

TP n°2 : Diviser Pour Régner

Exercice 1: Inversions musicales

Un site internet cherche à regrouper ses membres en fonction des goûts musicaux de chacun. Pour cela, chaque membre doit classer par ordre de préférence une liste d'artistes. On dit que deux membres, Arthur et Béatrice, ont des goûts musicaux proches lorsqu'il y a peu d'inversions dans leurs classements : une inversion est une paire d'artistes $\{L, M\}$ telle qu'Arthur préfère L à M et Béatrice préfère M à L . On cherche donc à compter le nombre d'inversions dans les classements d'Arthur et Béatrice. On numérote les artistes de 1 à n dans l'ordre de préférence d'Arthur (avec 1 étant l'artiste préféré d'Arthur). On note a_1, \dots, a_n le classement de Béatrice et on suppose qu'il forme une permutation de $\{1, \dots, n\}$.

On appelle **inversion** toute paire d'indices (i, j) avec $i < j$ et $a_i > a_j$. Votre tâche est de compter le nombre total d'inversions dans le classement de Béatrice.

Entrée :

- La première ligne contient un entier n ($1 \leq n \leq 2 \cdot 10^5$) — le nombre d'artistes.
- La deuxième ligne contient une permutation a_1, a_2, \dots, a_n des entiers de 1 à n — le classement de Béatrice.

Sortie : Affichez un entier — le nombre total d'inversions dans la permutation.

Exercice 2: Un jeu de coloriage

Alice et Bob jouent à un jeu en utilisant un tableau d'entiers a de taille n .

Au départ, tous les éléments du tableau sont incolores. D'abord, Alice choisit 3 éléments et les colore en rouge. Ensuite, Bob choisit n'importe quel élément et le colore en bleu (si l'élément était rouge, il est recoloré en bleu). Alice gagne si la somme des éléments rouges est strictement supérieure à la valeur de l'élément bleu.

Votre tâche est de calculer le nombre de façons pour Alice de choisir 3 éléments de sorte qu'elle gagne **quelle que soit** l'action de Bob.

Entrée :

- La première ligne contient un entier t ($1 \leq t \leq 1000$), le nombre de cas de test.
- La première ligne de chaque cas de test contient un entier n ($3 \leq n \leq 5000$).
- La deuxième ligne contient n entiers a_1, a_2, \dots, a_n ($1 \leq a_1 \leq a_2 \leq \dots \leq a_n \leq 10^5$).
- Contrainte supplémentaire : la somme des n sur l'ensemble des cas de test ne dépasse pas 5000.

Sortie : Pour chaque cas de test, affichez un entier — le nombre de façons pour Alice de choisir 3 éléments de manière à gagner dans tous les cas.

Certains problèmes sur Codeforces sont **interactifs**, c'est-à-dire que votre programme doit communiquer avec le juge pendant son exécution.

Tutoriel rapide. Pour poser une question au juge :

- En **Python3**, utilisez `print(..., flush=True)` pour envoyer votre question et `input()` pour lire la réponse.

- En **C++**, utilisez `cout << ... << endl;` ou `cout << ...; cout.flush();` et `cin >> ...` pour lire la réponse.

Il est crucial de bien **vider le tampon** après chaque requête pour que le juge reçoive la question.

Exercice 3: Distance jusqu'à Mars

Natasha s'apprête à voler vers Mars. Elle s'ennuie pendant le vol et décide de calculer la distance jusqu'à la planète rouge. Soit x la distance inconnue à Mars. On sait que $1 \leq x \leq m$, avec x et m des entiers positifs connus par Natasha.

Natasha peut poser des questions à la fusée : un entier y ($1 \leq y \leq m$). La fusée doit répondre :

$$-1 \text{ si } x < y, \quad 0 \text{ si } x = y, \quad 1 \text{ si } x > y.$$

Mais la fusée est défectueuse : elle peut mentir. Plus précisément, il existe une séquence p de longueur n composée de 0 et 1. La fusée parcourt cette séquence de manière cyclique :

- Si l'élément courant est 1, elle répond correctement.
- Si l'élément courant est 0, elle répond l'opposée.

Natasha connaît n , mais pas le contenu de p . Vous pouvez poser au maximum 60 questions.

Interaction : Pour poser une question, imprimez un entier y et videz le tampon. Lisez ensuite la réponse du juge :

- Si la réponse vaut 0, la distance est correcte et vous devez immédiatement terminer le programme.
- Si la réponse vaut -2 , votre programme a dépassé les limites ou posé une requête invalide : terminez immédiatement.
- Toute entrée non valide ou oubli de vider le tampon peut entraîner une erreur ou le verdict « Idleness limit exceeded ».

Entrée : La première ligne contient deux entiers m et n ($1 \leq m \leq 10^9, 1 \leq n \leq 30$), respectivement la distance maximale possible et la longueur de la séquence p .

Sortie : Vous devez poser vos questions comme expliqué et terminer le programme dès que vous trouvez la distance exacte x .

Exercice 4: Sous-médiane maximale

Un entier v est une médiane d'un tableau $[a_1, \dots, a_n]$ de longueur n si et seulement si :

- v est supérieur ou égal à au moins $\lceil m/2 \rceil$ éléments du tableau,
- v est inférieur ou égal à au moins $\lceil m/2 \rceil$ éléments du tableau.

Par exemple :

- La seule médiane de $[9, 3, 7]$ est 7.
- Les médianes de $[5, 3, 7, 9]$ sont 5, 6 et 7.
- La seule médiane de $[2, 2, 2]$ est 2.

Un entier v de 1 à n est une **k -sous-médiane** de $[a_1, \dots, a_n]$ si et seulement s'il existe au moins un couple d'indices (l, r) tel que :

- $1 \leq l \leq r \leq n$,
- $r - l + 1 \geq k$,

- v est une médiane du sous-tableau $[a_l, \dots, a_r]$.

On peut démontrer qu'il existe toujours au moins une sous-médiane. Il s'agit de trouver la **sous-médiane maximale** v_{\max} et un couple (l, r) correspondant.

Entrée : La première ligne contient un entier t ($1 \leq t \leq 50000$) — le nombre de cas de test. Chaque cas de test est décrit de la manière suivante :

- La première ligne contient deux entiers n et k ($1 \leq k \leq n \leq 300000$).
- La deuxième ligne contient n entiers a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$).

Il est garanti que la somme des n sur tous les cas de test ne dépasse pas 300000.

Sortie : Pour chaque cas de test, affichez trois entiers v_{\max}, l, r : la k -sous-médiane maximale v_{\max} et les bornes d'un sous-tableau de longueur au moins k ($r - l + 1 \geq k$) pour lequel v_{\max} est l'une de ses médianes.

Si plusieurs solutions existent, vous pouvez en afficher une quelconque.