

beCP

2019

Tâche 1.1: Des boîtes lourdes (boxes)

Auteur: Robin Jadoul Préparation: Victor Lecomte
Limite de temps: 1 s Limite mémoire: 512 MB

Vous travaillez dans une usine et votre travail est d'emballer des objets dans des boîtes au fur et à mesure qu'ils sortent de la ligne d'assemblage. Aujourd'hui, n objets seront produits. Le i -ème d'entre eux aura un poids w_i . Vous disposez de k boîtes. Vous les remplirez une par une avec les objets produits. Vous ne pouvez pas remplir plusieurs boîtes en parallèle, ce qui veut dire que chaque boîte contiendra une suite continue d'objets : si vous placez les objets l et r dans une boîte, vous devrez aussi y placer tous les objets entre l et r . Chaque objet devra être placé dans une boîte.

Les boîtes vides sont si légères que le poids d'une boîte est juste la somme des poids des objets qui s'y trouvent. Les boîtes lourdes sont plus difficiles à transporter, de telle sorte que vous voulez *minimiser le poids de la boîte la plus lourde*.

Input

La première ligne de l'entrée contient deux entiers n et k : le nombre d'objets et le nombre de boîtes respectivement. La deuxième ligne de l'entrée contient n entiers : les poids w_i des objets, dans leur ordre d'arrivée.

Output

Imprimez un entier : le plus petit poids que la boîte la plus lourde peut avoir.

Limites générales

- $1 \leq n \leq 5 \times 10^5$, le nombre d'objets ;
- $1 \leq k \leq n$, le nombre de boîtes ;
- $1 \leq w_i \leq 10^{10}$, le poids du i -ème objet.

Notez que les poids peuvent être trop grands pour être contenus dans un entier 32 bits. En C++, utilisez **long long**.

Contraintes supplémentaires

Sous-tâche	Points	Contraintes
A	10	$k = 1$
B	20	$k = 2$
C	30	$n \leq 500$
D	25	$n \leq 5000$
E	15	Pas de contrainte supplémentaire

Exemple 1

sample1.in 3 2 1 4 2	sample1.out 5
----------------------------	------------------

Dans cet exemple, vous avez 2 boîtes. La solution optimale place les deux premiers objets dans la première boîte, ce qui donne un poids de $1 + 4 = 5$, et le troisième objet dans une deuxième boîte, ce qui donne un poids de 2. Dans ce cas, la boîte la plus lourde a un poids de 5. Il n'y a pas de solution où chaque boîte a un poids inférieur à 5. En particulier, n'oubliez pas que chaque boîte doit contenir une suite continue d'objets.

Cette entrée est acceptable pour les sous-tâches B, C et D.

Exemple 2

sample2.in 3 1 2 3 5	sample2.out 10
----------------------------	-------------------

Dans cet exemple, vous n'avez qu'une boîte. Vous devez donc mettre tous les objets dedans. Le poids de cette boîte sera de $2 + 3 + 5 = 10$.

Cette entrée est acceptable pour les sous-tâches A, C et D.

Exemple 3

sample3.in 1 1 10000000000	sample3.out 10000000000
----------------------------------	----------------------------

Exemple 4

sample4.in	sample4.out
5 3 1 2 3 4 5	6