برنامج الرؤية

ستقوم ببرمجة برنامج رؤية لروبوت. يتم تخزين كل صورة تقوم كاميرا الروبوت بالتقاطها على شكل صورة بالأبيض والأسود في ذاكرة الروبوت. يتم تمثيل كل صورة بشبكة من البكسلات $H \times W$ بحيث تكون أسطر الشبكة مرقمة من 0 حتى $H-1$ والأعمدة مرقمة من 0 حتى $W-1$. يوجد هناك **بكسلين فقط باللون الأسود** في كل صورة بينما بقية البكسلات في الصورة هي بيضاء.

يستطيع الروبوت معالجة كل صورة عن طريق برنامج يتألف من تعليمات بسيطة. سيتم إعطاؤك القيم H, W, وعدد صحيح موجب K. سيكون هدفك كتابة تابع يقوم بإنتاج برنامج للروبوت, حيث من أجل أي صورة يقوم بتحديد فيما إذا كان **البعد** بين بكسلين باللون الأسود هو تماماً K. هنا، يكون البعد بين البكسل الموجود في السطر r_1 والعمود c_1 والبكسل الموجود في السطر $r2$$ والعمود $c$$ هو $x \geq 0$ و تساوي $-x$ إذا $x < 0$.

سنقوم بشرح الآن كيف سيعمل الروبوت.

تمثل ذاكرة الروبوت بمصفوفة كبيرة بشكل كافي من الخلايا مرقمة من 0. تخزن كل خلية إما 0 أو 1 وهذه القيمة بمجرد كتابتها لن تتغير. تخزن الصورة سطراً بسطر في خلايا مرقمة من 0 إلى $H \cdot W - 1$. يتم تخزين السطر الأول في خلايا مرقمة من 0 إلى $W-1$, وسيتم تخزين السطر الأخير في خلايا مرقمة من $(H-1) \cdot W$ إلى $H \cdot W - 1$. بشكل خاص، إذا كان البكسل الموجود في السطر i والعمود j هو أسود, ستكون قيمة الخلية $i \cdot W + j$ هي 1, وإلا فهي 0.

يتكون برنامج الروبوت من سلسلة من **التعليمات** مرقمة بأرقام صحيحة متسلسلة بدءاً من 0. عند تشغيل البرنامج، سوف يتم تنفيذ التعليمات واحدة تلو الأخرى. ستقوم كل تعليمة بقراءة قيم واحدة أو أكثر من الخلايا (ندعو هذه القيم **بمدخلات** هذه التعليمة) وإخراج قيمة واحدة مساوية لـ0 أو 1 (ندعو هذه القيمة بـ**خرج** هذه التعليمة). يتم تخزين خرج التعليمة i في الخلية $H \cdot W + i$. يمكن أن تكون مدخلات التعليمة i هي فقط الخلايا التي تخزن البكسلات أو خرج التعليمات السابقة، أي خلايا من 0 إلى $H \cdot W + i - 1$.

هناك أربعة أنواع من التعليمات:

- NOT: لها دخلاً واحداً فقط. سيكون خرجها 1 إذا كان الدخل 0, وإلا فالخرج سيكون 0.
- AND: لهل دخلاً واحداً أو أكثر. سيكون خرجها 1 إذا وفقط إذا **جميع** مدخلاتها هي 1.
- OR: لها دخلاً واحداً أو أكثر. سيكون خرجها 1 إذا وفقط إذا **واحد على الأقل** من مدخلاتها هو 1.
- XOR: تمتلك دخلاً واحداً أو أكثر. سيكون خرجها 1 إذا وفقط إذا كان عدد 1 في مدخلاتها هو **رقم فردي** .

يجب أن يكون قيمة خرج آخر تعليمة من البرنامج هو 1 إذا كان البعد بين البكسلين باللون الأسود هو مساوي تماماً لـ K, وإلا ستكون قيمة الخرج هي 0 .

تفاصيل برمجية

يجب عليك برمجة التابع التالي:

```
void construct_network(int H, int W, int K)
```

- H, W: تمثل أبعاد كل صورة تم أخذها عن طريق كاميرا الروبوت
- K: هو عدد صحيح موجب
- يجب على هذا التابع إنتاج برنامج للروبوت. من أجل كل صورة تم أخذها من قبل كاميرا الروبوت، يجب على هذا البرنامج تحديد فيما إذا كان البعد بين بكسلين باللون الأسود في هذه الصورة هو مساوي تماماً للقيمة K.

يجب على هذا التابع استدعاء واحد أو أكثر من التوابع التالية لإضافة تعليمات على برنامج الروبوت (الذي سيكون بشكل بدائي فارغ):

```
int add_not(int N)
int add_and(int[] Ns)
int add_or(int[] Ns)
int add_xor(int[] Ns)
```

- قم بإضافة التعليمات NOT , AND, OR, XOR كما يلزم.
- N (من أجل add_not): تمثل رقم الخلية التي ستقوم تعليمة NOT المضافة بقراءة الدخل منها.
- Ns (من أجل add_and, add_or, add_xor): هي مصفوفة تحتوي أرقام الخلايا التي ستقوم التعليمات AND, OR, or XOR المضافة بقراءة الدخل منها.
- يعيد كل تابع قيمة رقم الخلية التي تم تخزن خرج التعليمة فيها. يعيد الاستدعاء المتتالي لهذه التوابع أعداد صحيحة متتالية بدءاً من $H \cdot W$.

يمكن أن يتألف برنامج الروبوت من $10\,000$ تعليمة على الأكثر.

تستطيع كل التعليمات أن تقرأ $1\,000\,000$ قيمة بالمجموع.

بمعنى آخر، يجب أن ألا يتجاوز مجموع طول مصفوفات الدخل Ns لكل استدعاءات التوابع add_and, add_or and add_xor زائد عدد استدعاءات التابع add_not قيمة $1\,000\,000$.

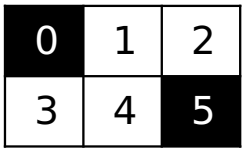
بعد استدعاء آخر تعليمة، يجب على التابعconstruct_network أن ينتهي. سيتم تقييم برنامج الروبوت على مجموعة من الصور. سينجح حلك باجتياز حالة اختبار معينة إذا تحقق من أجل كل صورة من هذه الصور كان خرج آخر تعليمة هو 1 إذا وفقط إذا كانت المسافة بين البكسلين باللون الأسود مساوية لـ K.

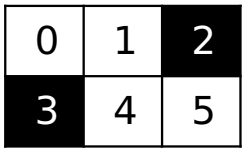
سينتج عن تقييم حلك واحدة من رسائل الخطأ التالية:

- Instruction with no inputs: تم تمرير مصفوفة فارغة إلى أحد التوابع add_and, add_or, أو add_xor.
- Invalid index: تم إعطاء خلية غير صحيح (ربما سالب) كدخل add_and, add_or, add_xor, أوadd_not.
- Too many instructions: حاول التابع الخاص بك إضافة أكثر من $10\,000$ تعليمة.
- Too many inputs: قامت التعليمات بقراءة أكثر من $1\,000\,000$ في المجموع.

أمثلة

لنفترض $H = 2$, $W = 3$, $K = 3$. يوجد فقط صورتين ممكنتين حيث المسافة بين البكسلين باللون الأسود هي 3.





- الحالة الأولى: البكسلين باللون الأسود هما 0 و5
- الحالة الثانية: البكسلين باللون الأسود هما 2 و3

أحد الحلول الممكنة هو بناء برنامج الروبوت عن طريق الاستدعاءات التالية:

1. `add_and([0, 5])`, والذي يضيف تعليمة تعطي 1 إذا وفقط إذا كانت الحالة الأولى محققة. سيتم تخزين الخرج في الخلية 6.
2. `add_and([2, 3])`, والذي يضيف تعليمة تعطي 1 إذا وفقط إذا كانت الحالة الثانية محققة. سيتم تخزين الخرج في الخلية 7
3. `add_or([6, 7])`, والذي يضيف تعليمة تعطي 1 إذا وفقط إذا كانت أحد الحالتين أعلاه محققة.

القيود

- $1 \leq H \leq 200$
- $1 \leq W \leq 200$
- $2 \leq H \cdot W$
- $1 \leq K \leq H + W - 2$

المسائل الجزئية

1. (10 نقطة) $\max(H, W) \leq 3$
2. (11 نقطة) $\max(H, W) \leq 10$
3. (11 نقطة) $\max(H, W) \leq 30$
4. (15 نقطة) $\max(H, W) \leq 100$
5. (12 نقطة) $\min(H, W) = 1$
6. (8 نقطة) البكسل في السطر 0 والعمود 0 يكون أسود في كل الصور.
7. (14 نقطة) $K = 1$
8. (19 نقطة) لايوجد قيود اضافية.

Sample grader

The sample grader reads the input in the following format:

- line 1: H W K
- line $2 + i$ $(i \geq 0)$: $r_1[i]$ $c_1[i]$ $r_2[i]$ $c_2[i]$
- last line: -1

Each line excepting the first and the last line represents an image with two black pixels. We denote the image described in line $2 + i$ by image i. One black pixel is in row $r_1[i]$ and column $c_1[i]$ and the other one in row $r_2[i]$ and column $c_2[i]$.

The sample grader first calls `construct_network(H, W, K)`. If `construct_network` violates some constraint described in the problem statement, the sample grader prints one of the error messages listed at the end of Implementation details section and exits.

Otherwise, the sample grader produces two outputs.

First, the sample grader prints the output of the robot's program in the following format:

- line $1 + i$ $(0 \leq i)$: output of the last instruction in the robot's program for image i (1 or 0).

Second, the sample grader writes a file `log.txt` in the current directory in the following format:

- line $1 + i$ $(0 \leq i)$: $m[i][0]$ $m[i][1]$ $\ldots$ $m[i][c-1]$

The sequence on line $1 + i$ describes the values stored in the robot's memory cells after the robot's program is run, given image i as the input. Specifically, $m[i][j]$ gives the value of cell j. Note that the value of c (the length of the sequence) is equal to $H \cdot W$ plus the number of instructions in the robot's program.