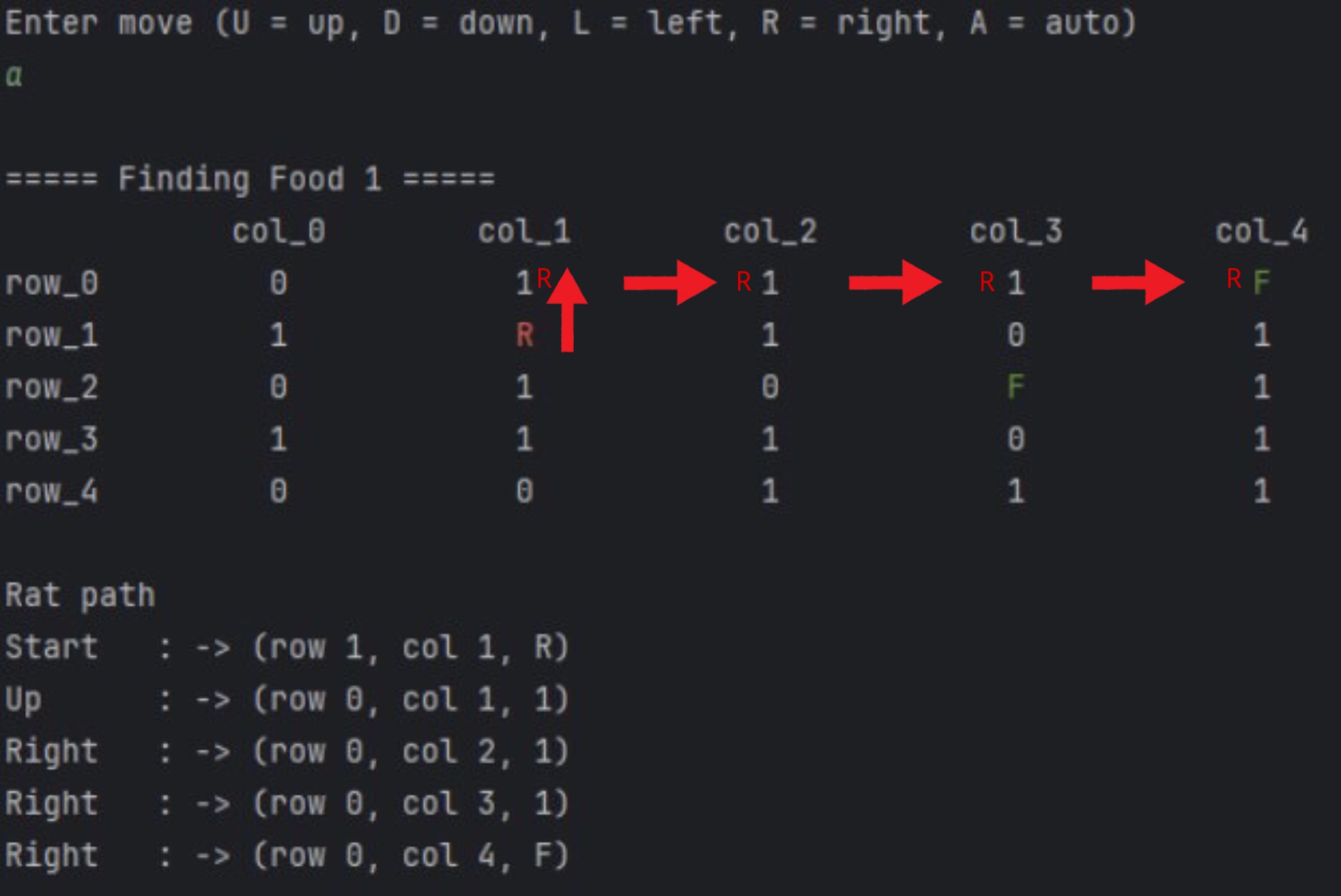
ตัวอย่างที่1 (Maize\_1.txt)

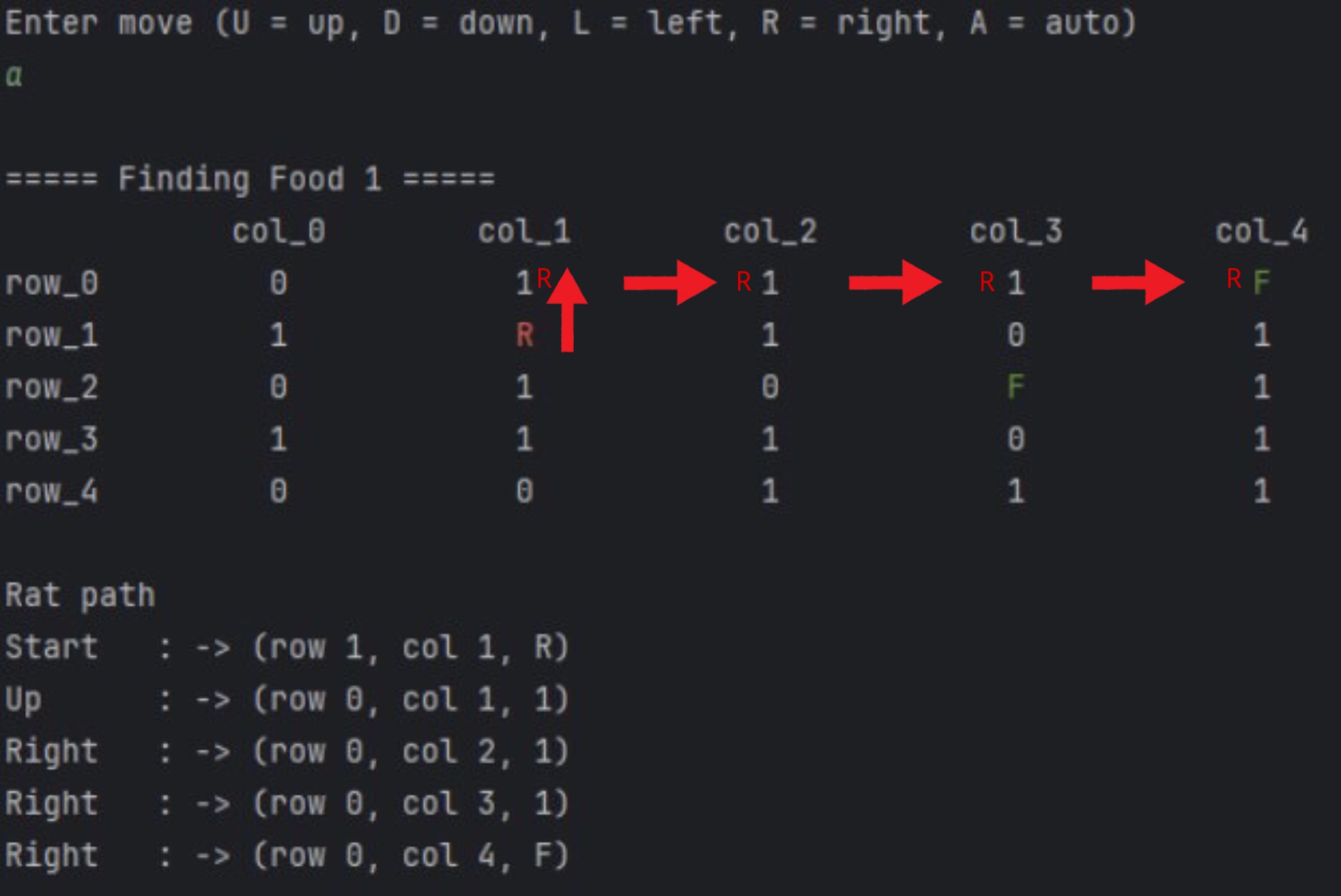


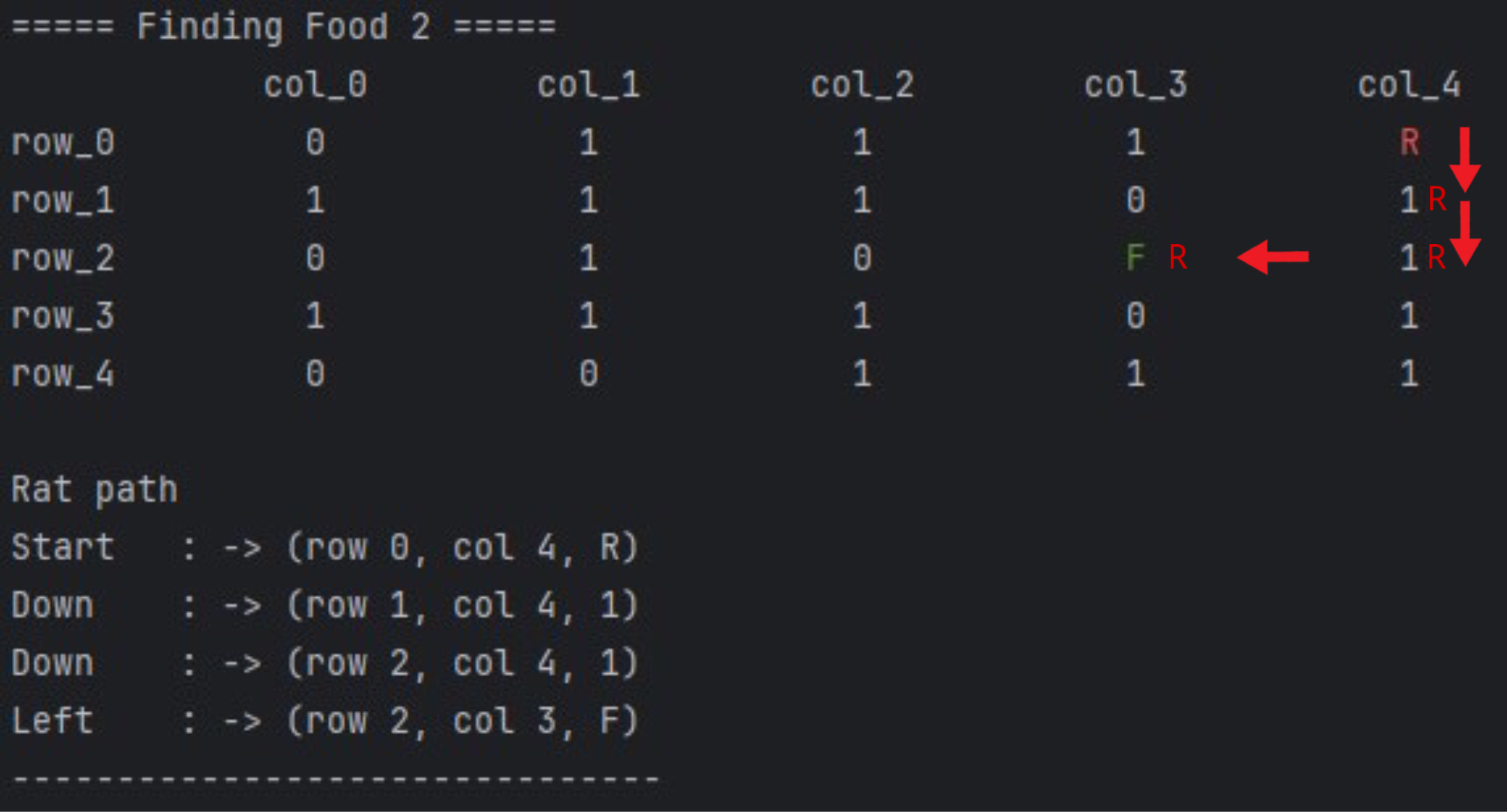
ตัวอย่างการใช้งาน Auto Mode ( ผู้ใช้พิมพ์ a เพื่อเข้าสู่ Auto Mode)

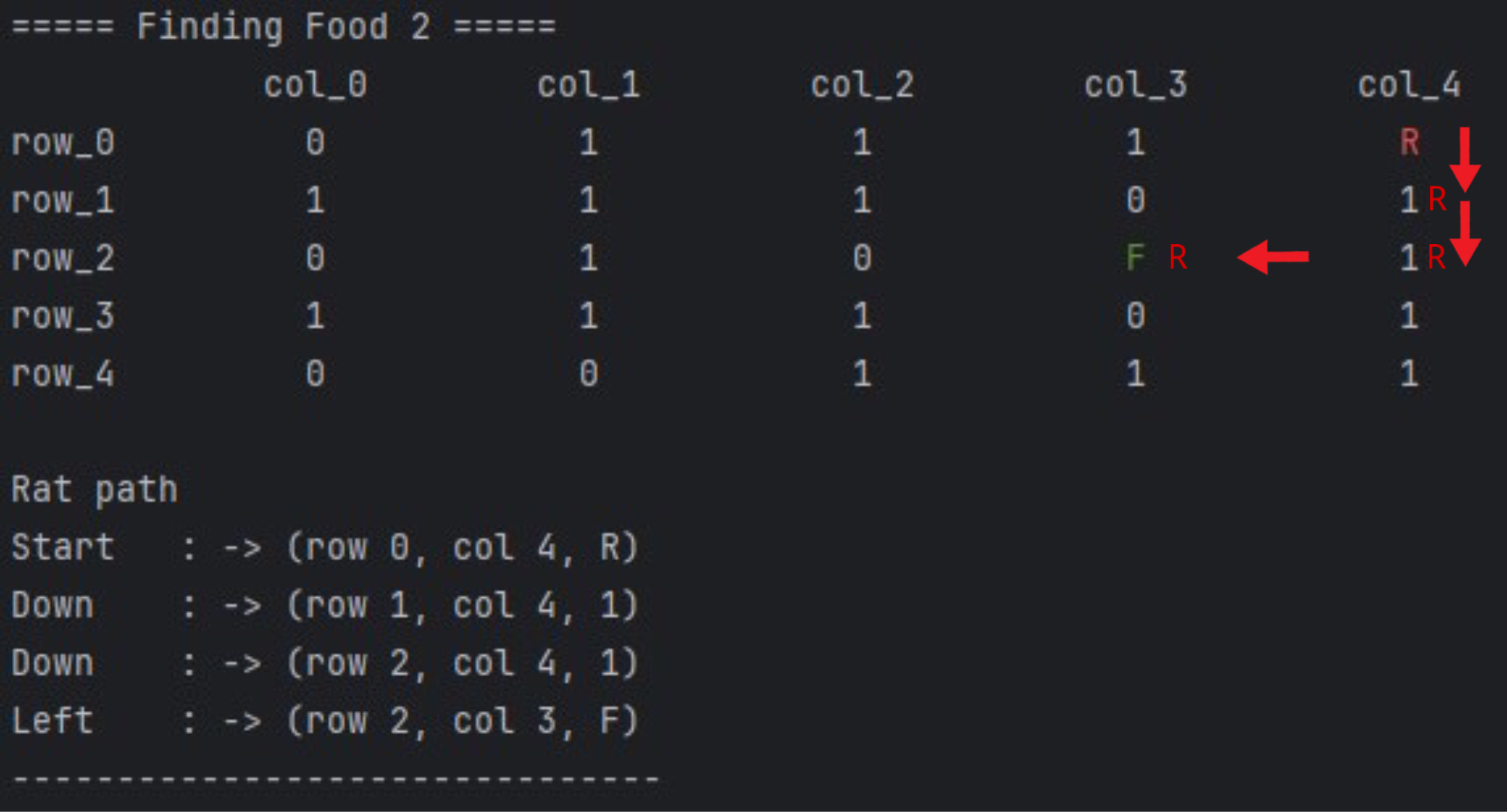
เส้นทางที่ rat เดิน(แบบไม่โชว์ ทางที่ตัน)





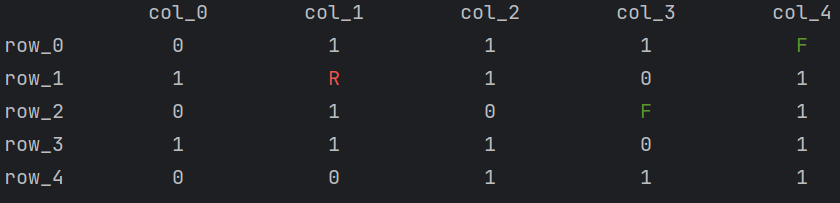


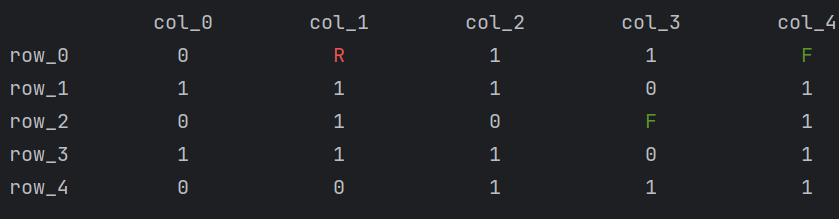


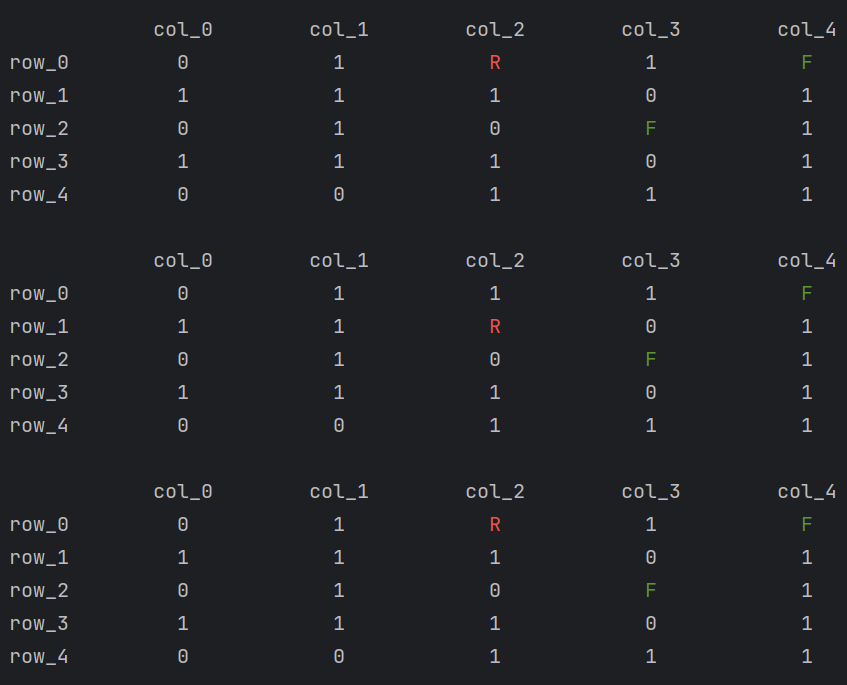


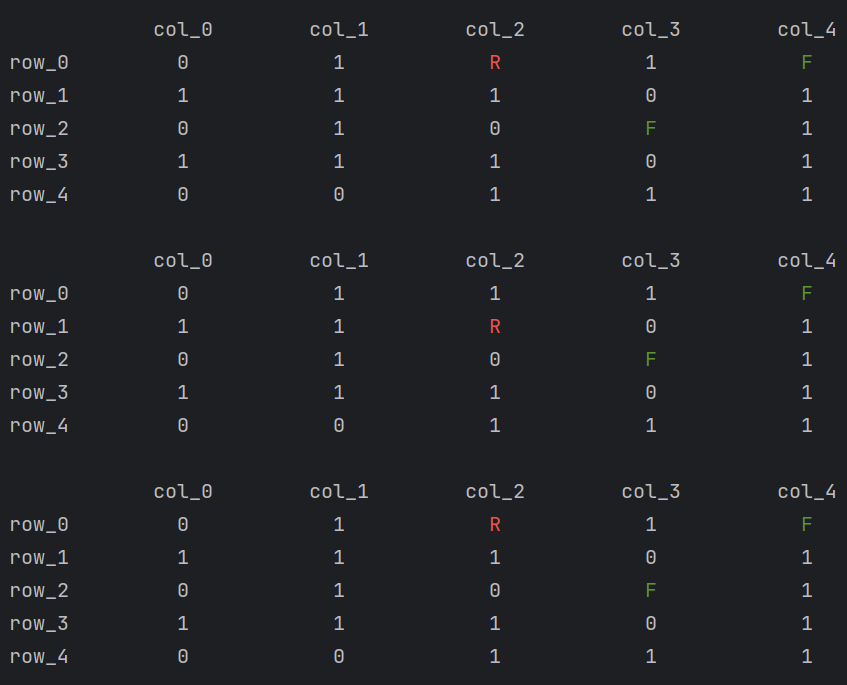
ใน Auto mode โปรแกรมจะทำการเรียกใช้ DFS เพื่อค้นหาเส้นทางจากตำแหน่งปัจจุบันของหนูไปยังอาหาร หากพบเส้นทางโปรแกรมจะแสดงทำการ path ดังที่แสดงในรูป จากนั้นโปรแกรมจะทำการตรวจสอบว่ามีอาหารเหลืออยู่หรือไม่ หากพบว่ามีอาหารเหลือโปรแกรมจะทำการวนลูปเพื่อหาตำแหน่งอาหารแล้วแสดง path อีกครั้ง

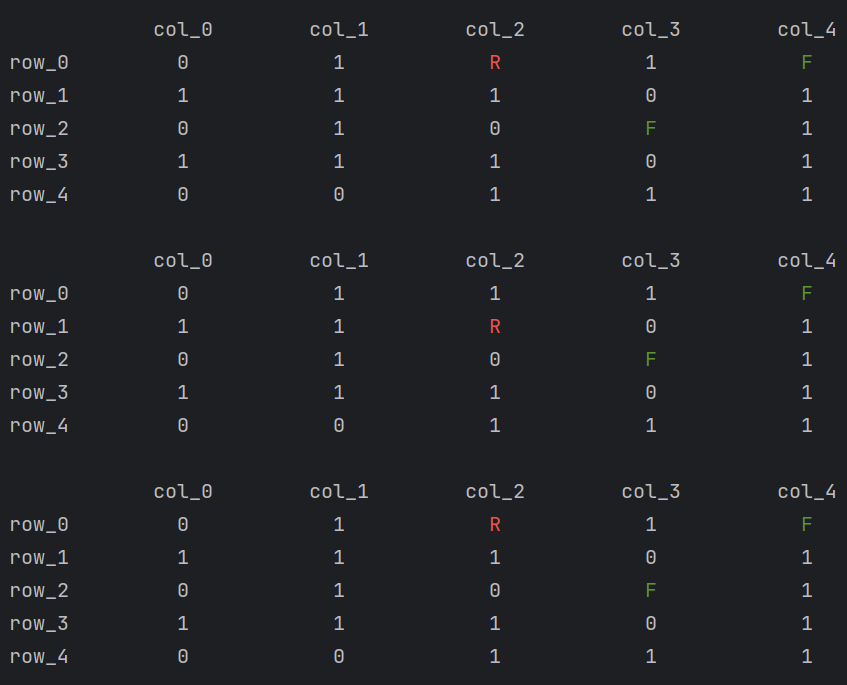
เส้นทางที่ rat finding food 1 เดิน(แบบแสดงทุกเส้นทาง)

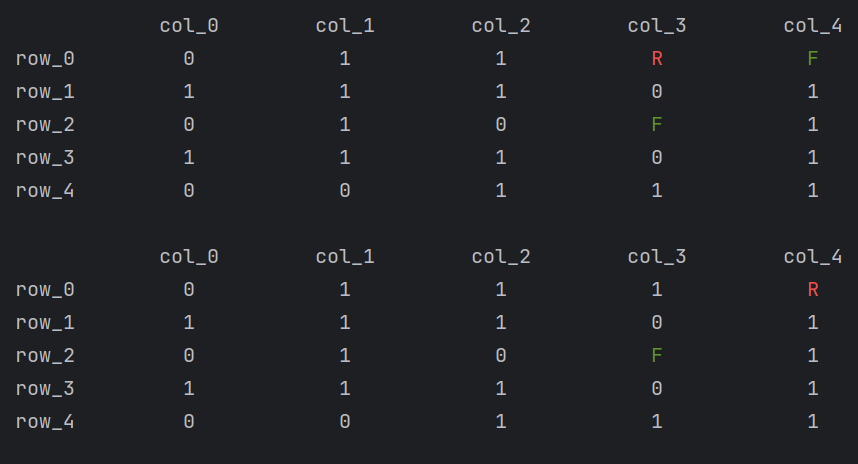


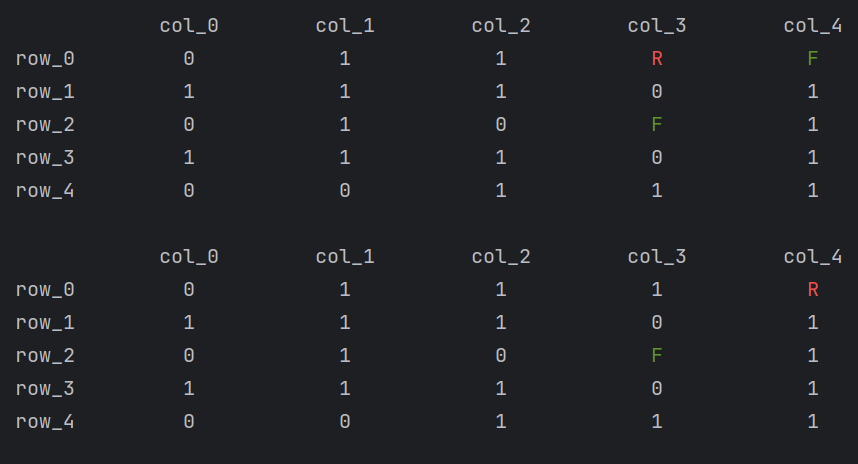








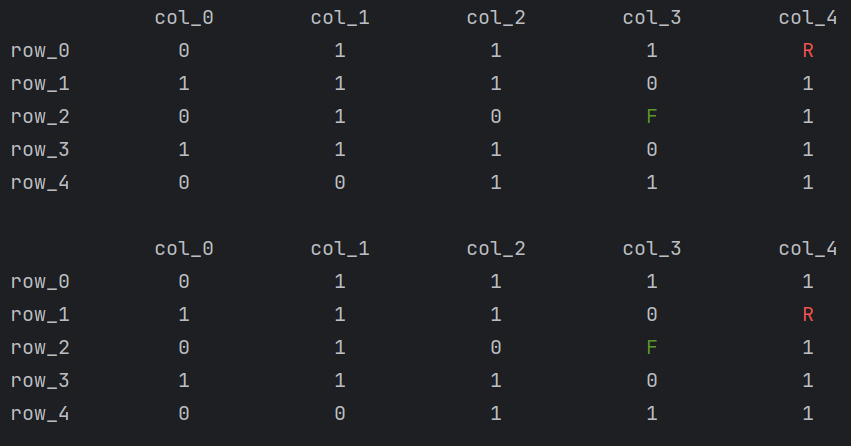


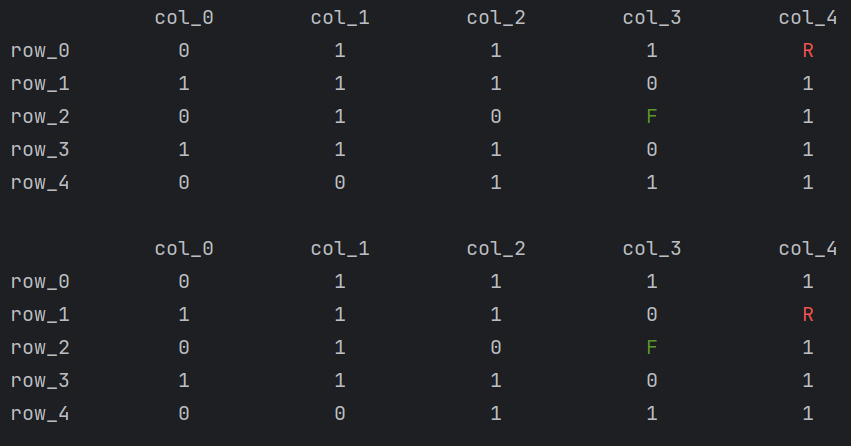


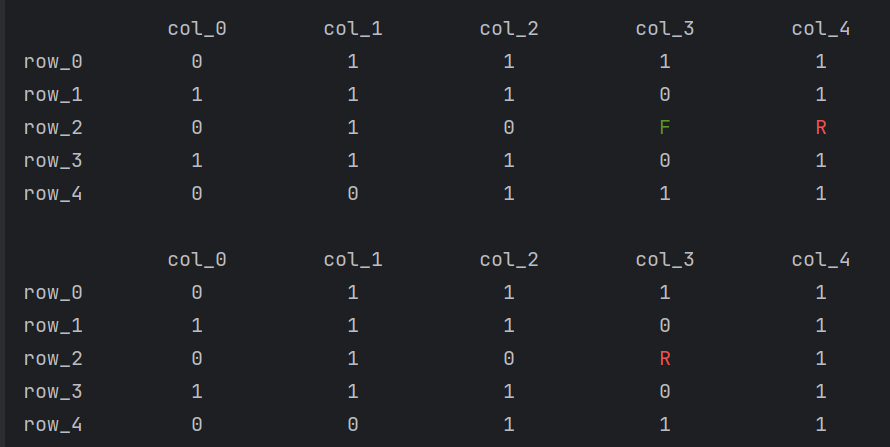
ตารางเส้นทางที่ rat finding food 1

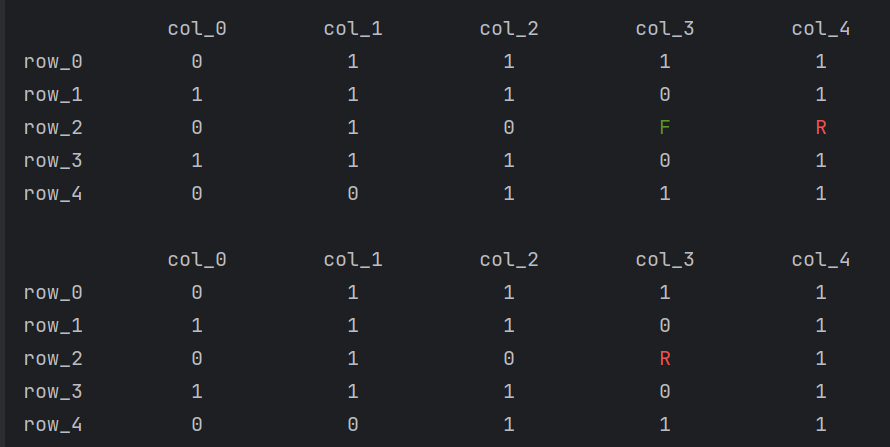
| Recursive Layer | Stack | Push/Pop |
| --- | --- | --- |
| 1 | up | Push up |
| 2 | up right | Push right |
| 3 | up right down | Push down |
| 2 | up right | pop |
| 3 | up right right | Push right |
| 4 | up right right right | Push right |

เส้นทางที่ rat finding food 2 เดิน(แบบโชว์ ทุกทาง)





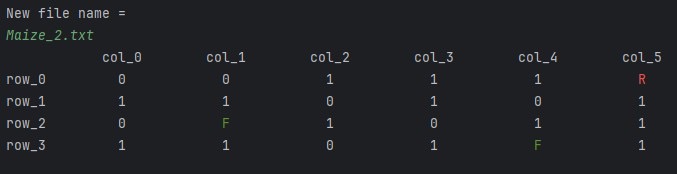




ตารางเส้นทางที่ rat finding food 2

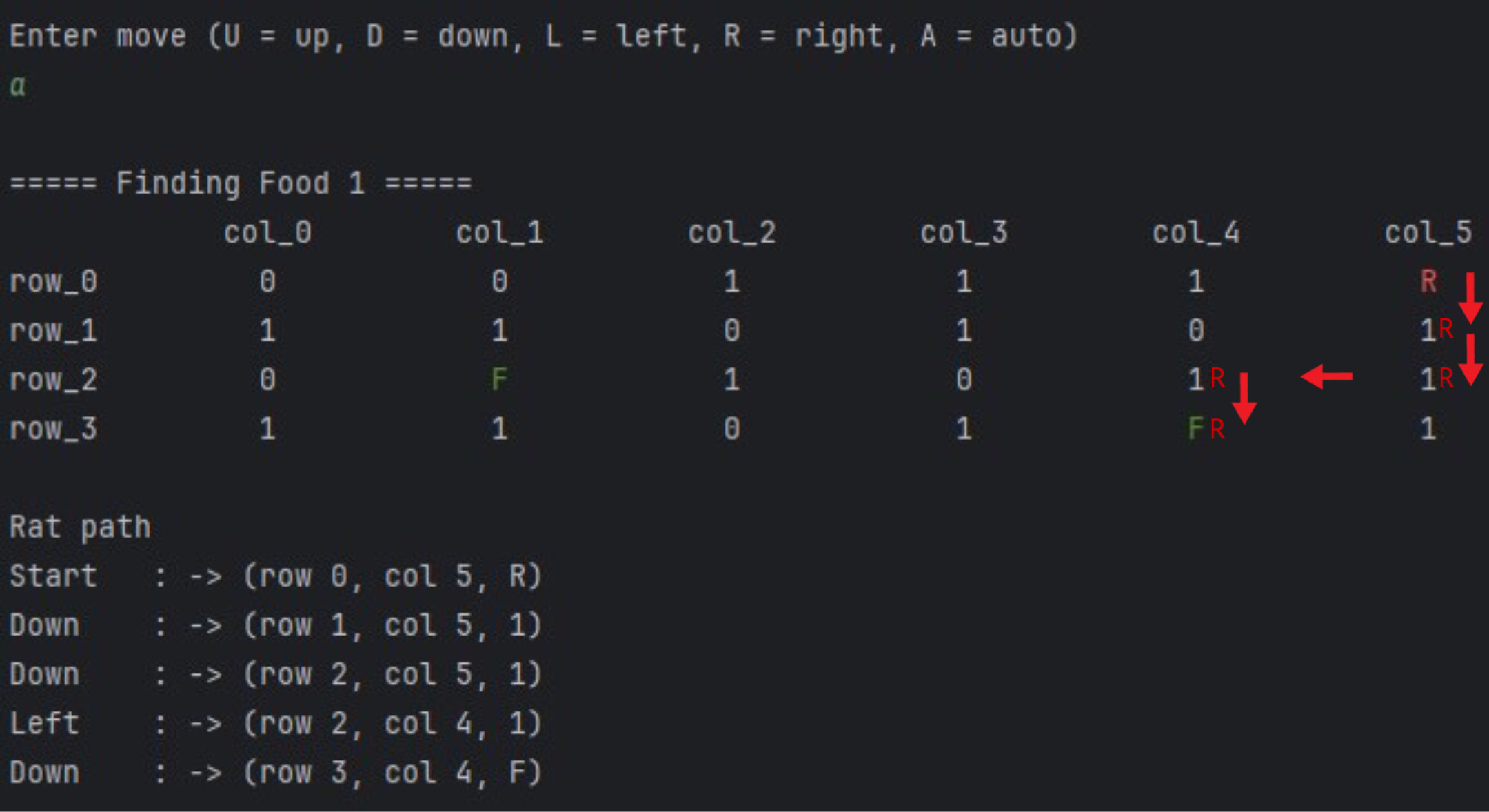
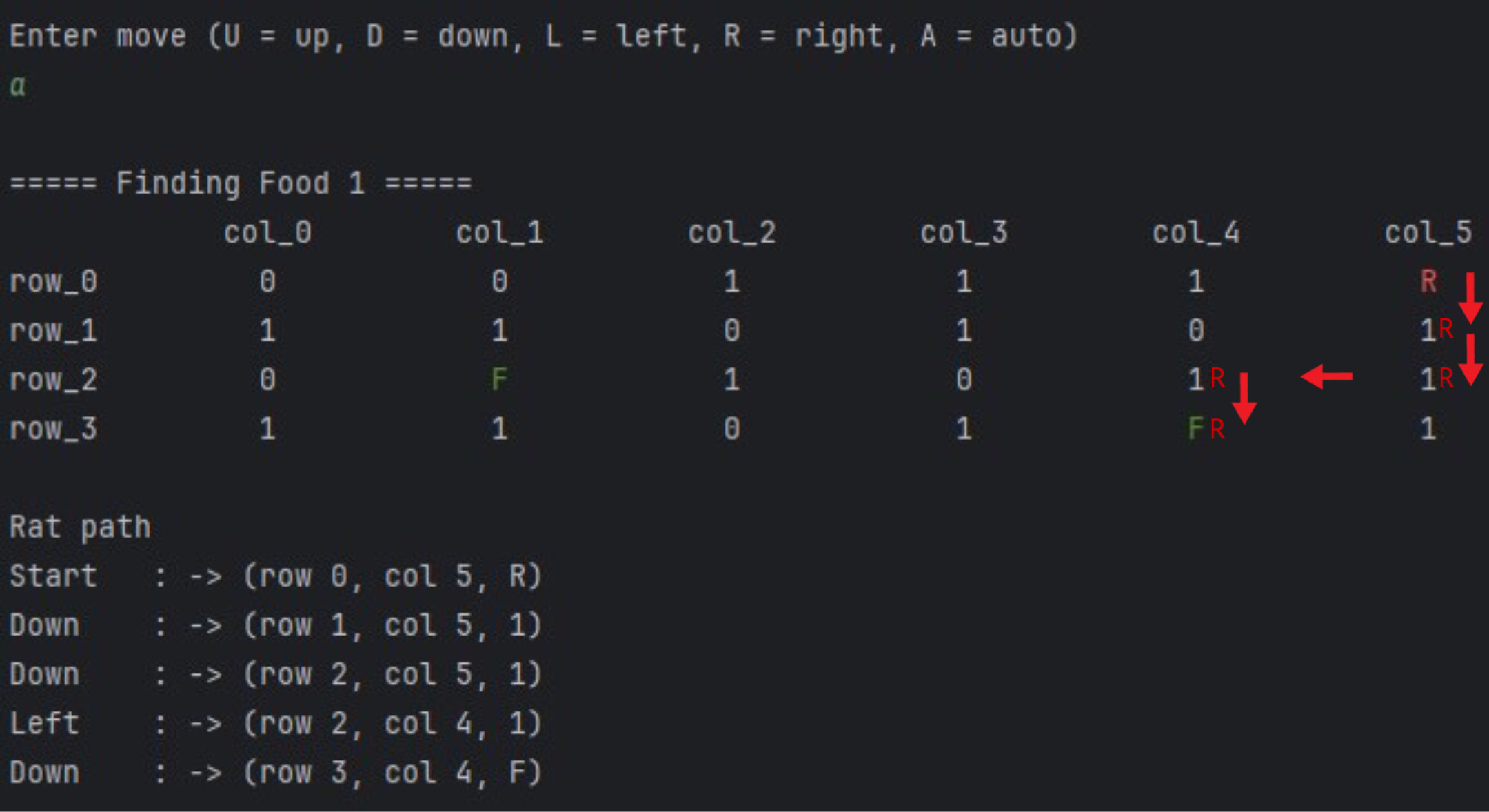
| Recursive Layer | Stack | Push/Pop |
| --- | --- | --- |
| 1 | down | Push down |
| 2 | down down | Push down |
| 3 | down down left | Push left |

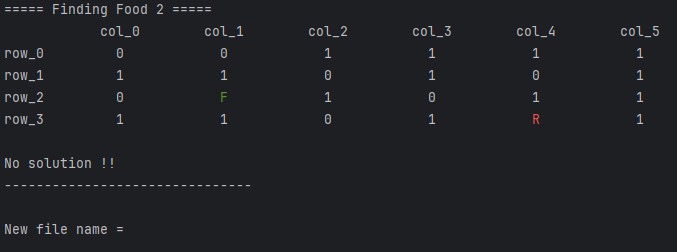
ตัวอย่างที่ 2 ( Maize\_2.txt)

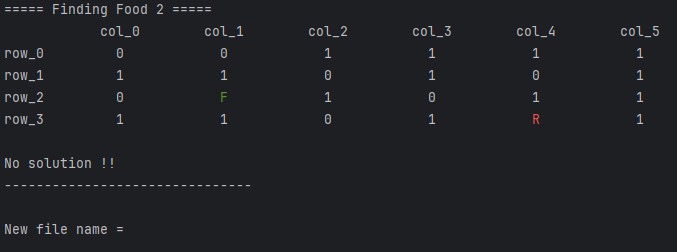


ตัวอย่างการใช้งาน Auto Mode ( ผู้ใช้พิมพ์ a เพื่อเข้าสู่ Auto Mode)

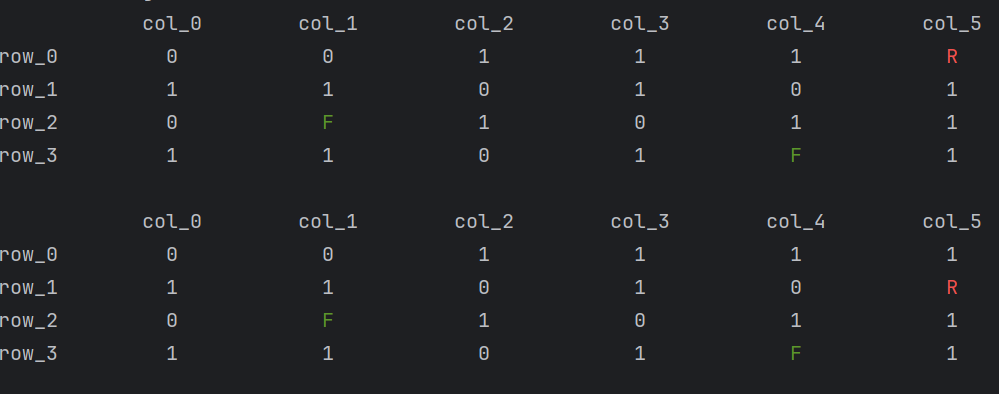
เส้นทางที่ rat เดิน(แบบไม่แสดงทางที่ตัน)

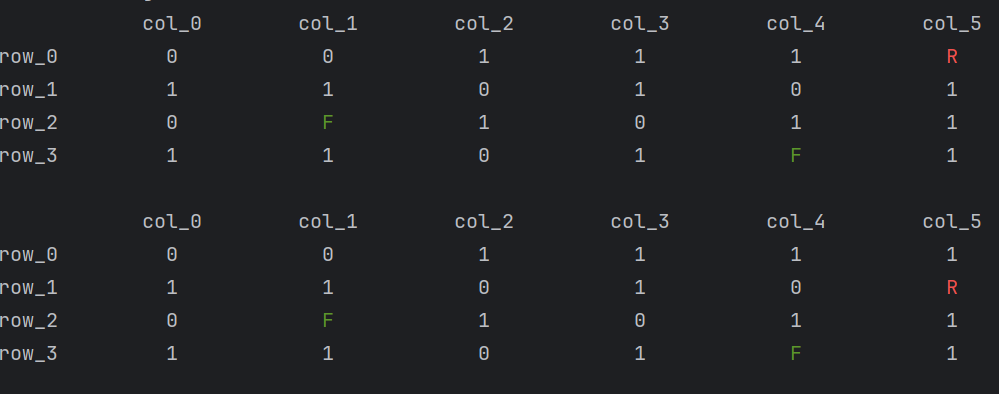


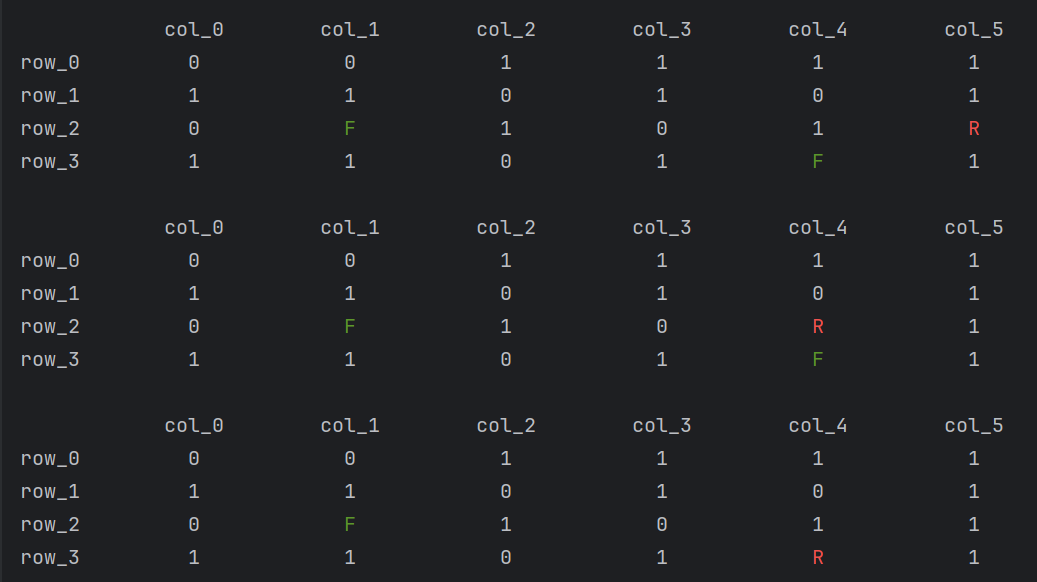


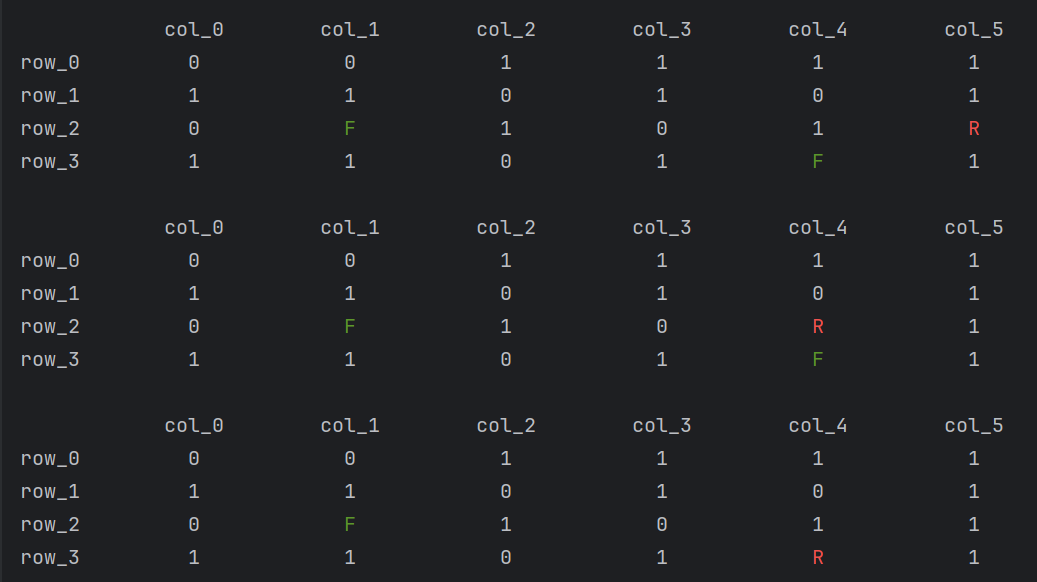


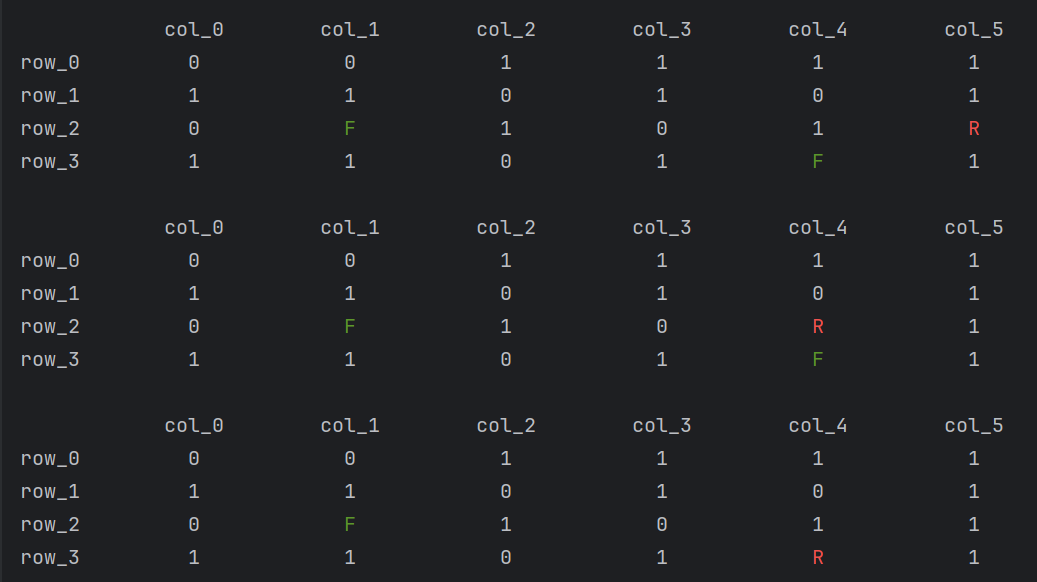
เส้นทางที่ rat finding food 1 เดิน(แบบแสดงทุกเส้นทาง)







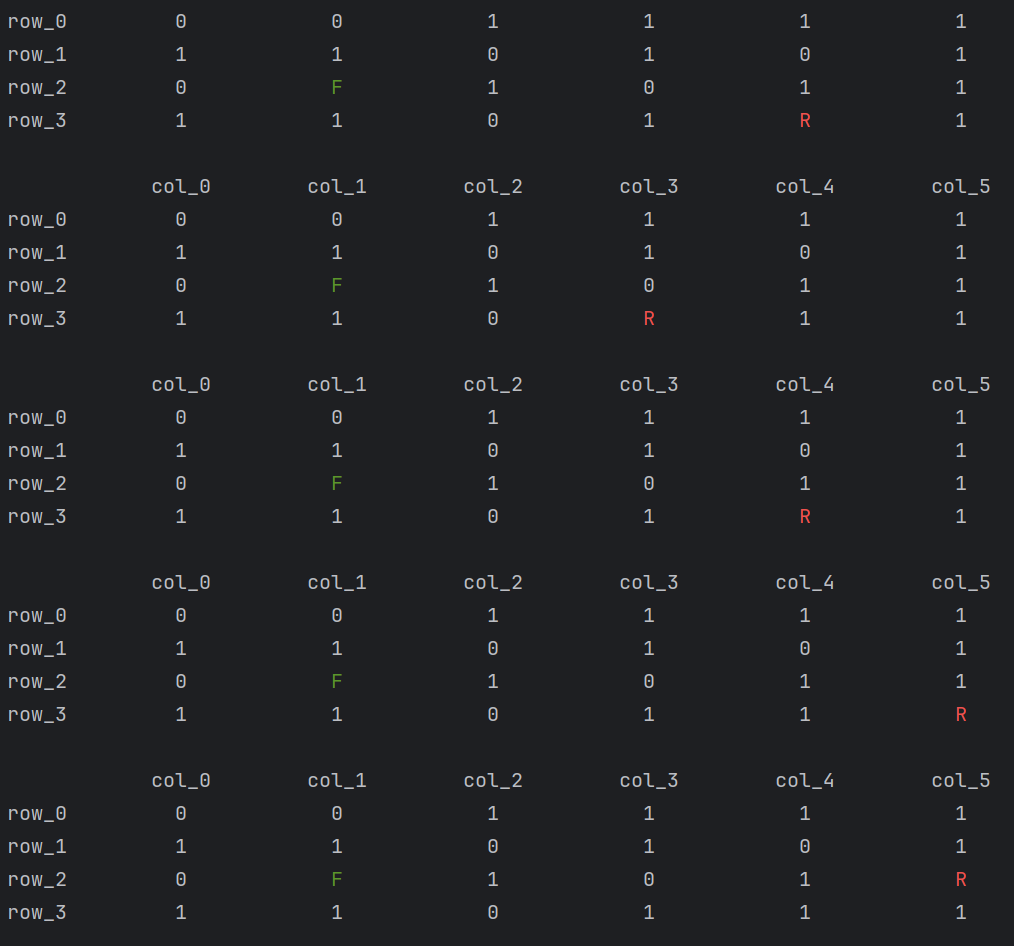


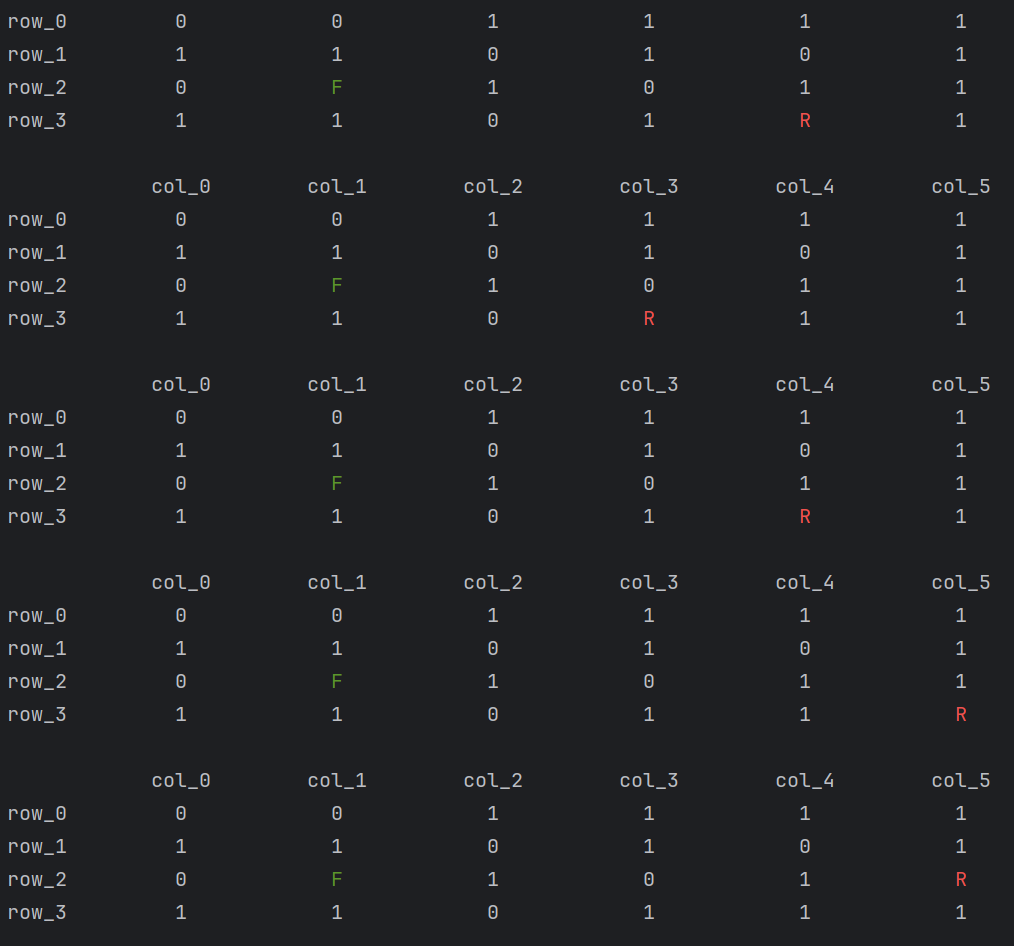


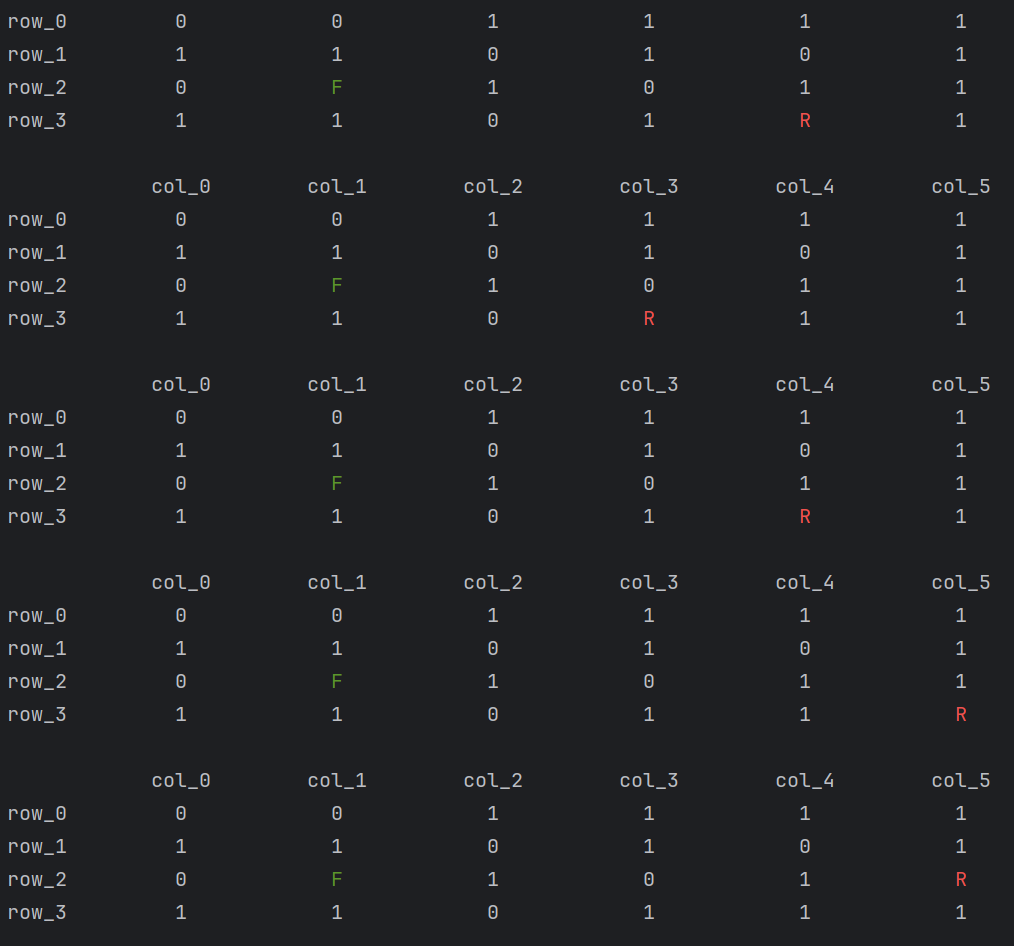
ตารางเส้นทางที่ rat finding food 1

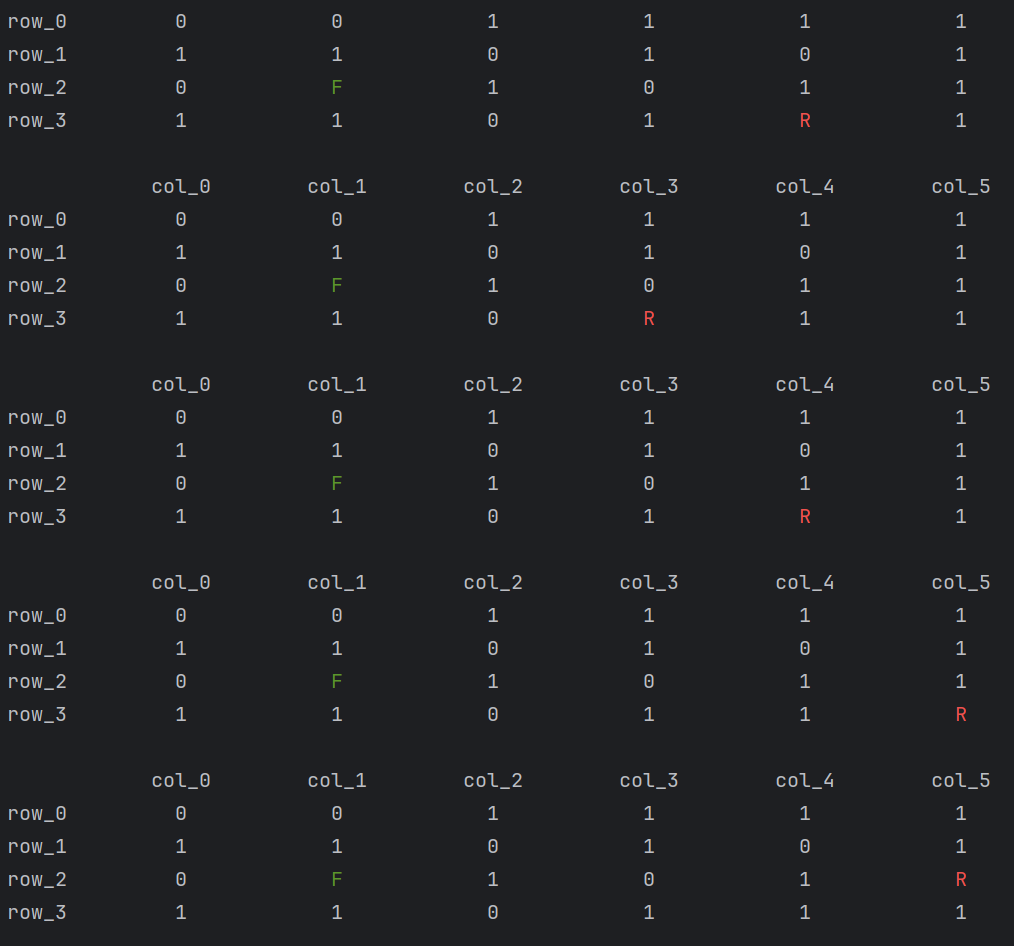
| Recursive Layer | Stack | Push/Pop |
| --- | --- | --- |
| 1 | down | Push down |
| 2 | down down | Push down |
| 3 | down down left | Push left |
| 4 | down down left down | Push down |

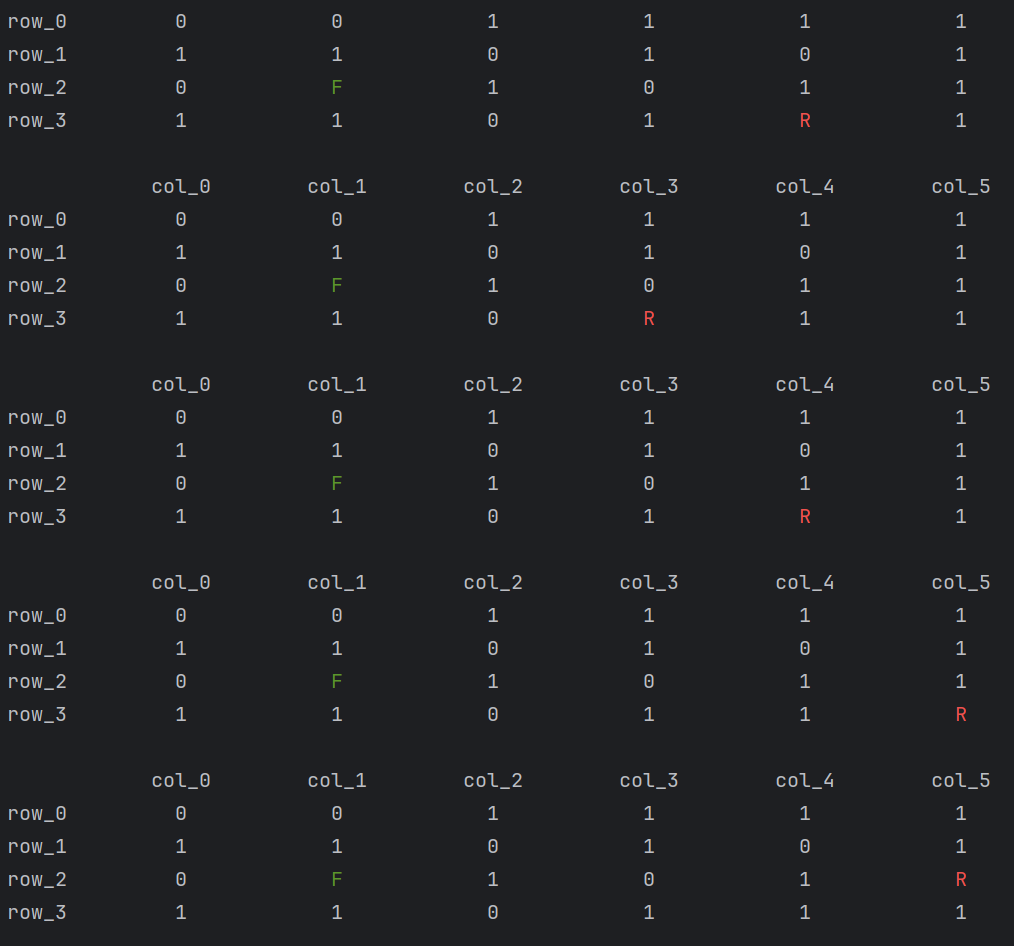
เส้นทางที่ rat finding food 2 เดิน(แบบแสดงทุกทาง)

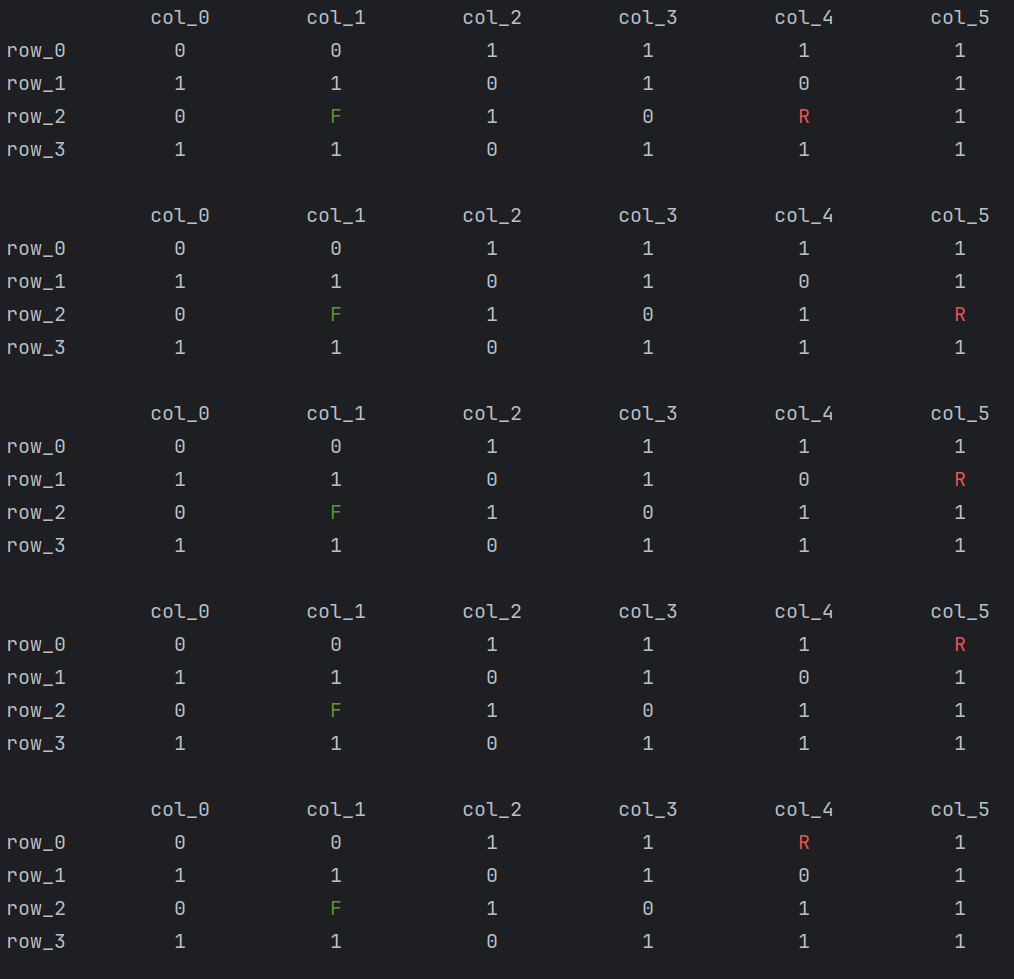


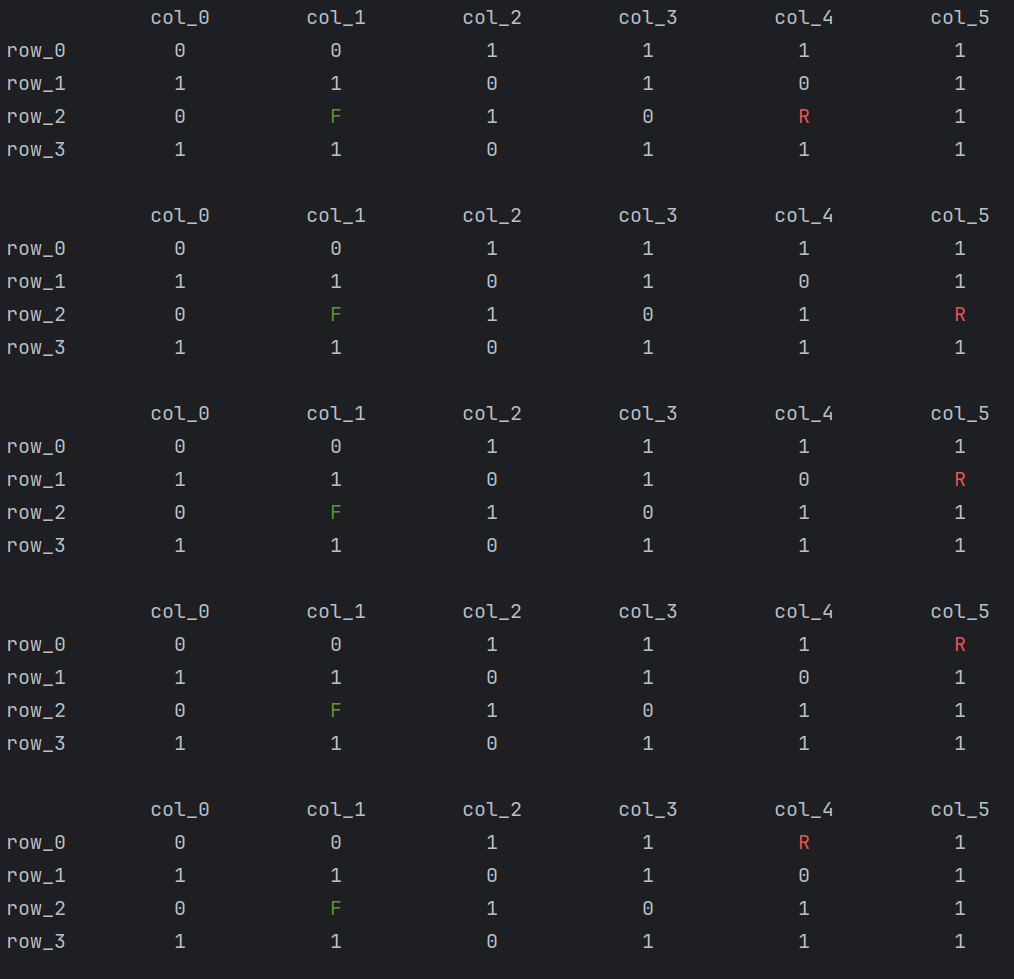


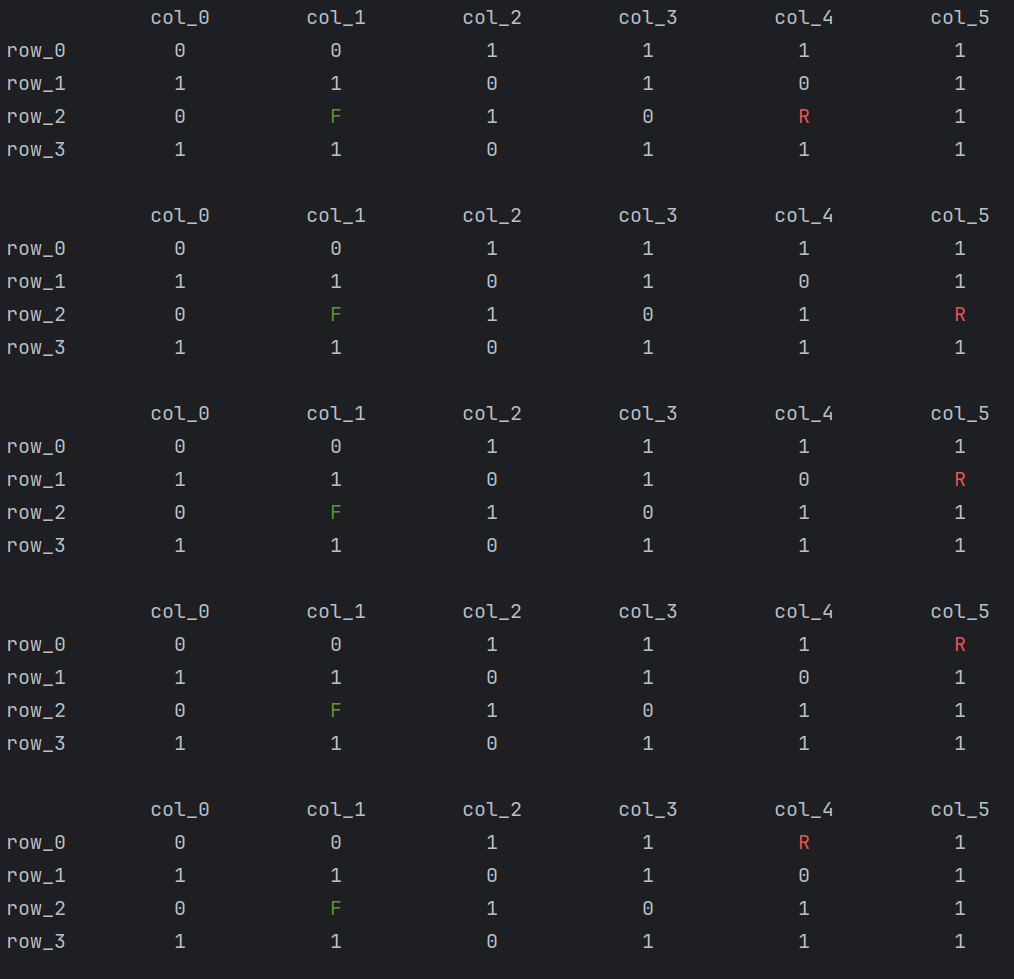


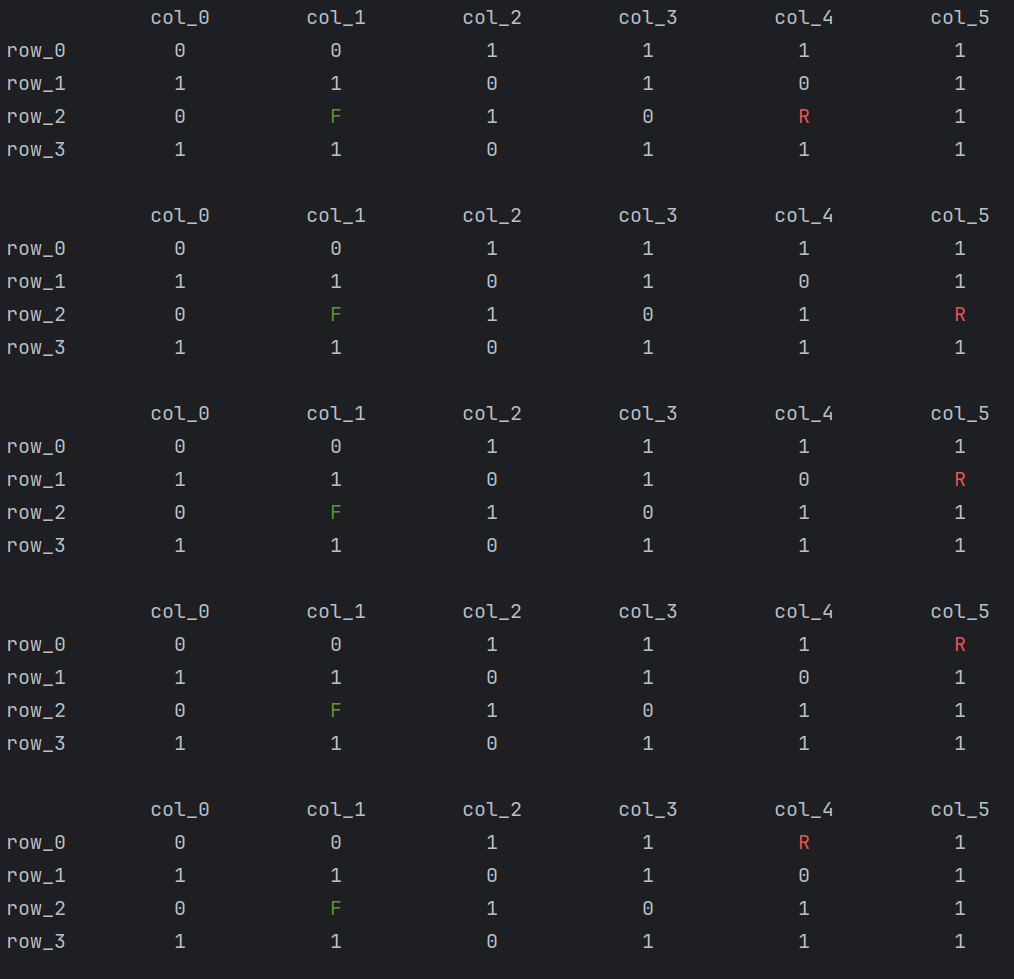


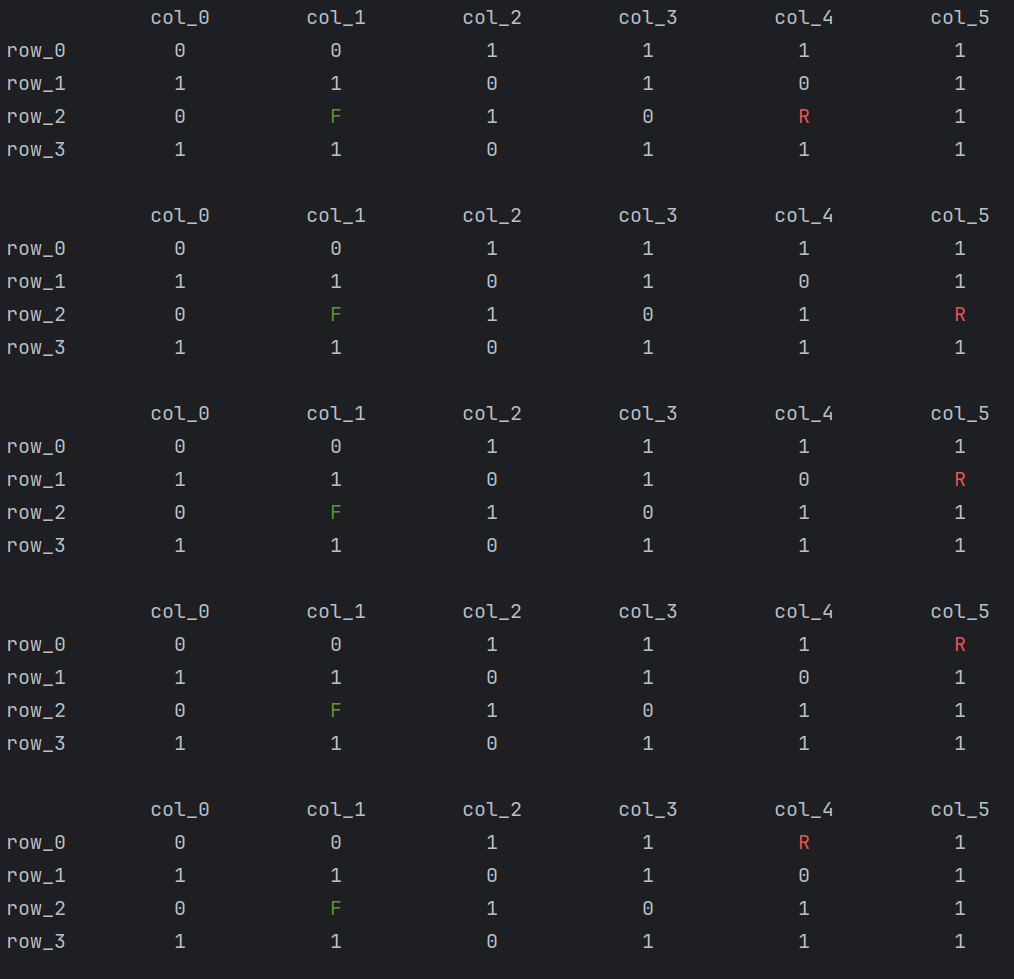


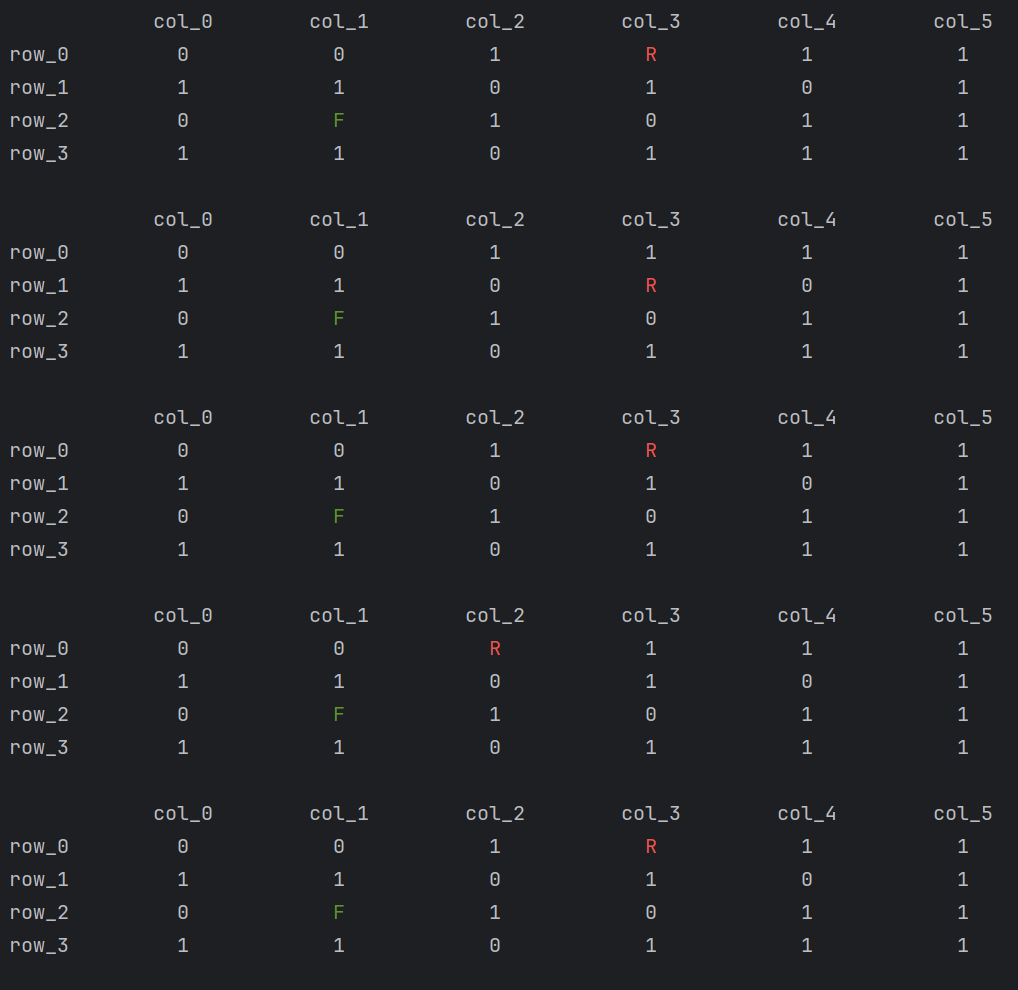


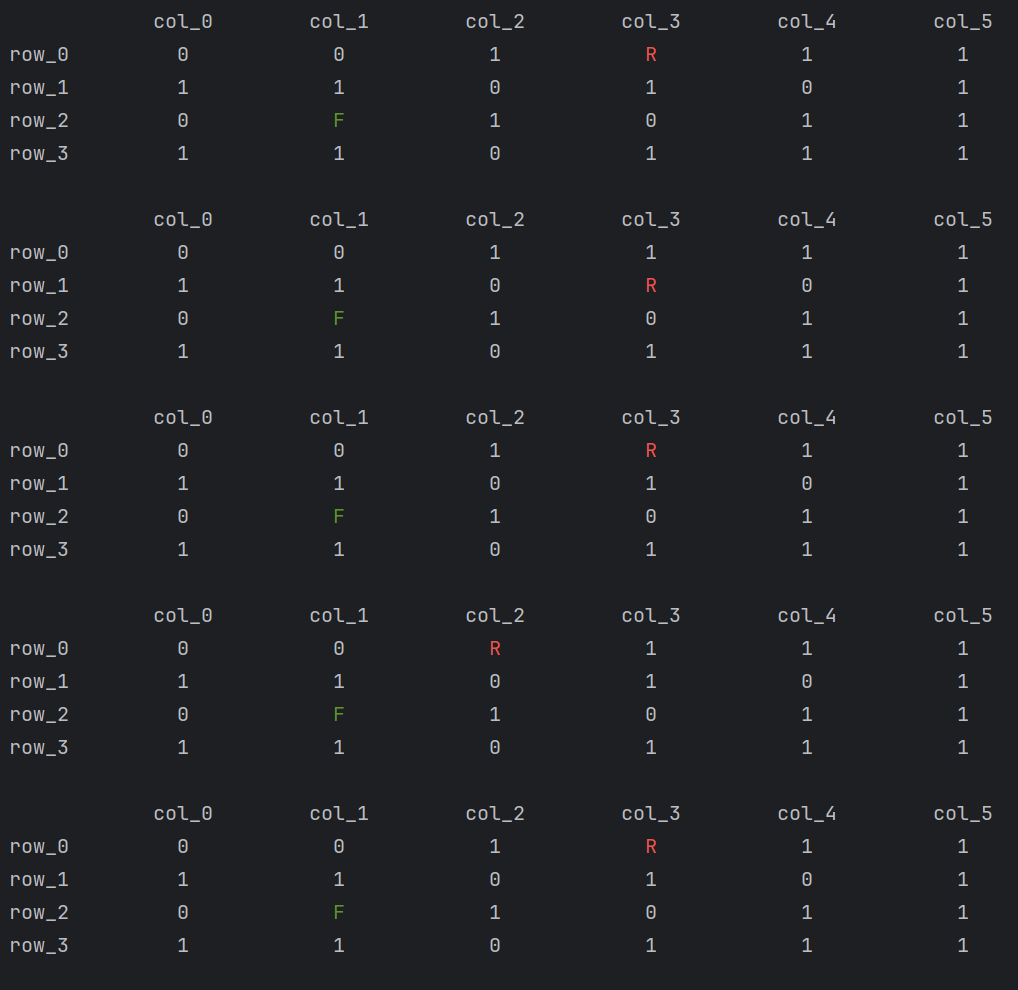


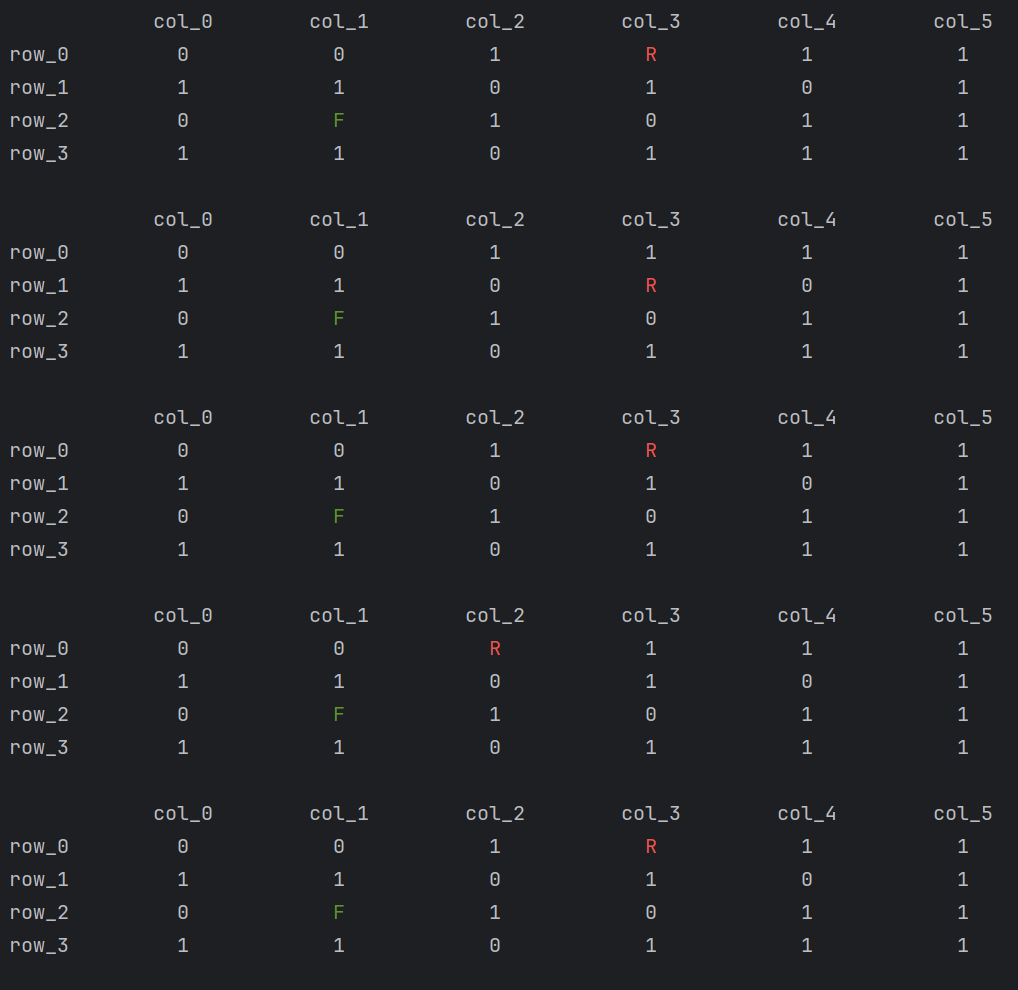


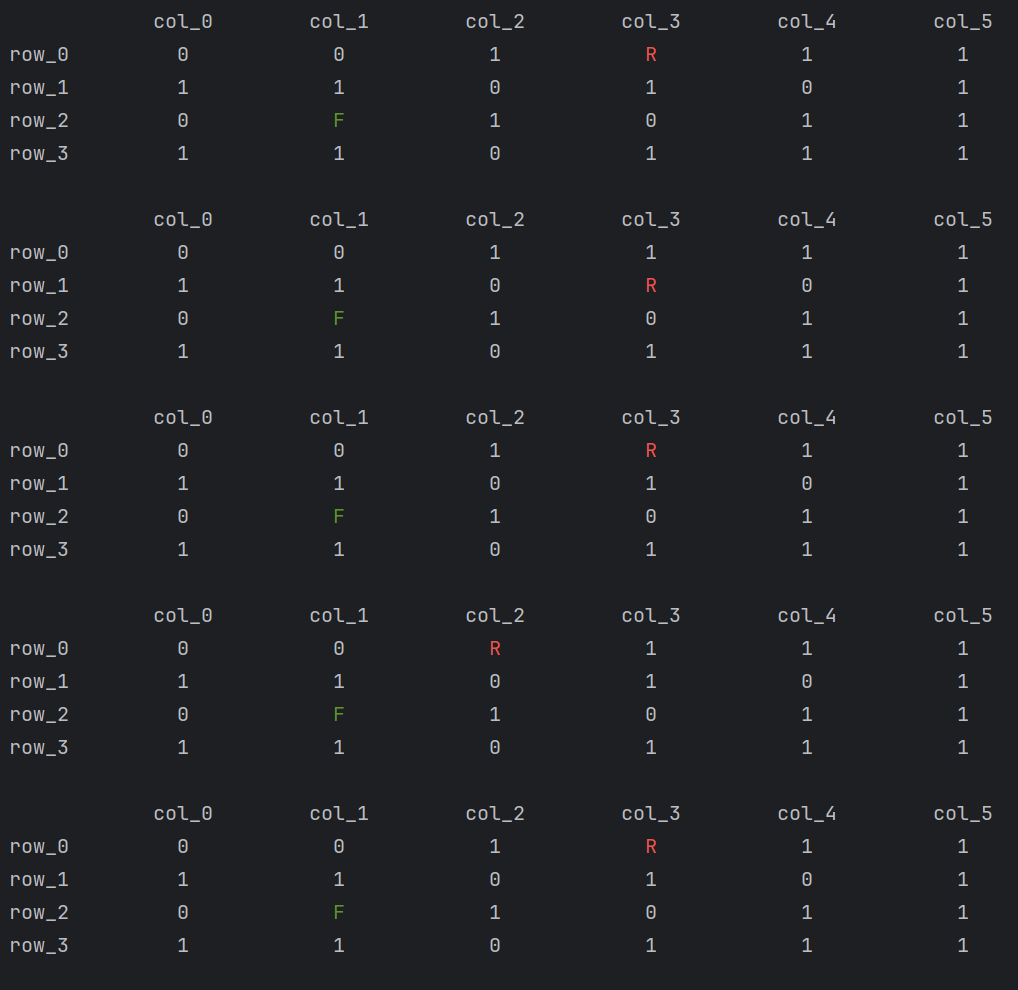


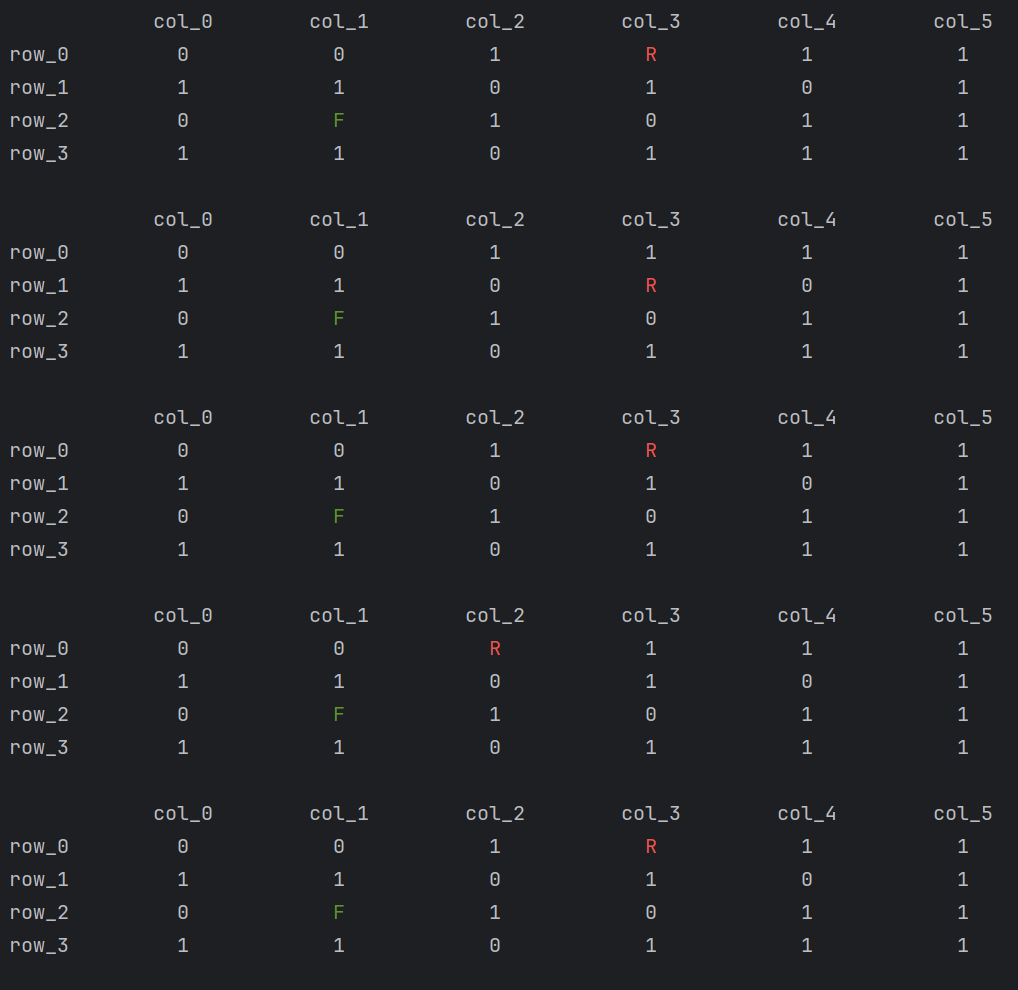


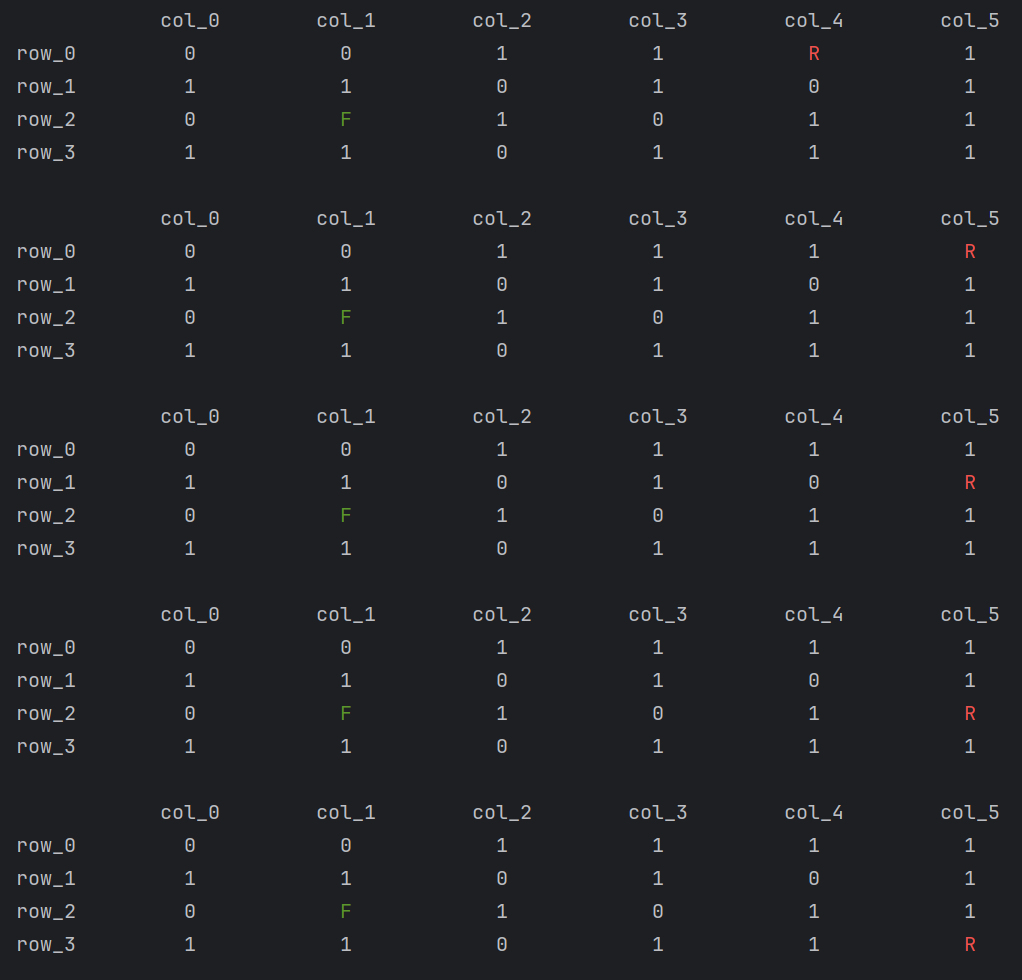


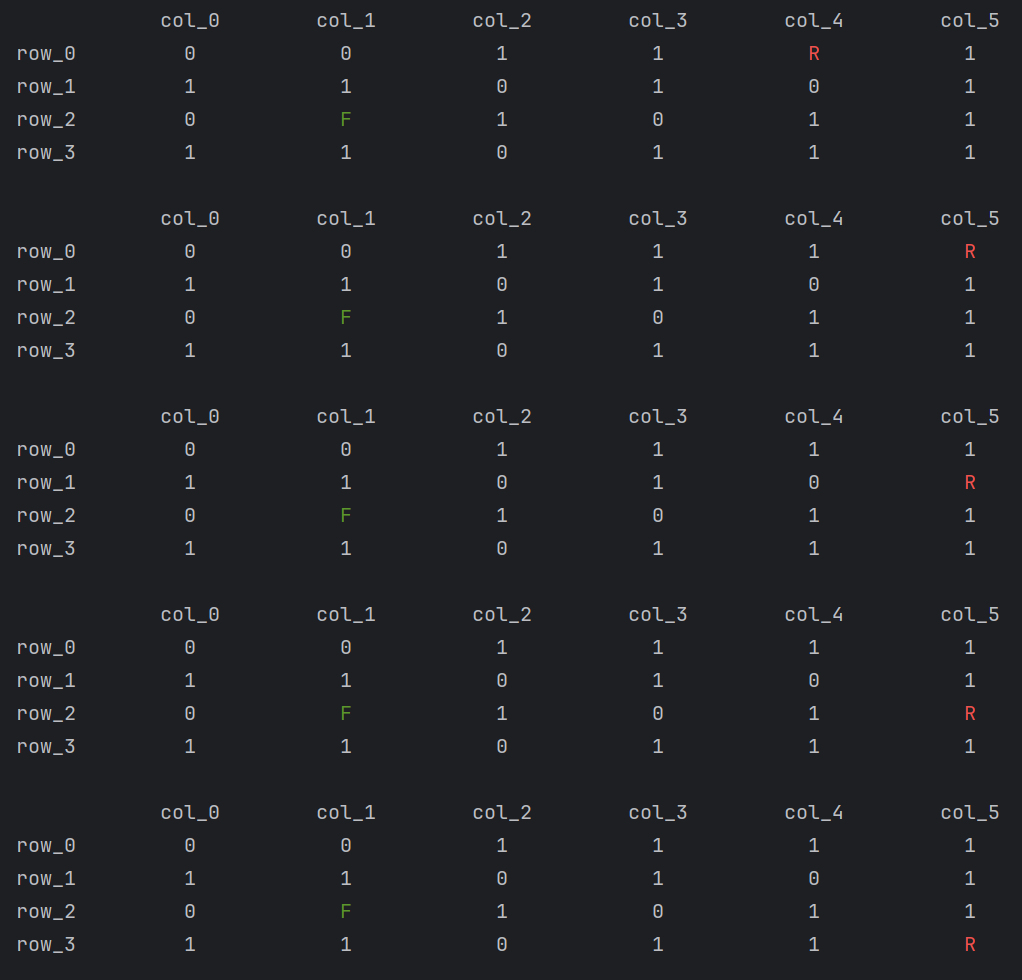


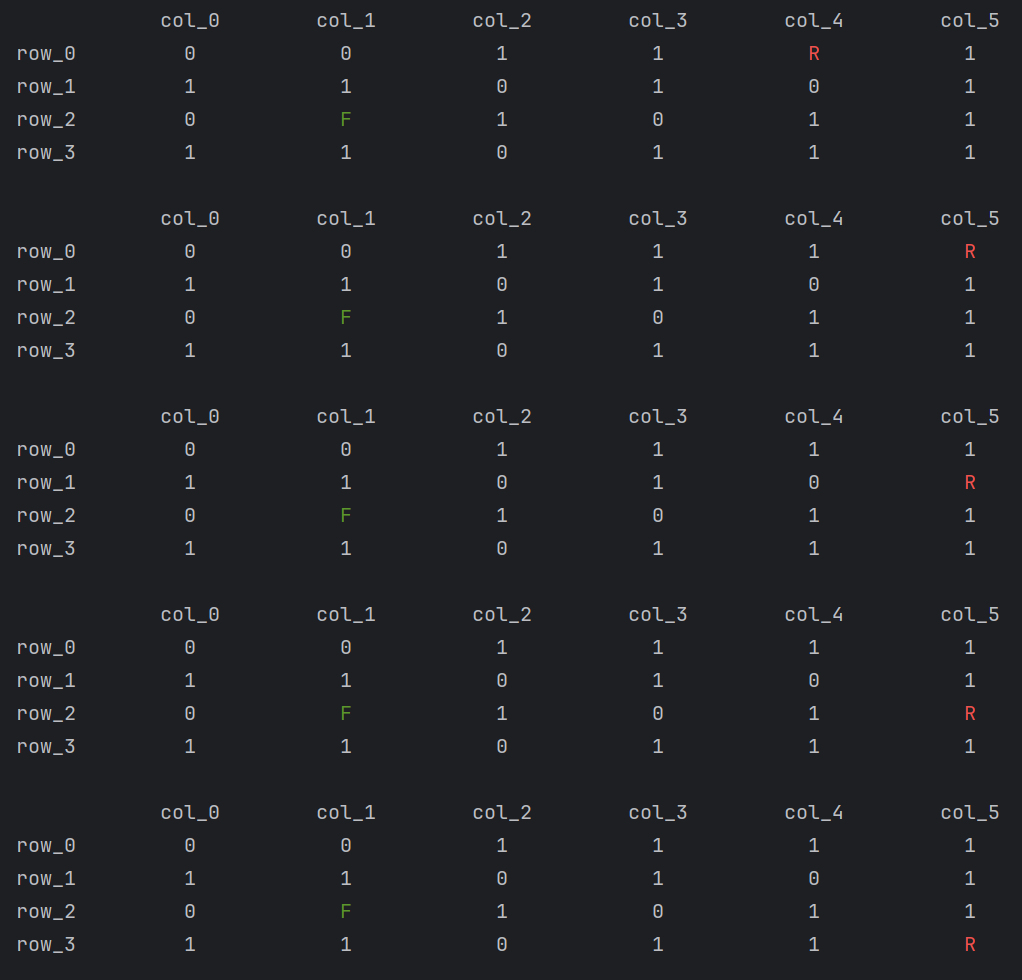


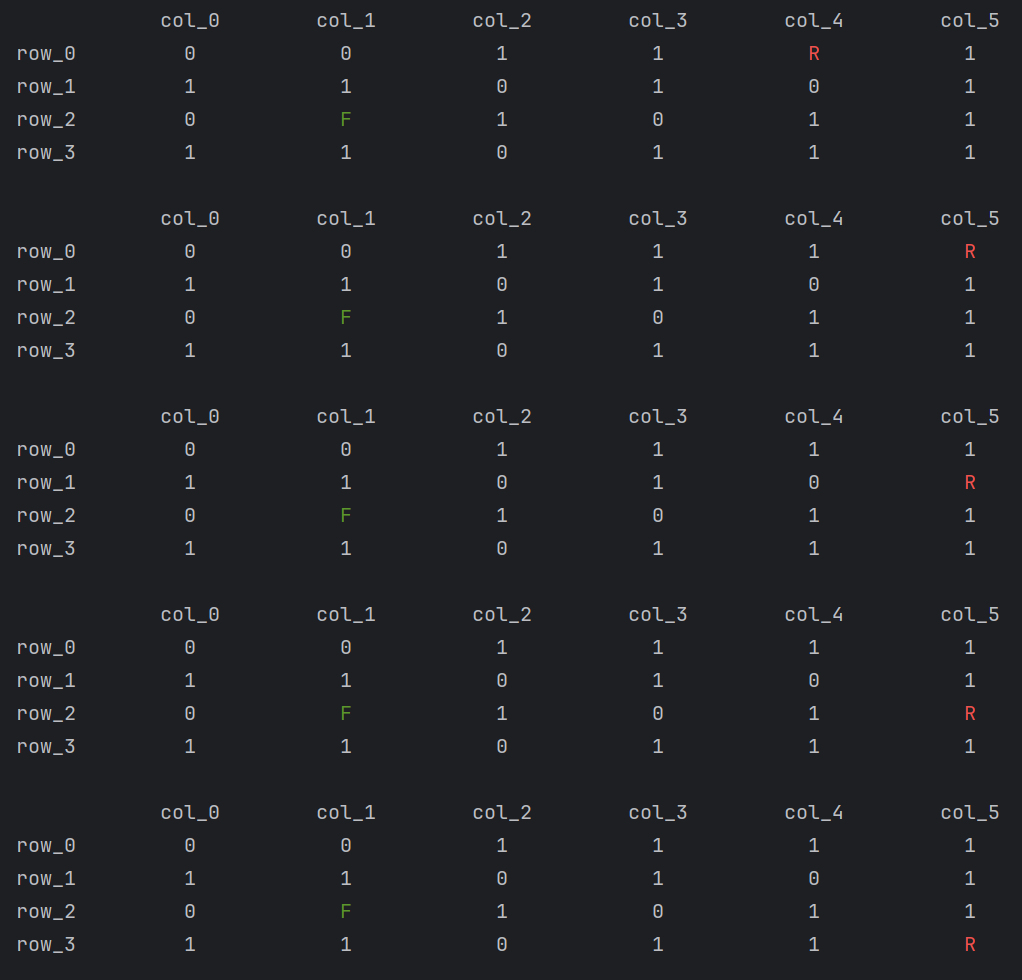


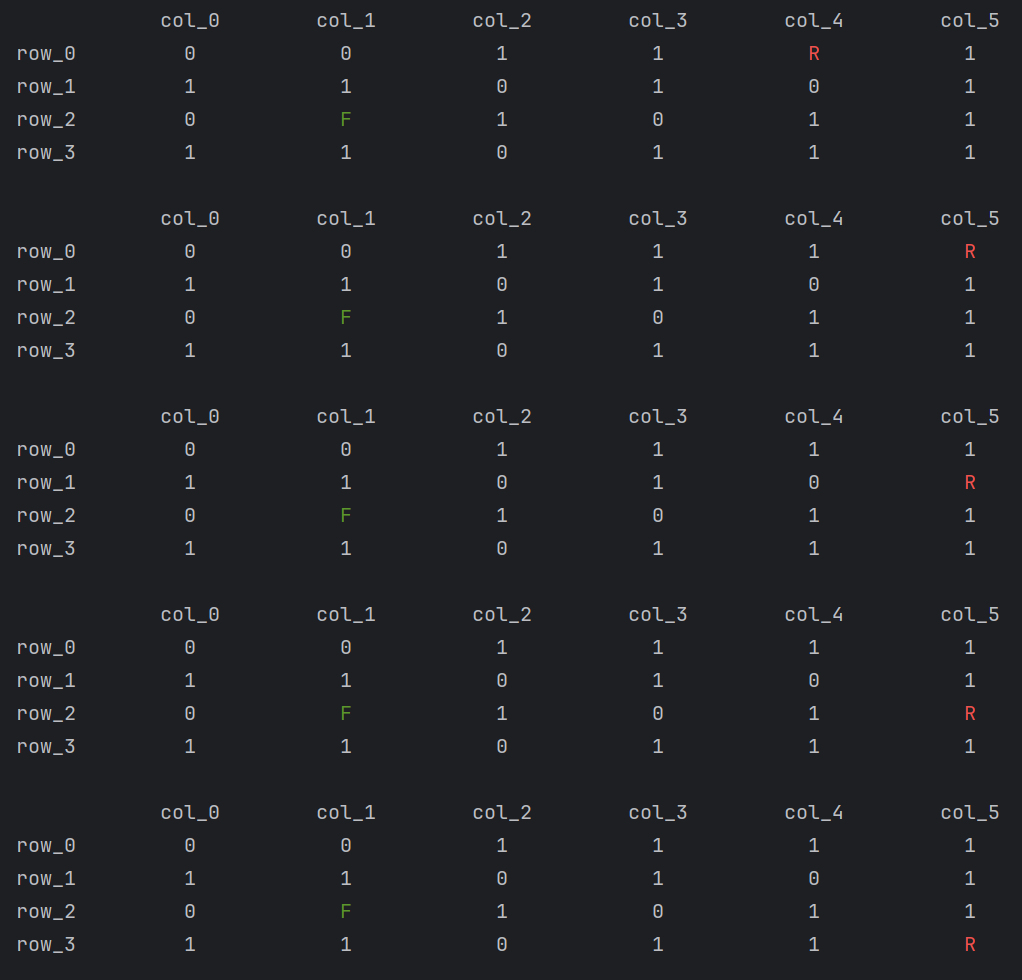


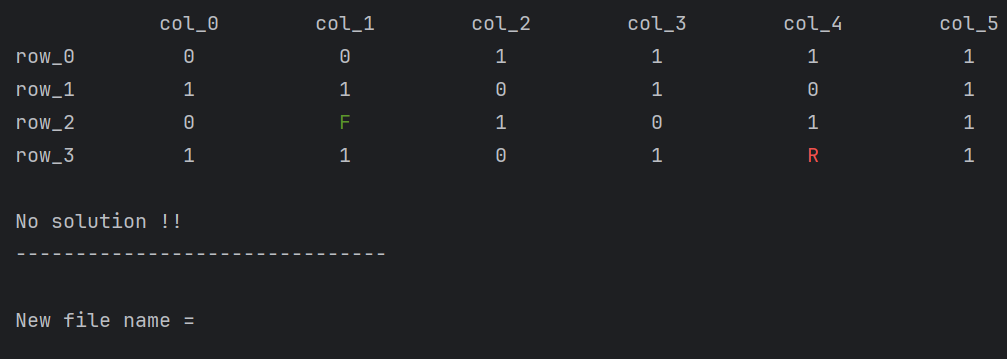










ไม่พบเส้นทาง โปรแกรมจะแสดง No solution !!

ตารางเส้นทางที่ rat finding food 2

| Recursive Layer | Stack | Push/Pop |
| --- | --- | --- |
| 1 | left | Push left |
| 0 |  | pop |
| 1 | right | Push right |
| 2 | right up | Push up |
| 3 | right up left | Push left |
| 2 | right up | pop |
| 3 | right up up | Push up |
| 4 | right up up up | Push up |
| 5 | right up up up left | Push left |
| 6 | right up up left left | Push left |
| 7 | right up up up left left down | Push down |
| 6 | right up up up left left | pop |
| 7 | right up up up left left left | Push left |
| 6 | right up up up left left | pop |
| 5 | right up up up left | pop |
| 4 | right up up up | pop |
| 3 | right up up | pop |
| 2 | right up | pop |
| 1 | right | pop |
| 0 |  | pop |

**Runtime ของโปรแกรม**

**limitation**

โปรแกรมไม่ได้รองรับการป้อนข้อมูลผู้ใช้ที่ผิดรูปแบบหรือไม่ถูกต้องได้ ซึ่งอาจทำให้เกิดข้อผิดพลาดหรือหยุดทำงานได้ในบางกรณี

โปรแกรมไม่สามารถกดย้อนกลับได้ในกรณีที่ผู้ใช้เลือกเส้นทางของหนูไปแล้ว หากผู้ใช้ต้องการกลับไปยังเส้นทางเดิมจะต้องทำการใส่ค่าเอง เช่น หากใส่ u (up) แล้วต้องการย้อนกลับมายังเส้นทางเดิม จะต้องใส่ d (down) เป็นต้น

**References**

<https://github.com/HydroXCarbon/Maze-Solver>

<https://www.boot.dev/assignments/0deb238d-8b3c-48b0-a367-caf08d65eed4>

<https://logicmojo.com/rat-in-a-maze>

Class Command ในโปรแกรมทำหน้าที่ควบคุมการเคลื่อนที่ของหนูในแผนที่และค้นหาเส้นทางที่นำหนูไปยังที่มีอาหาร นอกจากนี้ คลาส Command ใช้ตัวแปรเก็บข้อมูลแผนที่, เส้นทาง, จำนวนอาหาร, และสถานะการค้นหาทำให้classนี้สามารถควบคุมการเคลื่อนที่ของหนูและค้นหาอาหารในแผนที่ได้