

## وظيفة عملي مبادئ الذكاء الصناعي

**مجموعات الطلاب:** 3 طلاب على الأقل إلى 5 طلاب على الأكثر في كل مجموعة. ويمكن أن يكونوا من فئات مختلفة.

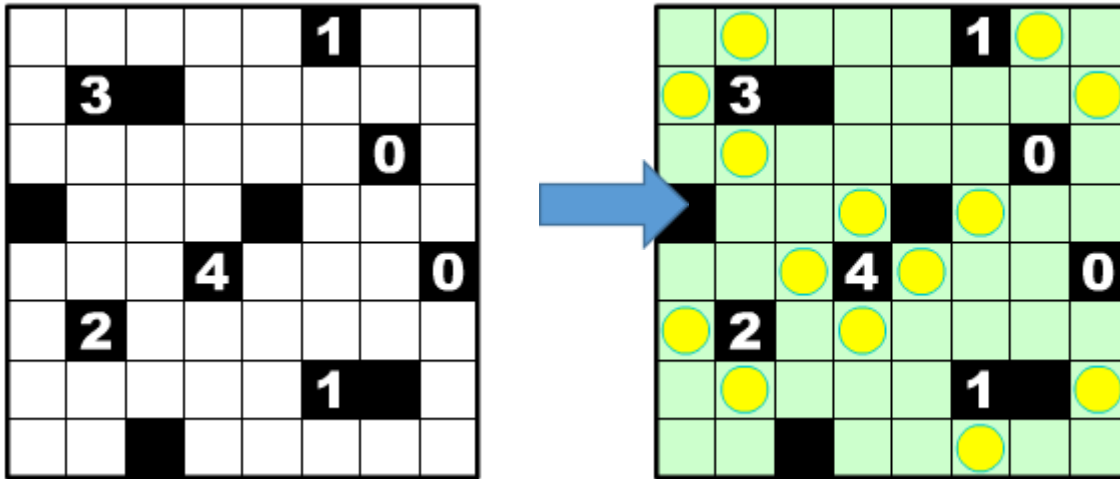
**موعد التسليم والمقابلات:** تسليم الوظيفة يوم الثلاثاء 14/6/2016. وتتم المقابلات في نفس اليوم ابتداءً من الساعة التاسعة صباحاً بعد التسجيل على الاستمارة على الرابط <https://forms.gle/dysX581ww6aCopES8> بحيث يسجل من كل مجموعة طالب واحد فقط. لا تمنح علامة للطلاب الذين لا يحضرون المقابلة.

**طريقة التقديم:** تقرير مطبوع يحتوي الشرح والاستنتاجات والبرامج، إضافة إلى مقابلة مخصصة للعرض والتقييم.

### Light- Up!!

لعبة light up والتي تسمى أحياناً akari من ألعاب الأحاجي التي يلعبها لاعب واحد والتي تعتمد على المنطق.

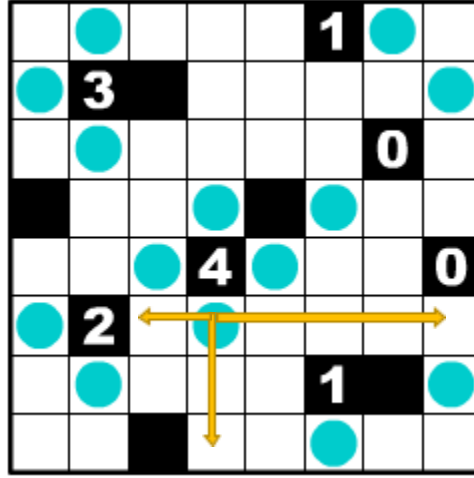
يمكنك معرفة قواعد اللعبة وتجربتها على هذا الرابط: <https://krazydad.com/play/akari>



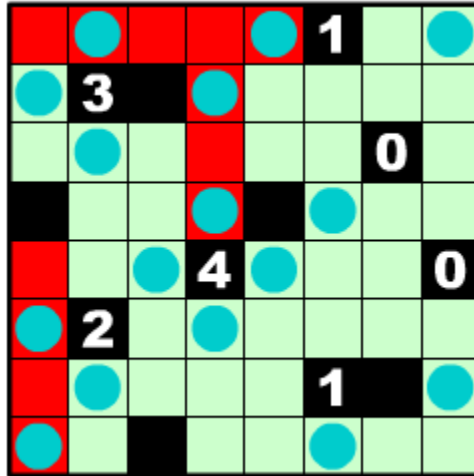
تتألف اللعبة من جدول مقسم إلى خلايا، يمكن أن يكون للجدول قياسات مختلفة. بعض هذه الخلايا مظلمة (نسبياً حائط) وبعض الخلايا المظلمة تحتوي على أرقام، هدف اللعبة هو وضع مصباح ضوئي (لمبة) في بعض خلاياها بهدف اضاءة جميع الخلايا غير المظلمة وفق القواعد التالية:

- العدد في الخلية يشير إلى عدد المصابيح التي يمكن أن توضع بشكل أفقي أو عمودي (وليس قطري) حول الخلية، يمكن أن توضع المصابيح في أي خلية غير مظلمة (حتى التي لا تحتوي على أرقام بجوارها).

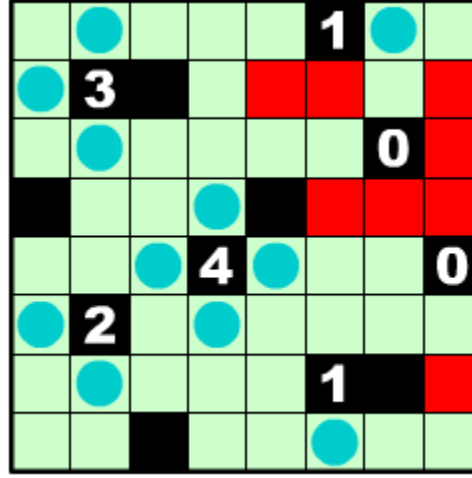
- كل مصباح يوضع في خلية بيضاء يضيء الخلايا البيضاء التي في سطره وعموده والمحدودة بحائط (خلية سوداء) أو حدود الرقعة).



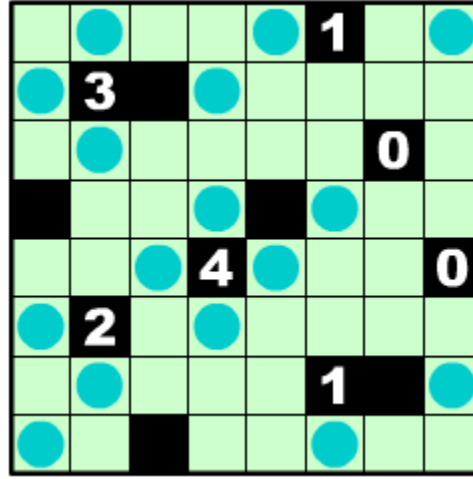
- لا يجب لمصباح أن ينير خلية تحتوي على مصباح آخر، (أي لا يمكن وضع مصباحين في نفس السطر أو العمود المحاطين بحائط أو بحدود الرقعة)



- في نهاية اللعبة كل مربع يجب أن يضاء



- تنتهي اللعبة عندما يتم إضاءة جميع المربعات البيضاء



أولاً: تحقيق المسألة بهدف التأكد من صحة حل (8 علامات)

تمثيل المسألة

سنفترض طريقة لتمثيل المعطيات، التزم في حلك بهذه البنية حصراً. تتيح هذه البنية إمكانية حل أي لعبة موصفة قابلة للحل (ليس فقط المثال السابق)

1- نمثل بعد الرقعة بالحقيقة التالية، التي تعبر عن العلاقة بين عدد الأسطر وعدد الأعمدة:

```
size(Rows, Columns) .
```

فمثلاً من أجل الرقعة السابقة يكون لدينا:

```
size(8,8) .
```

2- نمثل الخلايا المظلمة (الحائط) ضمن اللوح عن طريق اسناديات، بحيث تعبر عن علاقة بين معامليين: السطر والعمود، بحيث يكون لدينا حقيقة من أجل كل خلية مظلمة ضمن الجدول.

```
wall(Row, Column) .
```

من أجل المثال السابق يكون لدينا 12 حقيقة على الشكل التالي:

wall(1,6) . wall(2,2) . wall(2,3) . wall(3,7) .	wall(4,1) . wall(4,5) . wall(5,4) . wall(5,8) .	wall(6,2) . wall(7,6) . wall(7,7) . wall(8,3) .
--	--	--

3- نمثل خلايا الحائط التي تحتوي على أرقام بإسناديات تعبر عن علاقة بين ثلاثة معاملات، السطر والعمود والرقم، بحيث يكون لدينا حقيقة من أجل كل خلية حائط تحوي رقماً ضمن الجدول.

```
wall_num(Row, Column, Num) .
```

من أجل المثال السابق يكون لدينا 7 حقائق على الشكل التالي:

wall_num(1,6,1) . wall_num(2,2,3) . wall_num(3,7,0) . wall_num(5,4,4) .	wall_num(5,8,0) . wall_num(6,2,2) . wall_num(7,6,1) .
--	---

4- نمثل الخلايا الخالية التي تحتوي على مصباح (خلايا الحل) بإسناديات تمثل علاقة بين معامليين: السطر والعمود. هذه الخلايا يجب أن تكون مولدة آلياً بشكل ديناميكي خلال الحل، ولكن مبدئياً يمكنك إضافة حقائق ثابتة تمثل خلايا الحل في هذا الجزء من الوظيفة، ثم إزالتها من الكود في القسم الثاني من الوظيفة.

من أجل المثال السابق يكون لدينا مثلاً

```
light(1,1).  
light(1,5).  
light(1,7).  
etc.
```

### التحقق من كون رقعة ما تمثل حلاً صحيحاً لمسألة

للمساعدة سنستعرض بداية مجموعة من الإجراءات التي يمكن أن تستخدمها في حلك، يمكنك جمع هذه الإجراءات في إجراءات أكبر ولك الحرية في طريقة تحقيقها (أو عدمه في حال أحببت أن تحل بطريقة أخرى مع الالتزام بالبنية المقترحة).

- إجرائية تعطي سلسلة تمثل "جيران" خلية ما ذات سطر وعمود (تستخدم لإيجاد جوارات الخلايا المرقمة)
- إجرائية تعطي سلسلة تمثل مجموعة خلايا السطر المحيطة بخلية ما من اليمين واليسار والمحاطة بحدود الرقعة أو بحائط، ومثلها إجرائية تعطي سلسلة تمثل مجموعة خلايا الأعمدة المحيطة بخلية ما من الأعلى والأسفل والمحاطة بحدود الرقعة أو بحائط.
- إجرائية تقوم بعدد خلايا المصباح الموجودة ضمن سلسلة تحوي خلايا تكون محددة بسطرها وعمودها.
- إجرائية تتحقق من كون خلية ما مضاءة.
- إجرائية تتحقق من أن خلية حائط مرقمة محاطة بخلايا مصباح عددها يساوي الرقم في الخلية.

المطلوب منك في هذا الجزء من المسألة كتابة إجرائية solved التي تتحقق من أن المصابيح في مكانها الصحيح وذلك من خلال التأكد من قواعد اللعبة، يمكن تقسيمها إلى قواعد جزئية على النحو التالي:

1. إجرائية all\_cells\_lighted تتأكد أن جميع الخلايا في الرقعة مضاءة من قبل مصباح.
  2. إجرائية no\_double\_light تتأكد أن خلايا المصباح ليست مضاءة من قبل مصباح آخر<sup>1</sup>.
  3. إجرائية light\_count\_correct تتأكد من عدد خلايا المصباح المحيطة بجميع خلايا الحائط المرقمة.
- بعد كتابتك لهذه القواعد، ما عليك سوى كتابة قاعدة تقوم بتجميعها للتحقق من الشروط جميعاً مثلاً على الشكل التالي:

```
solved:- all_cells_lighted,  
         no_double_light ,  
         light_count_correct .
```

لاحظ أن هذه القاعدة والقواعد التي تحتويها لا تأخذ متحولات حيث أنها ليست بحاجة إليها.

لتحقيق هذه الإجراءات لا بد من المرور على الخلايا المعنية بكل إجرائية واحدة تلو الأخرى، وذلك بالمرور عليها ضمن قائمة. فيمكننا مثلاً لتحقيق الشرط الأول المرور على جميع خلايا الرقعة من خلال قائمة تحوي جميع الثنائيات الممكنة للسطر والعمود.

<sup>1</sup> يمكنك حل هذه الإجرائية بطريقتين الأولى هي عدد خلايا المصباح الموجودة ضمن خلايا السطر والعمود المحيطة بها من اليمين واليسار والمحاطة بحدود الرقعة أو بحائط والتأكد أن هذا العدد لا يتجاوز الواحد. أو يمكنك حلها بطريقة نقض الفرض، أي أن تكتب قاعدة توجد مصباحين ضمن سطر أو عمود لا يوجد بينها حائط، ثم تقوم بنفي هذه القاعدة.

توليد هذه القائمة هو عملية بسيطة جداً، وذلك بتجميع حلول إسنادية ما<sup>2</sup> (لنسميها مثلاً `get_a_cell(Row,Col)`) تعطي مجموعة حلول تمثل كل الثنائيات الممكنة للأسطر والأعمدة.

وبنفس الطريقة يمكن تجميع خلايا الحائط المرقمة (لتحقيق الشرط الثاني) وخلايا المصباح (لتحقيق الشرط الثالث).

### ثانياً: حل المسألة باستخدام القواعد (4 علامات)

بدايةً قم بتعليق جميع الحقائق التي تمثل الحل (`light(Row, Col)`) حيث أن مهمة البرنامج هو إيجاد هذه الحقائق أثناء الحل بشكل ديناميكي.

يتبع لاعبو Akari قواعد تمثل استراتيجيات للحل تأتي من خبرتهم بلعب هذه اللعبة، ويقومون بملء الرقعة تدريجياً بتكرار الحل بهذه القواعد حتى إيجاد الحل كاملاً (في حال كانت هذه القواعد كافية لحل المسألة). كلما كانت القواعد أكثر تعقيداً كلما كان برنامجك قادراً على حل مسائل أصعب.

قد تحتاج في هذه الحالة إلى تعريف حقائق ديناميكية أخرى (مثل خلية مستحيل أن يوجد فيها مصباح، خلية مضاعة ...)، يمكن أن تعتمد في بعض الإجراءات مبدأ إقصاء الحلول الغير ممكنة (تمثل بحقائق ديناميكية أخرى) حتى لا يبقى لك إلا الحل الممكن.

الرابط التالي فيديو يشرح مجموعة استراتيجيات، حقق ما تستطيع منها:

<https://www.youtube.com/watch?v=ZHeEEGFIGY>

لاحظ أنه هناك نوعين من القواعد، منها قواعد تطبق مرة واحدة فقط ومنها قواعد تطبق أكثر من مرة حتى الوصول إلى حل.

ستحتاج مجموعة قواعد مساعدة منها:

1. تهيئة اللعبة: والذي يتضمن حذف كل الإسناديات الديناميكية الموجودة مسبقاً (من تشغيل سابق للعبة)<sup>3</sup>.
2. طباعة الرقعة: (ننصحك بكتابه هذه القاعدة بدايةً لتتأكد من شكل الرقعة عنك) يجب أن تتم الطباعة بعد كل خطوة لعب، بحيث تطبع الرقعة سطراً سطراً<sup>4</sup>، نقترح طباعة الرقعة بطباعة B1 مثلاً مكان خلية الحائط السوداء التي الرقم فيها هو

<sup>2</sup> نذكر أنه يمكنك دائماً تجميع الحلول المختلفة التي تعطيها أي إسنادية (حقائق ثابتة أو قواعد) ضمن سلسلة لاستخدامها لاحقاً، باستخدام تعليمة `findall` يؤمن برولوج إجرائية لتحقيق هذا الهدف تدعى `findall`، والتي تكتب على الشكل: `findall(X, p(...,X,...), L)`، حيث تعيد سلسلة L تحوي على جميع قيم X التي تحقق الإسنادية p. (طبعاً يجب أن تحوي p على المتحول X).  
مثال: ليكن لدينا الإسنادية التي تعطي مجموعة ثنائيات...

`get_a_cell(Row,Col) :- get_row(Row), get_col(Col).`  
عندها يكون:

`findall(cell(Row,Col), get_a_cell(Row,Col), Empty_Board) → [cell(1,1), c(1,2), c(1,3) ...]`

<sup>3</sup> إن استدعاء الحل أكثر من مرة بدون إعادة تشغيل برولوج يؤدي إلى أن القواعد التي تم إضافتها عند التنفيذ السابق لم تحذف، وقد تؤدي إلى عمل البرنامج بشكل خاطئ، يمكنك كتابة قاعدة `clear` تحذف جميع الحقائق الديناميكية التي قمت بإضافتها باستخدام `retractall` واستدعائها قبل كل تنفيذ.

<sup>4</sup> هذه القاعدة (وقواعد أخرى في هذه المسألة) تتطلب المرور على جميع عناصر الجدول، هذا يشبه تطبيق حلقة `for` على الأسطر والأعمدة، طبعاً هذه التعليمة غير متوفرة في `prolog` ولكن يمكن كتابة القواعد على الشكل التالي لتقوم بنفس الأثر:

```
% rule that needs looping (e.g. print...)
```

```
loop_for_some_reason :- get_row(Row), get_col(Col) %predicate that generates
numbers from 1 to the size of the grid...
```

واحد، و B- مكان خلية الحائط التي لا تحوي على رقم، و-L مكان خلية المصباح، أما الخلايا الأخرى نكتفي بطباعة رمز \* (محاطة بفراغات للحفاظ على التنسيق).

3. حل اللعبة: تبدأ اللعبة بالتهيئة ثم تطبيق القواعد الثابتة ثم تطبيق القواعد المكررة (التي تكرر<sup>5</sup> مجموعة الخطوات ثم تتحقق من انتهاء الحل)، يجب أن تقوم بطباعة الرقعة بعد كل دورة منها للتأكد من تقدم الحل.

### تحدي أكاري ☺

إذا استطاع برنامجك حل المسائل من المستوى المتقدم ستنتال علامة أعلى من العلامة الكاملة (يمكن أن تعوض نقصاً في التقييمات) هذا يتطلب منك استخدام استراتيجيات حل أكثر تعقيداً.

---

```
process(Row, Col, ..) %do something with one solution
, fail. %so the rule will fire again!
```

كتابة القاعدة بهذا الشكل يؤدي إلى أنها ترد false في كل مرة، بالتالي هذا يدعوها إلى تجريب جميع الاحتمالات الممكنة، ولكن في النهاية هذه القاعدة ترد false دائماً (تفشل دائماً)، بالتالي عند استدعائها في أي قاعدة أخرى يجب أن نقوم بعملية negation (تطبيق not) على هذه القاعدة حتى لا تتسبب بفشل القاعدة التي استدعتها.

```
Solve(... ):-\+loop_for_some_reason, ...
```

<sup>5</sup> يمكنك تحقيق هذا التكرار بنفس فكرة المرور على الحلقات الموجودة في الطباعة.

## توجيهات :

4. ننصحك بتجريب القواعد كل على حدة، وعدم كتابتها كلها معاً وذلك لتسهيل عملية اكتشاف الأخطاء، يمكن الاستعانة بالتعليمة trace و gtrace التي تمكنك من القيام بـ debugging للقواعد التي تكتبها.
- يجب أن تقوم بتجهيز أكثر من رقعة لتجريب الحل عليها، يمكنك الحصول على رقع مختلفة من الروابط السابقة.
- لا ننصح بتقسيم طلبي المسألة بين طلاب المجموعة، وذلك لتعلق الحلول والقواعد ببعضها، وإنما يجب على جميع أفراد المجموعة وضع قواعد المسألة معاً، ثم يمكن تقسيم تحقيقها برمجياً بينهم، ويجب توضيح تقسيم العمل في التقرير.
- يحوي التقرير على جميع القواعد والإسناديات لحل المسألة مع شرح بسيط وواضح لكل منها، بالإضافة إلى أمثلة اختبار لتوضيح الفكرة عند الحاجة. مراعاة ترتيب التقرير، وجودة الحل، وكتابة المتحولات بأسماء معبرة.
- ننصح بالتحضير للمقابلة كي تتم خلال 5 دقائق، يتم خلالها عرض العمل الشخصي بمشاركة جميع طلاب المجموعة والنتائج العملية التي تم التوصل إليها من خلال مثال اختبار مجهز مسبقاً للمسألة كاملة ولكل خطوة جزئية على حدة.
- على جميع طلاب المجموعة المشاركة في حل الوظيفة، والطلاب الذين لم يشاركوا في الحل لن ينالوا أي علامة.
- أية وظيفة منقولة من جهة ثانية أو منقول منها أو مأخوذة من الإنترنت ستنال علامة الصفر حتماً، نتوخى الالتزام بأمانة الحل.

Good  
Luck!

مدرسو العملي