

С. Защитники башни

Ограничение времени	1 секунда
Ограничение памяти	512Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В не очень далёкой галактике на планете Икс-Бомбикс тоже проводится чемпионат мира по футболу. На этой планете солнечные лучи в течение всего дня падают ровно вертикально вниз.

Возле главного стадиона расположена сторожевая башня, состоящая из n плит, расположенных друг над другом так, что их левые края прикреплены к общей колонне. Длина i -й плиты равна a_i .

Занимать пост на сторожевой башне стадиона вызвались m добровольцев - охранников. j -й доброволец имеет ширину плеч b_j , а высота любого из них меньше, чем расстояние между соседними плитами.

С башни открывается отличный вид на стадион, поэтому очень много добровольцев хотят на неё попасть. В то же время из-за техники безопасности при распределении охранников по плитам башни должны выполняться следующие условия:

- Охранник будет стоять на плите боком, поэтому ширина плеч охранника не должна превышать длины плиты.
- Охранник должен быть расположен на плите полностью - по краям плиты стоят защитные ограждения (чтобы с неё нельзя было упасть).
- Охранник должен полностью находиться под солнечными лучами (если он будет в тени хотя бы частью тела, то за время матча замерзнет).
- На одной плите может быть не более одного охранника (два добровольца не поделят место под солнцем).

Изучите **графическое представление** первого теста **ниже в примечании** для лучшего понимания задачи.

Вам необходимо расположить максимальное число добровольцев по плитам с выполнением описанных условий. Скорее, матч начнётся с минуты на минуту, а добровольцы так и не знают, кого возьмут охранять главный стадион галактики и кто сможет насладиться крутым видом на игру!

Формат ввода

Первая строка входных данных содержит два числа n и m ($1 \leq n, m \leq 2 \times 10^5$) — количество плит, находящееся в башне, и количество добровольцев соответственно.

Вторая строка входных данных содержит n натуральных чисел a_i ($1 \leq a_i \leq 10^{18}$) — длина i -й плиты в порядке снизу вверх.

Третья строка входных данных содержит m натуральных чисел b_j ($1 \leq b_j \leq 10^{18}$) — ширина плеч j -го добровольца.

Формат вывода

В единственной строке выведите максимальное число добровольцев-охранников, которых можно расположить на плитах, с учетом описанных условий.

Пример 1

Ввод	<input type="text"/>	Вывод	<input type="text"/>
5 3		3	
7 3 4 2 2			
3 2 1			

Пример 2

Ввод Вывод

2 1
2 10
11

0

Пример 3

Ввод Вывод

5 4
100 98 96 40 30
2 4 60 3

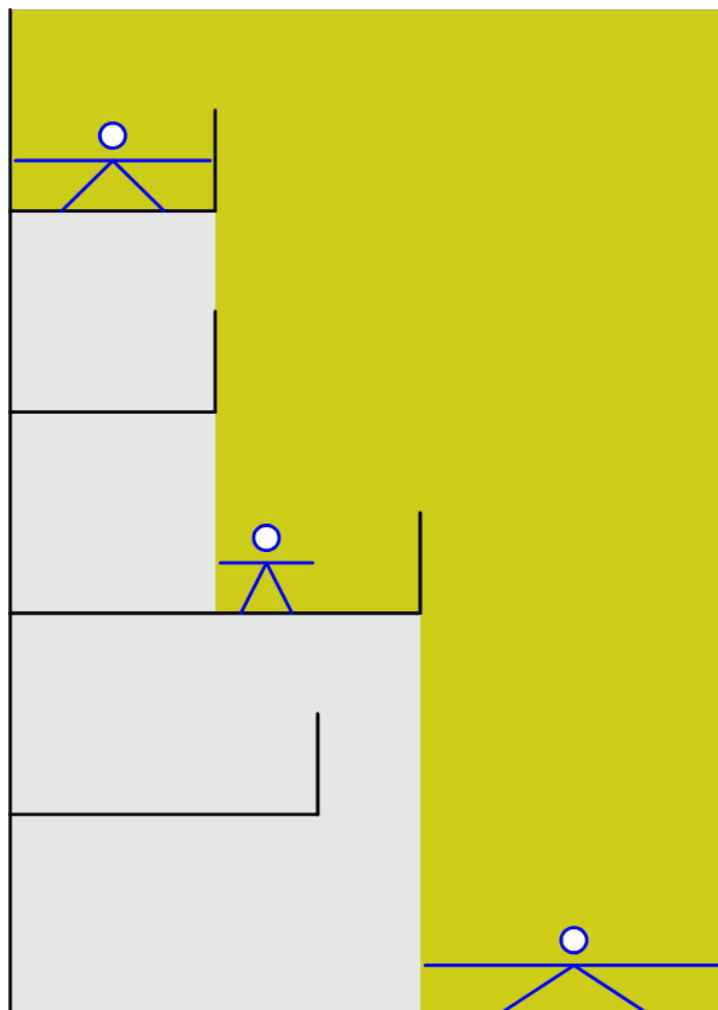
3

Примечания

В первом тесте из условия есть 3 светлых участка:

1. 5-ю (самую верхнюю) плиту солнце освещает полностью, поэтому на ней находится солнечный участок размера 2;
2. 4-я плита - так же размера 2, поэтому она полностью закрыта 5-й.
3. 3-я плита имеет общий размер 4, поэтому солнечный участок на ней имеет размер 2 (над остальной частью плиты нависают плиты 5 и 4);
4. 2-я плита имеет размер 3 и полностью закрыта от солнца плитой 3.
5. 1-я плита имеет общий размер 7, поэтому солнечный участок на ней имеет размер 3 (остальную часть закрывает 3-я плита).

Соответственно, доброволец с шириной плеч 3 займет место на солнечном участке 1-й плиты, а добровольцы с шириной плеч 1 и 2 могут встать на плиты 3 и 5 в любом порядке.



Язык GNU C++20 10.2

Набрать здесь

Отправить файл

```
1 #include <iostream>
2 #include <algorithm>
3
4 long long max_el(long long* a, long beg, long en)
5 {
6     long long max_v = 0;
7     for (long i = beg; i < en; i++)
8         if (a[i] > max_v)
9             max_v = a[i];
10    return max_v;
11 }
12
13
14 using namespace std;
15 int main() {
16     long n, m;
17     cin >> n;
18     cin >> m;
19
20     long long *layers = new long long[n];
21     long long *warriors = new long long[m];
22     long long max_v = 0;
23
24     for (long i = 0; i < n; i++)
25     {
26         cin >> layers[i];
27         if (layers[i] > max_v)
28             max_v = layers[i];
29     }
30
31     for (long i = 0; i < m; i++)
32         cin >> warriors[i];
33
34     for (long i = 0; i < n-1; i++) //Вычисление всех солнечных участков
35     {
36         if (layers[i] == max_v)
37             max_v = max_el(layers, i+1, n);
38         layers[i] -= max_v;
```

Отправить

Предыдущая

Следующая