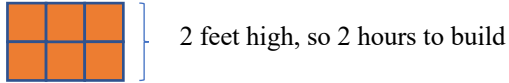


Bricklayers

We have to build $N > 0$ brick walls each the same length but with varying heights in feet $\{h_1, h_2, \dots, h_N\}$. It takes a bricklayer one hour to add a foot in height to a wall.



We have a team of $B > 0$ bricklayers.
We assign each wall to exactly one bricklayer.
Each bricklayer may be assigned zero or more walls.

What is the minimum amount of time it will take to build all N walls?

Write a function

```
int bricklayers(int b, int a[], int len)
```

where

b is the number of bricklayers

$a[]$ is an array of wall heights

len is the number of walls (i.e., the number of elements in $a[]$)

and returns the minimum time to build the walls if $b, len > 0$ and all $a[i] > 0$, otherwise returns -1

File you must submit: `soln_func.cc`

Examples:

$b=1, a[]=\{10, 10\}, len=2$

Returns: 20

Explanation: The only bricklayer ($b=1$) must build both walls. Minimum time is time it takes to build all the walls, one after the other.

$b=2, a[]=\{10, 30\}, len=2$

Returns: 30

Explanation: Two bricklayers, two walls, assign a bricklayer to each wall. Minimum time is the height of the tallest wall.

$b=5, a[]=\{40, 10\}, len=2$

Returns: 40

Explanation: Many more bricklayers than walls. Assign a bricklayer to each wall (other bricklayers do nothing). Minimum time is the height of the tallest wall.

$b=2, a[]=\{40, 10, 30, 20\}, len=4$

Returns: 50

Explanation: First bricklayer builds first two walls (40,10), second bricklayer remaining walls (30,20).

$b=3, a[]=\{40, 10, 30, 20\}, len=4$

Returns: 40

Explanation: 1st bricklayer builds first walls (40), other two bricklayers are assigned (10,30) (20) or (10,20) (30).

$b=0, a[]=\{40, 10, 30, 20\}, len=4$

$b=1, a[]=\{1\}, len=0$

$b=1, a[]=\{0\}, len=1$

Each returns: -1

Explanation: Each does not satisfy $b, len > 0$ and all $a[i] > 0$.