**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 nn cards. The ii-th his card has strength aiai. Second friend got a deck with mmcards. The ii-th his card has strength bibi.

First friend wants to win very much. So he decided to improve his cards. He can increase by 11 the strength of any card for 11 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 nn (1≤n≤3⋅1051≤n≤3⋅105) - the number of first friend's cards.

The second line of the input contains nn space separated integers aiai (1≤ai≤1091≤ai≤109) - the strength of the ii-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 66 dollars first player can improve cards from [1,3,10][1,3,10] to [5,5,10][5,5,10]. Second player has cards [3,4][3,4]. All possible game situations:

1. a1=5>b1=3a1=5>b1=3
2. a1=5>b2=4a1=5>b2=4
3. a2=5>b1=3a2=5>b1=3
4. a2=5>b2=4a2=5>b2=4
5. a3=10>b1=3a3=10>b1=3
6. a3=10>b2=4a3=10>b2=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

<https://www.hackerearth.com/es/challenge/competitive/october-circuits-17/algorithm/card-game-1-44e9f4e7/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int n = int.Parse(Console.ReadLine());

long[] a = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => long.Parse(e));

int m = int.Parse(Console.ReadLine());

long[] b = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => long.Parse(e));

//int n = 3;

//int[] a = { 1, 3, 10 };

//int m = 2;

//int[] b = { 3, 4 };

//int n = 3;

//int[] a = {2,2,2 };

//int m = 2;

//int[] b = { 3,3,3,3 };

//int n = 3;

//int[] a = { 0,0,0 };

//int m = 2;

//int[] b = { 1,2};

//int n = 3;

//int[] a = {4,4, 10 };

//int m = 2;

//int[] b = { 3, 4 };

//int n = 3;

//int[] a = { 0,0,0};

//int m = 2;

//int[] b = { 0,0,0,0};

long max\_b = b.Max();

long[] copia = new long[a.Length];

for (int i = 0; i < a.Length; i++)

{

if (a[i] <= max\_b)

{

//a[i] = max\_b + 1;

copia[i] = max\_b + 1;

}

else

{

copia[i] = a[i];

}

}

long sum = 0;

for (int i = 0; i < copia.Length; i++)

{

sum += copia[i] - a[i];

}

//Console.WriteLine();

Console.WriteLine(sum);

Console.ReadLine();

}

}

}