

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	Вывод
5 3	3
7 3 4 2 2	
3 2 1	

Пример 2

Ввод	Вывод
------	-------

Ввод Вывод

```
2 1
2 10
11
```

0

Пример 3

Ввод Вывод

```
5 4
100 98 96 40 30
2 4 60 3
```

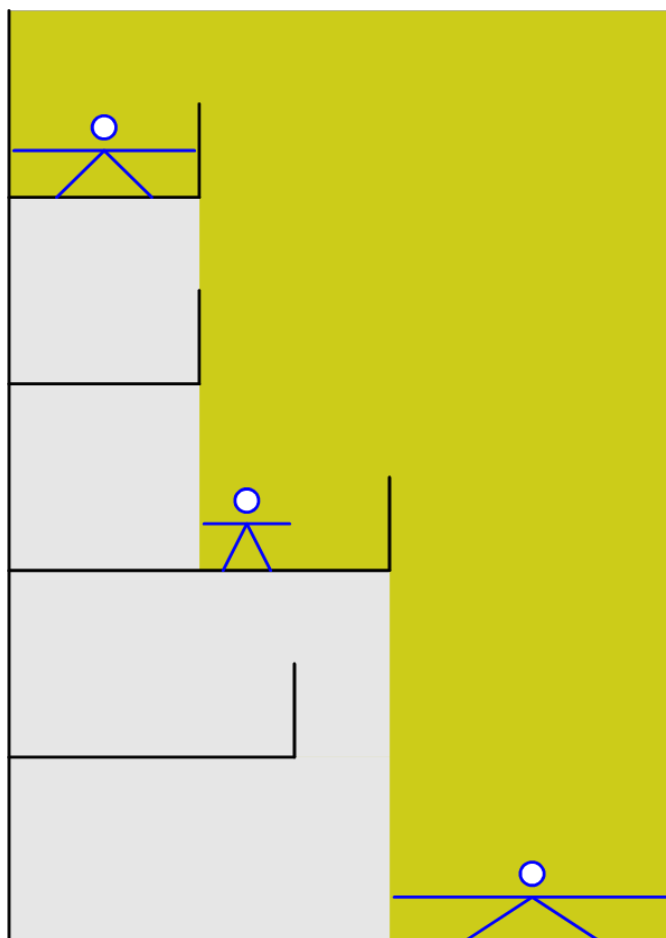
3

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 [Набрать здесь](#)[Отправить файл](#)

```
1 #include <iostream>
2 #include <vector>
3 #include <set>
4
5 using namespace std;
6 typedef unsigned long long ll;
7
8 int n, m;
9 vector<ll> plates;
10
11 multiset<ll> people;
12
13 multiset<ll> sunnyPlaces;
14
15 void foundSunnyPlaces() {
16     sunnyPlaces.insert(plates[n - 1]);
17     ll maxLength = plates[n - 1];
18     for (int i = n - 2; i > -1; i--) {
19         if (plates[i] > maxLength) {
20             sunnyPlaces.insert(plates[i] - maxLength);
21             maxLength = plates[i];
22         }
23     }
24 }
25
26 int main() {
27     ios::sync_with_stdio(false);
28     cin.tie(nullptr);
29     cout.tie(nullptr);
30     cin >> n >> m;
31     plates.resize(n);
32     for (int i = 0; i < n; i++) {
33         cin >> plates[i];
34     }
35     for (int i = 0; i < m; i++) {
36         ll temp;
37         cin >> temp;
38         people.insert(temp);
```

[Отправить](#)[Предыдущая](#)[Следующая](#)