

Cartes

Vous avez X cartes noires, Y cartes blanches et Z cartes cyan (total $X + Y + Z$). Vous les poserez une par une sur une seule pile, dans l'ordre de votre choix.

Après avoir placé chaque carte, vérifiez la pile :

- Si la pile contient (maintenant) au moins une carte de chacune des trois couleurs, vous marquez 1 point, puis toutes les cartes **sauf celle que vous venez de jouer** sont détruites (retirées de la pile). La carte que vous venez de jouer reste la seule carte de la pile. *Sinon, rien ne se passe.

Votre tâche consiste à calculer le score maximum que vous pouvez obtenir.

Détails d'implémentation

Vous devez implémenter une procédure appelée `maximum_score` :

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

- X : le nombre de cartes noires ;
- Y : le nombre de cartes blanches ;
- Z : le nombre de cartes cyan ;
- Cette procédure ne peut être appelée plus de 100 fois pour chaque cas de test au début du programme.

La procédure doit renvoyer le score maximum que vous pouvez obtenir.

Contraintes

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

Sous-tâches

1. Sous-tâche 1 (4 points) : $X = Y = Z = 1$
2. Sous-tâche 2 (7 points) : $X + Y + Z \leq 8$
3. Sous-tâche 3 (15 points) : $X + Y + Z \leq 16$

4. Sous-tâche 4 (25 points) : $X, Y, Z \leq 50$
5. Sous-tâche 5 (12 points) : $X = Y = Z$
6. Sous-tâche 6 (24 points) : $X < Y = Z$
7. Sous-tâche 7 (13 points) : Aucune contrainte supplémentaire

Exemples

Exemple 1

Considérez l'appel suivant.

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

La procédure devrait renvoyer 2.

Exemple 2

Considérez l'appel suivant.

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

La procédure devrait renvoyer 5.

Grader

Le grader lit l'entrée dans le format suivant :

- Ligne 1 : Un entier T , indiquant le nombre d'appels à `maximum_score`
- Lignes T suivantes : Trois entiers X , Y et Z

Le grader appelle `maximum_score(X, Y, Z)` et affiche la valeur renvoyée.