

B. Bonbon

Nom du problème	Bonbon
Limite de temps	3 secondes
Limite de mémoire	1 gigaoctet

Dans la ville antique Ica, on raconte qu'il existe un palais rempli de richesses inimaginables. À l'intérieur, il y a un couloir contenant N boîtes de bonbons venus du monde entier. Les voyageurs traversant ce couloir peuvent prendre autant de bonbons qu'ils le désirent, s'ils peuvent payer la masse de ces bonbons avec de l'or.

Les boîtes de bonbons sont numérotées de 0 à $N - 1$ de la gauche vers la droite. Dans la boîte i , il y a a_i unités de bonbons restantes, où a_i est un entier positif ou nul.

Comme vous gardez le palais, vous voulez déplacer les boîtes pour faire en sorte que celles ayant le plus de bonbons soient rapprochées de l'entrée.

On vous donne le tableau a_0, a_1, \dots, a_{N-1} , ainsi que les valeurs F et T . Une opération consiste à échanger deux éléments **adjacents** de a_0, a_1, \dots, a_{N-1} . Quel est le nombre minimal d'opérations requises pour que les F premiers éléments du tableau aient une somme d'au moins T ?

Entrée

La première ligne de l'entrée contient trois entiers, N , F , et T .

La deuxième ligne de l'entrée contient N entiers a_0, a_1, \dots, a_{N-1} .

Sortie

S'il est impossible d'arriver à l'objectif en utilisant les opérations disponibles, affichez "NO".

Sinon, affichez un unique entier, le nombre minimal d'opérations requises.

Contraintes et Score

- $1 \leq N \leq 100$.
- $1 \leq F \leq N$.

- $0 \leq T \leq 10^{11}$.
- $0 \leq a_i \leq 10^9$ pour $i = 0, 1, \dots, N - 1$.

Note : Les nombres donnés en entrée peuvent ne pas tenir dans un entier de 32 bits (int), prenez donc garde aux dépassements d'entier (overflows) si vous utilisez C++.

Votre solution sera testée sur un ensemble de groupes de tests (sous-tâches), chacun valant un certain nombre de points. Chaque sous-tâche contient un ensemble de tests. Pour obtenir tous les points d'une sous-tâche, vous devez valider tous les tests de cette sous-tâche.

Sous-tâche	Score	Contraintes
1	6	$N \leq 2$ et $a_i \leq 100$ pour $i = 0, 1, \dots, N - 1$ et $T \leq 10^9$
2	19	$a_i \leq 1$ pour $i = 0, 1, \dots, N - 1$
3	16	$N \leq 20$
4	30	$a_i \leq 100$ pour $i = 0, 1, \dots, N - 1$
5	29	Pas de contraintes additionnelles

Exemple

Dans le premier exemple de test, les deux premiers éléments doivent avoir pour somme au moins 27. Pour arriver à cet objectif, il suffit d'échanger deux valeurs adjacentes : échanger 4 et 20. Après cet échange, le tableau devient 10 20 4 6 3 3, et on a bien que la somme des deux premiers éléments vaut $10 + 20 = 30 \geq 27$.

Dans le second exemple de test, le 0 doit se déplacer jusqu'au bout du tableau ; cela prend trois échanges.

Dans le troisième exemple de test, il est impossible de faire en sorte que la somme des deux premiers éléments soit d'au moins 100 ; le maximum possible étant $60 + 30 = 90$.

Entrée	Sortie
<pre>6 2 27 10 4 20 6 3 3</pre>	1
<pre>6 5 5000000000 1000000000 1000000000 0 1000000000 1000000000 1000000000</pre>	3
<pre>3 2 100 20 30 60</pre>	NO
<pre>1 1 100 100</pre>	0