

预览输出

题目名称	武器
题目类型	传统型
可执行文件名	midday.exe
输入文件名	midday.in
输出文件名	midday.out
每个测试点时限	1.0 秒
内存限制	512 MiB
测试点数目	20
测试点是否等分	是

提交源程序文件名

对于 C++ 语言	midday.cpp
-----------	------------

编译选项

对于 C++ 语言	<code>-O2 -std=c++14 -Wl,--stack=536870912</code>
-----------	---

武器 (midday)

【题目背景】

他们无休无止地搜罗着精良的武器。无论是“翼”的科技，还是“后巷”的发明，亦或是“郊区”的战利品，乃至“废墟”的遗物...

他们的所作所为，无非是紧握手中的武器，一如既往地用暴力解决眼中的一切。

【题目描述】

作为一名收尾人，正如前文所说，你需要高级的武器来强化自己，无论这个武器来自哪里，出自何方，只要可以强化自己的战斗力，就是优秀的武器。

不同的武器可以通过各种方法提供战斗力，包括但不限于身体强化，力量提升。当然，这些东西太具体了，我们就统统认为，对于第 i 把武器，其可以提供 B_i 的战斗力，装备上时，你的战斗力会增加 B_i 。

当然，这些强大的武器使用会有各种限制，其会存在两条限制：

- 装备武器也需要一定的战斗力，对于第 i 把武器，你自身的战斗力至少需要 T_i 才可以拿起。
- 除了拿起武器的时刻外，你无法同时携带两把武器，每当你携带武器的数量大于等于 2 把时，你需要舍弃至只剩下一把武器。被舍弃的武器所提供的战斗力，自然就丢失了。

很明显，打铁还需自身硬，你自身需要有强大的战斗力，才可以装上这些武器。

那么，问题很简单，给定 n 把武器和 q 次询问，每次询问给定你的初始战斗力 X ，问你最大可以取得的战斗力是多少？

【输入格式】

从文件 *midday.in* 中读入数据。

第一行包含两个正整数，分别是 n, q

第二行包含 n 个正整数，第 i 个数为 B_i

第三行包含 n 个正整数，第 i 个数为 T_i

第四行包含 q 个正整数，第 i 个数为第 i 次查询的 X

【输出格式】

输出到文件 *midday.out* 中。

输出 q 行，每行包含一个正整数，为第 i 次询问最大可以取得的战斗力。

【样例 1 输入】

```
1 3 2
2 21 15 7
3 4 6 12
4 6
5 1
```

【样例 1 输出】

```
1 27
2 1
```

【样例 1 解释】

对于第一组查询，可以拿起武器 1, 2，显然选择拿武器 1 更优秀。拿起武器后不需要拿其他武器。

【样例 2 输入】

```
1 5 10
2 1 2 3 4 5
3 2 4 6 8 10
4 1
5 2
6 3
7 4
8 5
9 6
10 7
11 8
12 9
13 10
```

【样例 2 输出】

```
1 1
2 3
3 5
4 7
5 9
6 11
7 12
8 13
9 14
10 15
```

【样例 2 解释】

对于第六组查询，可以拿起武器 3，此时战斗力到达 9，可以拿起武器 4，拿起武器 4 放下武器 3 后战斗力到达 10，可以拿起武器 5，然后放下武器 4。

最终战斗力为 $5 + 6 = 11$ 。

【样例 3】

见选手目录下的 *midday/midday3.in* 与 *midday/midday3.ans*。

【子任务】

对于 40% 的数据， $n, q \leq 1000$

对于 65% 的数据， $n \leq 10^3, q \leq 10^5; X, T_i, B_i \leq 10^5$

另有 15% 的数据，保证所有的 X 单调递增

对于 100% 的数据， $n, q \leq 10^5; X, T_i, B_i \leq 10^9$