

بطاقات

لديك X بطاقة سوداء، Y بطاقة بيضاء، و Z بطاقة سماوية (المجموع $X + Y + Z$). ستلعب جميع البطاقات واحدة تلو الأخرى في كومة واحدة (مجموعة بطاقات موضوعة فوق بعضها بشكل مرتب) بأي ترتيب تختاره.

بعد وضع كل بطاقة، تحقق من الكومة:

- إذا احتوت الكومة (الآن) على بطاقة واحدة على الأقل من كل لون من الألوان الثلاثة، فستحصل على نقطة واحدة، ثم تُدمر جميع البطاقات باستثناء البطاقة الأخيرة التي لعبتها (تُزال من الكومة). وتبقى البطاقة الأخيرة التي لعبتها هي البطاقة الوحيدة في الكومة.
- وإلا فلن يحدث شيء.

مهمتك هي حساب الحد الأقصى للنتيجة (أي مجموع النقاط) التي يمكنك الحصول عليها.

تفاصيل التنفيذ

يجب عليك تنفيذ إجراء واحد يسمى `maximum_score`:

```
int64 maximum_score(int32 X, int32 Y, int32 Z);
```

- X : عدد البطاقات السوداء؛
- Y : عدد البطاقات البيضاء؛
- Z : عدد البطاقات السماوية؛
- قد يُستدعى هذا الإجراء على الأكثر 100 مرة لكل حالة اختبار في بداية البرنامج.

يجب أن يؤدي الإجراء إلى إرجاع الحد الأقصى للنتيجة التي يمكنك الحصول عليها.

قيود

- $1 \leq X \leq 10^9$
- $1 \leq Y \leq 10^9$
- $1 \leq Z \leq 10^9$

التنقيط

- المهمة الفرعية 1 (4 نقاط): $X = Y = Z = 1$
- المهمة الفرعية 2 (7 نقاط): $X + Y + Z \leq 8$
- المهمة الفرعية 3 (15 نقطة): $X + Y + Z \leq 16$
- المهمة الفرعية 4 (25 نقطة): $X, Y, Z \leq 50$
- المهمة الفرعية 5 (12 نقطة): $X = Y = Z$

6. المهمة الفرعية 6 (24 نقطة): $X < Y = Z$

7. المهمة الفرعية 7 (13 نقطة): لا توجد قيود إضافية

أمثلة

المثال 1

خذ بعين الاعتبار الإجراء التالي.

```
maximum_score(2, 2, 1);
```

ينبغي أن يعود الإجراء 2.

المثال الثاني

خذ بعين الاعتبار الإجراء التالي.

```
maximum_score(4, 3, 4);
```

ينبغي أن يعود الإجراء 5.

Grader

يقوم ال grader بقراءة المدخلات بالتنسيق التالي:

- السطر 1: عدد صحيح T ، يشير إلى عدد تنفيذات maximum_score
- الأسطر T التالية: ثلاثة أعداد صحيحة X و Y و Z

يقوم ال grader ببدء maximum_score(X , Y , Z) ويطبع القيمة المرتجعة.