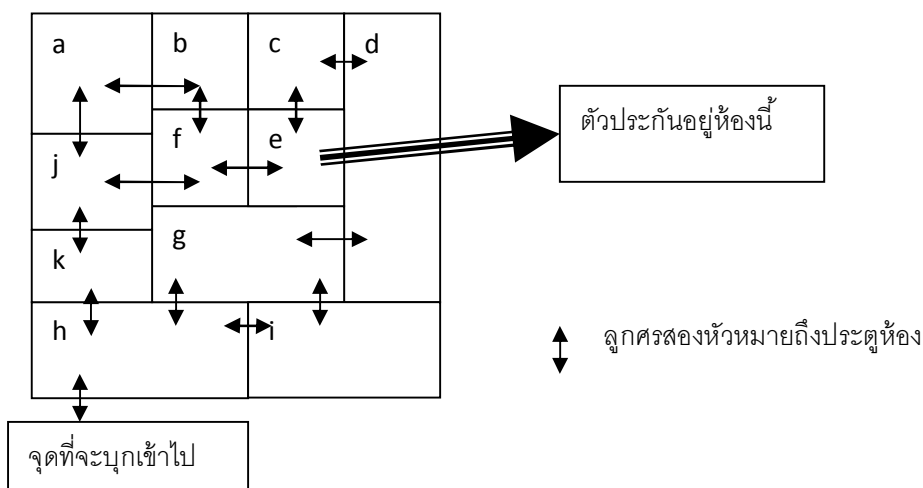


ตัวอย่างโจทย์ **prolog** สำหรับปัญหา **Artificial intelligence** ที่ใช้กับเกม หรือ **puzzle** ง่ายๆ ตัวพิมพ์ดีดจะเป็นตัวโจทย์ ส่วนตัวพิมพ์สีน้ำเงินจะเป็นคำตอบ

ตัวอย่างที่ 1

มีผู้ก่อการร้ายบุกยึดศูนย์เก็บอาวุธนิวเคลียร์ที่เล็กใช้แล้วของสหประชาชาติ แล้วจับตัวผู้บังคับการศูนย์นี้เป็นตัวประกัน ทางหน่วยปฏิบัติการพิเศษจึงส่ง **Solid Snake** บุรุษผู้ได้ชื่อว่าไม่มีวันตาย ลอบเข้าสู่ศูนย์ โดยมีจุดมุ่งหมายคือ เข้าช่วยเหลือตัวประกันให้ได้โดยหลีกเลี่ยงการถูกศัตรูพบให้มากที่สุด ตึกของศูนย์ที่กลุ่มคนร้ายยึดไว้มีลักษณะเป็นโกดังชั้นเดียวตัวรูป



Solid Snake นี้ไม่มีวันตายก็จริง แต่เขามีนิสัยเสียอย่างร้ายกาจคือ ถ้าไม่มีการยืนยันด้วยโปรแกรมคอมพิวเตอร์ถึงเส้นทางการบุกละก็ เขาจะไม่ยอมทำงานเด็ดขาด ดังนั้น **Otacon** ซึ่งเป็นโปรแกรมเมอร์ของหน่วยปฏิบัติการพิเศษ จะต้องเขียน **Prolog** เพื่อหาเส้นทางที่ดีที่สุด(ผ่านห้องต่างๆน้อยที่สุด) ให้ได้

แต่ **Otacon** เกิดต้องเสียอย่างกระหน่ำหนักสาเหตุเนื่องจากออกข้อสอบยากเกินไป ทำให้นักเรียนแข่งให้ต้องเสีย) เขาจึงให้คุณเป็นคนเขียนแทน โดย **Otacon** วางโครงโปรแกรมไว้ดังนี้

Fact ต่างๆ จะเขียนได้ดังนี้

way(X,Y) หมายถึงมีทางเดินจากห้อง X ไปห้อง Y

ส่วนการคำนวณเส้นทางนั้น จะใช้คำนวณโดยเรียกใช้

route(X, Y, [H|T], Avoid, NumberOfDoor).

โดยให้ X คือห้องเริ่มต้น ถือเป็น input

Y คือห้องที่ตัวประกันอยู่ ถือเป็น input

[H|T] จะเป็นคำตอบของเรา นั่นก็คือ ลิสต์ของห้องที่อยู่ระหว่างทางของ X กับ Y (ไม่รวม X กับ Y)

Avoid คือลิสต์ของห้องที่เราไม่ต้องการเข้าซ้ำ ถือเป็น input

NumberOfDoor เป็นคำตอบอย่างที่สองของ predicate ที่ชื่อว่า route นี้ โดยเป็นตัวเลขแสดง

จำนวนประตูห้องที่ต้องเดินผ่านโดยใช้เส้นทาง [H|T]

a) จงเขียน fact ทั้งหมด(3 คะแนน)

way(a,b).

way(a,j).

way(b,a).

way(b,f).

way(c,d).

way(c,e).

way(d,c).

way(d,g).

way(e,f).

way(e,c).

way(f,b).

way(f,e).

way(f,j).

way(g,d).

way(g,i).

way(g,h).

way(h,k).

way(h,g).

way(h,i).

way(i,g).

way(i,h).

way(j,a).

way(j,f).

way(j,k).

way(k,j).

way(k,h).

b) จงเขียน predicate route (3 คะแนน)

member(X,[X|T]).

member(X,[H|T]):- not(X=H), member(X,T).

route(X,Y,[],Avoid,1):- way(X,Y).

route(X,Y,[H|T],Avoid,N):- way(X,H), not(member(H,Avoid)), route(H,Y,T,[H|Avoid],N2), N is N2+1.

- c) จงแสดงการเรียกใช้ `route` เพื่อหาทางจากห้อง h ไปห้อง e รวมทั้งเขียนคำตอบที่ได้จากProlog ด้วย (2 คะแนน)

`route(h,e,Route,[h,e],N).`

`Route = [k, j, a, b, f]`

`N = 6 ;`

`Route = [k, j, f]`

`N = 4 ;`

`Route = [g, d, c]`

`N = 4 ;`

`Route = [i, g, d, c]`

`N = 5 ;`

`No`

ตัวอย่างที่ 2

ต่อไปนี้จะให้อ่านเรื่องที่แต่งขึ้น แล้วแก้ปัญหาโดยใช้ prolog

จอมยุทธ์ฝ่ายธรรมะสามคน คือ ก๊วยเจ๋งผู้เปี่ยมด้วยคุณธรรม อี้ยังผู้ปราดเปรื่อง และ อี้ซิงคิงผู้เป็นประมุขพรรคกกำลังจะโดยสารเรือน้อยลำหนึ่งข้ามฝั่งแม่น้ำฮวงโหเพื่อไปงานประลองยุทธ์ที่เขาหัวซาน ขณะนั้นเอง ได้ปรากฏเงาร่างคนสามคนโฉบทะยานถึงฝั่งเช่นเดียวกัน ฝีเท้าของทั้งสามคนที่มาใหม่นั้น รวดเร็วยิ่งนัก วิชาตัวเบา นับว่าไม่ต่ำทราชม พวดคืออวเอียงซงฉายาพิษประจิมผู้โหดเหี้ยมอำมหิต ปีกที่เอื้อมปีกสาจแดงปีกที่เอื้อมปีกที่เอื้อมปีกนี้ นมีชื่อเสียงเรื่องความยุติธรรม ชื่อตรงในการประลอง อัสวินแห่งเมืองซีฝียังเคยกล่าวคำยกย่อง) คนสุดท้ายคือ จอมยุทธ์นิรนามฉายาปีกสาจไข แต่มีฉายาว่าปีกสาจ แต่รอบรู้วิชาฟุ่กันแตกฉานยากมีใครเทียม ผากผลงานให้คนรุ่นหลังมากมาย แม้อัจฉริยะนอกด้านอย่างงจากแต่ยังต้องการคบหาเป็นสหาย

นายท้ายเรือลำเล็กนั้น ตื่นตระหนกกลัวเหล่าจอมยุทธ์ทั้งกระโจนลงน้ำ หนีไป สรุพบว่าตอนนี้ มีจอมยุทธ์ฝ่ายธรรมะแนวๆ 3 คน และฝ่ายที่ไม่เย็นยันตนอีก 3 คน กับเรือข้ามแม่น้ำ อีก 1 ลำ ซึ่งเรือลำนี้ จุคนได้ที่ละ 2 คนเท่านั้น

ทั้งหกต่างต้องการข้ามฝั่ง แต่อี้ยังนั้น เกรงว่า หากบนฝั่งไหนฝั่งหนึ่งมีจอมยุทธ์ฝ่ายธรรมะมากกว่าฝ่ายธรรมะแล้วละก็ ฝ่ายธรรมะจะรวมทำร้ายฝ่ายธรรมะอย่างแน่นอน ดังนั้นจึงต้องหาทางให้ทุกคนข้ามฝั่งได้หมดโดยไม่ให้มีฝ่ายธรรมะมากกว่าฝ่ายธรรมะที่ฝั่งใดในเวลาหนึ่งเลย (อี้ยังคิดว่า แม้ปีกที่เอื้อมปีกยุติธรรมในการประลอง แต่ไม่ใช่การประลองจึงมีโอกาสไว้ใจได้ ปีกสาจไขนั้น ไปมาใช้ของรอย ผากฝีมือฟุ่กันตามที่ต่างๆ แม้ดูเป็นจอมยุทธ์ผู้ สุนทรียะ แต่เต็มไปด้วยปริศนา จึงมีโอกาสไว้ใจได้ว่าเป็นฝ่ายธรรมะเช่นกัน)

โชคดีที่อี้ยังนั้นฉลาดนัก ในสมองเหมือน prolog interpreter อยู่ ทั้งยังจดจำปัญหา solid snake ที่อี้ยังเอื้อมปีกผู้พ่อสอนได้เป็นอย่างดี จึงคิดใช้หลักวิธีเดียวกับการแก้ปัญหา solid snake แก้ปัญหานี้ โดย กำหนดให้ state หนึ่งของการข้ามแม่น้ำ เป็น

state(Tamma, Atam, Side)

โดย Tamma คือจำนวนจอมยุทธ์ฝ่ายธรรมะที่อยู่ฝั่งซ้ายของแม่น้ำ (ฝั่งที่ทั้งหกคนอยู่ในตอนแรก)

Atam คือจำนวนจอมยุทธ์ฝ่ายธรรมะที่อยู่ฝั่งซ้ายของแม่น้ำ

Side คือฝั่งของแม่น้ำ ที่เรืออยู่(มีฝั่งซ้ายกับขวาเท่านั้น)

คำตอบของ prolog จะอยู่ในรูปของลิสต์ของการลำเลียงคน ดังตัวอย่างนี้

[move(1, 1, right), move(2, 0, left)]

ลิสต์นี้ หมายถึง ฝ่ายธรรมชาติหนึ่งคนและฝ่ายธรรมชาติหนึ่งคนนั่งเรือไปฝั่งขวา จากนั้น ฝ่ายธรรมชาติสองคนนั่งเรือมายังฝั่งซ้าย

นอกจากนี้ ต้องเก็บลิสต์ของstate ที่เคยเกิดขึ้นมาแล้วด้วย(เช่นเดียวกับลิสต์ของทางที่เคยเลือกในปัญหา solid snake)

เพื่อไม่ให้เจอ state ซ้ำ ๆ

1. จงเขียน state สุดท้าย (1 คะแนน)

```
state(0, 0, right).
```

ยังมีต่อนะ

ตัวโปรแกรมหลักที่ขี้ขี้เขียนขึ้นนั้นเป็นดังนี้

```
% solution(CurrentState, Visited, Path)
```

```
solution(ตรงนี้อาจค่าตอบขี้ขี้ใส่ลงไป, _, []).
solution(CurrentState, Visited, [Move | RestOfMoves]) :-
    newstate(CurrentState, NextState),
    not(member(NextState, Visited)),
    make_move(CurrentState, NextState, Move),
    solution(NextState, [NextState | Visited], RestOfMoves).
```

จะเห็นว่าโปรแกรมหลักนั้น ละม้ายคล้ายคลึงโปรแกรมที่แก้ปัญหасolid snake (นั่นก็คือเราสามารถแปลงปัญหาต่างๆ เป็นรูปแบบนี้ แล้วแก้ด้วยวิธีเดียวกันยังงั้น)

2. จงเขียนนิยามของ

```
make_move(CurrentState, NextState, Move)
```

ซึ่งรับ CurrentState กับ NextState เป็น input แล้ววิธีที่รับ Move ที่เกิดจากการเปลี่ยน state เป็นคำตอบ (2 คะแนน)

```
make_move(state(M1, C1, left), state(M2, C2, right), move(M, C, right)) :-
    M is M1 - M2,
    C is C1 - C2.
make_move(state(M1, C1, right), state(M2, C2, left), move(M, C, left)) :-
    M is M2 - M1,
    C is C2 - C1.
```

ให้ carry(T, A) หมายถึงเรือได้ขนฝ่ายธรรมชาติ T คน และฝ่ายธรรมชาติ A คน ข้ามแม่น้ำ ำในเที่ยวหนึ่ง

3. จงเขียนนิยามของ

```
newstate(CurrentState, NextState)
```

ซึ่งรับ CurrentState เป็น input แล้ววิธีที่รับ NextState เป็นคำตอบ (4 คะแนน)

```
legal(X, X).  
legal(3, X).  
legal(0, X).
```

```
newstate(state(M1, C1, left), state(M2, C2, right)) :-  
    carry(M, C),  
    M <= M1,  
    C <= C1,  
    M2 is M1 - M,  
    C2 is C1 - C,  
    legal(M2, C2).
```

```
newstate(state(M1, C1, right), state(M2, C2, left)) :-  
    carry(M, C),  
    M2 is M1 + M,  
    C2 is C1 + C,  
    M2 <= 3,  
    C2 <= 3,  
    legal(M2, C2).
```