Card game

Max. Marks: 100

Two friends decided to play a very exciting online card game. At the beginning of this game, each player gets a deck of cards, in which each card has some strength assigned. After that, each player picks random card from his deck and they compare strengths of picked cards. The player who picked card with larger strength wins. There is no winner in case both players picked cards with equal strength.

First friend got a deck with n cards. The i-th his card has strength a_i. Second friend got a deck with m cards. The i-th his card has strength b_i.

First friend wants to win very much. So he decided to improve his cards. He can increase by 1 the strength of any card for 1 dollar. Any card can be improved as many times as he wants. The second friend can't improve his cards because he doesn't know about this possibility.

What is the minimum amount of money which the first player needs to guarantee a victory for himself?

Input format

The first line of the input contains single integer n ($1 \leq n \leq 3 \cdot 10^5$) - the number of first friend's cards.

The second line of the input contains n space separated integers a_i ($1 \leq a_i \leq 10^9$) - the strength of the i-th first friend's card.

Second friend's cards are given in the next two lines in the same format.

Output format

Print single integer - the minimum amount of money which the first friend needs to guarantee a victory for himself.

SAMPLE INPUT

```
3
1 3 10
```

```
2
3 4
```

SAMPLE OUTPUT

```
6
```

Explanation

For 6 dollars first player can improve cards from $[1, 3, 10]$ to $[5, 5, 10]$. Second player has cards $[3, 4]$. All possible game situations:

1. $a_1 = 5 > b_1 = 3$
2. $a_1 = 5 > b_2 = 4$
3. $a_2 = 5 > b_1 = 3$
4. $a_2 = 5 > b_2 = 4$
5. $a_3 = 10 > b_1 = 3$
6. $a_3 = 10 > b_2 = 4$

First player always wins.

Time Limit:	1.0 sec(s) for each input file.
Memory Limit:	512 MB
Source Limit:	1024 KB
Marking Scheme:	Marks are awarded if any testcase passes.
Allowed Languages:	C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, Racket, Ruby, Rust, Scala, Swift, Visual Basic

CODE EDITOR

Enter your code or Upload your code as file. Save Python (python 2.7.6)

```
"""
    {
        "dateOfCreation": "29 Oct 2017",
        "codedBy": "Rishikesh Agrawani"
    }
"""

n = int(raw_input().strip())
strengths1 = [int(num) for num in raw_input().strip().split()]

m = int(raw_input().strip())
strengths2 = [int(num) for nu

max_strength = max(strengths2)
```

```
15 max_inc = max_strength + 1
16 amount = 0
17 for strength in strengths1:
18     if max_inc > strength:
19         amount += (max_inc - strength)
20
21 print amount
22
```

21:13

Provide custom input

Press Ctrl-space for autocomplete suggestions.

COMPILE & TEST SUBMIT

Submission ID: 12748047 / 2 seconds ago

RESULT: Accepted

Score	Time (sec)	Memory (KiB)	Language
100.0	9.21584	25560	Python

Input	Result	Time (sec)	Memory (KiB)	Score
Input #1	✓	0.110528	64	2
Input #2	✓	0.375231	25556	2
Input #3	✓	0.109806	64	2
Input #4	✓	0.186838	25560	2
Input #5	✓	0.192609	8320	2
Input #6	✓	0.187728	21656	2
Input #7	✓	0.341331	8320	2
Input #8	✓	0.359982	21860	2
Input #9	✓	0.109991	64	2
Input #10	✓	0.284531	8752	2
Input #11	✓	0.165926	8752	2
Input #12	✓			
Input #13	✓			
Input #14	✓	0.110204	64	2

Input #15	✓	0.110298	64	2
Input #16	✓	0.18987	8524	2
Input #17	✓	0.340845	25556	2
Input #18	✓	0.110108	64	2
Input #19	✓	0.195751	8320	2
Input #20	✓	0.31443	21852	2
Input #21	✓	0.109534	64	2
Input #22	✓	0.109939	64	2
Input #23	✓	0.215234	25556	2
Input #24	✓	0.163503	8320	2
Input #25	✓	0.15934	8320	2
Input #26	✓	0.109552	64	2
Input #27	✓	0.109551	64	2
Input #28	✓	0.192383	24268	2
Input #29	✓	0.160067	8320	2
Input #30	✓	0.198199	8320	2
Input #31	✓	0.109362	64	2
Input #32	✓	0.182249	25556	2
Input #33	✓	0.207264	21856	2
Input #34	✓	0.374819	21852	2
Input #35	✓	0.358563	8320	2
Input #36	✓	0.109013	64	2
Input #37	✓	0.202552	21860	2
Input #38	✓			
Input #39	✓			
Input #40	✓			

Input #41	✓	0.19212	8320	2
Input #42	✓	0.164652	21932	2
Input #43	✓	0.1096	64	2
Input #44	✓	0.110331	64	2
Input #45	✓	0.220216	21856	2
Input #46	✓	0.193984	25556	2
Input #47	✓	0.153563	22244	2
Input #48	✓	0.218022	22144	2
Input #49	✓	0.109825	64	2
Input #50	✓	0.109558	64	2

Compilation Log

No compilation log for this submission.

Tip: You can submit any number of times you want. Your best submission is considered for computing total score.

Your Rating: ★★★★★ Like 3 Share Tweet

COMMENTS (109) SORT BY: Relevance

Join Discussion...

Cancel Post

Ashish Dahiya 7 days ago

How to view testcases?
method 1: wait until the problem moves to practice problems... then submit ... click on any input#x to view the testcase
method 2: guess the testcases

▲ 7 votes ● Reply ● Message ● Permalink

Manish Billava 7 days ago

at the start of the game player1 knows play

▲ 1 vote ● Reply ● Message ● Permalink

Bohdan Pryshchenko Moderator 7 days

Yes.

● Reply ● Message ● Permalink