Towers

يقوم إياد بلعب لعبة على الهاتف المحمول حيث الهدف هو حماية القاعدة من الزومبي عن طريق وضع الأبراج. يمكن تمثيل القاعدة في هذه اللعبة كمصفوفة بحجم $N \times M$. يمكن لإياد وضع الأبراج في أي خلية لا تتجاوز حدود المصفوفة. تُعتبر القاعدة محمية إذا كان هناك على الأقل برج واحد في أي مربع $K \times K$.

ساعد إياد في إيجاد وضعية للأبراج ستحمي قاعدته. وبما أنه قد بدأ للتو باللعب وليس لديه الكثير من المال،

فمن بين جميع الوضعيات الممكنة، أخرج الوضعية التي تحتوي على أقل عدد من الأبراج.

Implementation Notes

You must implement the following function:

```
int32 solve(int32 N, int32 M, int32 K)
```

المدخلات N, M, K لها نفس المعنى كما ذُكر أعلاه، ويجب على الدالة أن تُرجع الجواب المحدد أعلاه.

ملاحظة: قد يتم استدعاء الدالة عدة مرات أثناء تنفيذ البرنامج.

Constraints

- $1 \leq N, M \leq 50$.
- $1 \leq K \leq \min(N, M)$.
- The function is called at most $50\,000$ times in a single execution.

Scoring

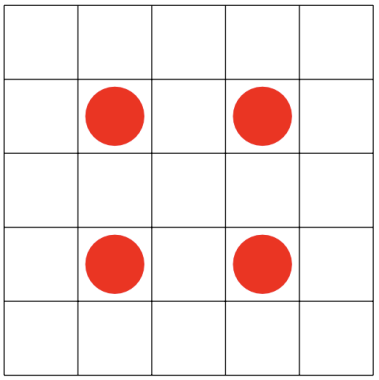
1. Subtask 1 (8 points): $K = 1$.
2. Subtask 2 (27 points): $K = 2$.
3. Subtask 3 (31 points):
4. $K | N$, $N = M$.
5. Subtask 4 (34 points): No additional constraints

Examples

في الصور التالية، الأبراج عبارة عن دوائر حمراء.

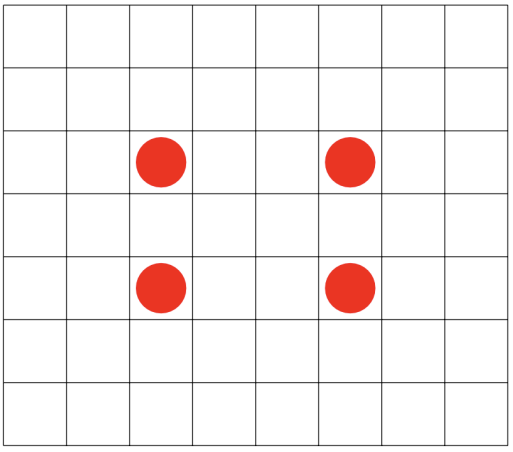
لننظر إلى الاستدعاء `solve(5, 5, 2)`.

في هذه الحالة ($N = M = 5, K = 2$)، نصنع الترتيب التالي:



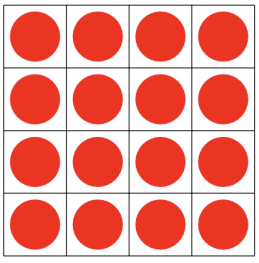
لننظر إلى الاستدعاء `solve(7, 8, 3)`.

في هذه الحالة ($N = 7, M = 8, K = 3$)، نصنع الترتيب التالي:



لننظر إلى الاستدعاء `solve(4, 4, 1)`.

في هذه الحالة ($N = M = 4, K = 1$)، نصنع الترتيب التالي:



Sample grader

المصحح التجريبي يقرأ T، عدد مرات استدعاء الدالة `solve`،

ثم T أسطر، يحتوي كل منها على القيم N, M, K لاستدعاء `solve`.

ثم يطبع القيم الناتجة عن استدعاء `solve` على أسطر مختلفة.

ملفات الإدخال والإخراج في الأمثلة تستخدم هذا التنسيق.