Talamontes

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

**Descripción**

El Gobierno de la República desea ser más ecológico y preservar los árboles, talando menos y sembrando más; como uno de sus compromisos de campaña. Se ha adquirido una máquina que corta los árboles estando en fila. La máquina trabaja especificándole una altura hh en metros en que la sierra se eleva, y luego corta todos los árboles que son más altos que hh (claro que los árboles que no son más altos de hhmetros, no se tocan). Luego las partes de los árboles que han sido cortadas, se llevan al camión de carga.

Por ejemplo, si hay una fila de árboles de altura 20, 15, 10 y 17 metros, y la sierra de la máquina se eleva 15 metros, la alturas de los árboles después de cortarlos sería 15, 15, 10 y 15 metros, respectivamente; así se tomará un tronco cortado de 5 metros del primer árbol y otro de 2 metros del cuarto árbol (7 metros de madera en total)

La Secretaría del Medio Ambiente no desea cortar más árboles ni madera de la necesaria. Es por eso que desean que la sierra esté lo más alto posible. Ayuda a la Secretaría a encontrar la altura entera máxima en que la sierra permita cortar a lo más mm metros de madera.

**Entrada**

Dos números positivos nn (número de árboles, 1≤n≤10000001≤n≤1000000) y mm (la cantidad de madera necesaria, 1≤m≤20000000001≤m≤2000000000). La segunda línea de la entrada contendrá nn enteros positivos separados por un espacio menor a 1 000 000 000, la altura de cada árbol en metros. La suma de todas las alturas no excederá a mm, así que siempre será posible obtener la cantidad de madera necesaria.

**Salida**

Solamente indicar la altura requerida

**Ejemplo**

|  |  |
| --- | --- |
| 4 7  20 15 10 17 | 15 |
| 5 20  4 42 40 26 46 | 36 |

*Fuente:*

Problema subido por: [Jorge Fernández Quezada](https://omegaup.com/profile/jofeque/)

| **ID** | **Status** | **Porcentaje** | **Penalty** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/talamontes#problems/new-run) | | | | | | | | |
| 273e5099 | Respuesta parcialmente correcta | 33.00% | 0 | cpp | 3.05 | 0.00 |  |  |
| 4525b2bb | Respuesta parcialmente correcta | 33.00% | 0 | cpp | 3.05 | 0.00 |  |  |
| 2cacc78f | Respuesta parcialmente correcta | 33.00% | 0 | cpp | 3.05 | 0.00 |  |  |
| 8ca4b6e6 | Tiempo límite excedido | 33.00% | 0 | cpp | 2.96 | 4.81 |  |  |
| e546d10b | Respuesta correcta | 100.00% | 0 | cpp | 3.12 | 0.02 |  |  |
| **Envíos** | | | | | | | |

<https://omegaup.com/arena/problem/talamontes#problems>

#include <iostream>

#include <stdio.h>

#include <map>

#define ll long long int

using namespace std;

int main()

{

   int N, m;

   scanf("%d %d", &N, &m);

   ll arboles[N];

   for(int i = 0; i < N; i++) {

        scanf("%lld", &arboles[i]);

    }

   ll sierra = 0;

   for (int i = 0; i < N; i++)

   {

       sierra = std::max( sierra, arboles[i]);

   }

   ll total\_madera = 0;

   while (true)

   {

       total\_madera = 0;

       for (int i = 0; i < N; i++) {

           if (arboles[i] > sierra) {

               total\_madera += arboles[i] - sierra;

           }

       }

       if (total\_madera >= m) {

           break;

       }

       sierra--;

   }

   printf("%lld**\n**", sierra);

  //  system("pause");

    return 0;

}