

## Zakład

Jasio założył się ze swoim kolegą, że do końca roku uda mu się mieć tą samą średnią. Niestety, rok dobiega końca, a Jaś jest dalej z tyłu. Wpadł więc na świetny pomysł - pozmienia oceny. Oczywiście, wyglądało by to podejrzanie, gdyby nagle doszło mu parenaście ocen, tym bardziej, że wzorowych. Z tego powodu, Jaś będzie zwiększał swoje oceny i zmniejszał oceny kolegi, dopóki ich średnie nie będą równe (albo Jasio nie będzie miał średniej większej). W każdym ruchu zmieni jakąś ze swoich ocen na dowolną inną lub jakąś ocenę kolegi na dowolną inną. Teraz chce wiedzieć, czy zdąży zmienić oceny tak, aby wyrównać średnie. Chce poznać w ile ruchów może takie zmiany wprowadzić. Napisał program, lecz ten nie działał poprawnie. Oto kod:

```
#include <iostream>
#include <queue>
#include <vector>
using namespace std;

int main()
{
    unsigned int nj, nk;
    cin >> nk >> nj;

    // Dla szybszego liczenia średniej
    unsigned long long suma_ocen_jasia, suma_ocen_kolegi;

    // Ta kolejka będzie przechowywać ocenę, którą najopłatalniej zwiększyć u Jasia
    priority_queue<unsigned int, vector<unsigned int>, greater<unsigned int>> oceny_jasia;
    // Ta kolejka będzie przechowywać ocenę, którą najopłatalniej zmniejszyć u Kolegi
    priority_queue<unsigned int, vector<unsigned int>> oceny_kolegi;

    // Wczytywanie ocen
    unsigned int a;
    for (int i=0; i<nj; i++) {
        cin >> a;
        oceny_jasia.push(a);
        suma_ocen_jasia += a;
    }

    for (int i=0; i<nk; i++) {
        cin >> a;
        oceny_kolegi.push(a);
        suma_ocen_kolegi += a;
    }

    unsigned int wynik=0;
    double srednia_jasia = double(suma_ocen_jasia) / double(nj);
    double srednia_kolegi = double(suma_ocen_kolegi) / double(nk);

    while (srednia_jasia < srednia_kolegi)
```

```
{
    // Pobieramy wartości, które hipotetycznie moglibyśmy wziąć do zmniejszenia/zwiększenia
    unsigned int zwiekszana_jasia = oceny_jasia.top();
    unsigned int zmniejszana_kolegi = oceny_kolegi.top();

    // Tworzymy średnie, które hipotetycznie moglibyśmy wziąć jako nowe średnie
    double nowa_srednia_jasia = double(suma_ocen_jasia-zwiekszana_jasia+6) / double(nj);
    double nowa_srednia_kolegi = double(suma_ocen_kolegi-zmniejszana_kolegi+1) / double(nk);

    // Sprawdzamy, czy bardziej opłaca nam się zwiększyć ocenę Jasia, czy zmniejszyć ocenę
    if (srednia_kolegi - nowa_srednia_jasia >= nowa_srednia_kolegi - srednia_jasia) {
        suma_ocen_jasia -= zwiekszana_jasia;
        suma_ocen_jasia += 6;
        oceny_jasia.push(6);
    } else {
        suma_ocen_kolegi -= zmniejszana_kolegi;
        suma_ocen_kolegi += 1;
        oceny_kolegi.push(1);
    }

    wynik++;
}

cout << wynik << endl;
}
```

**Twoim zadaniem jest napisanie rozwiązania, które:**

- Wczyta ilość ocen Jasia i kolegi oraz ich oceny;
- Obliczy w ile ruchów Jasiu może wyrównać średnią;
- Wypisze ilość ruchów.

### Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby  $1 \leq n_j, n_k \leq 10^6$  oznaczające ilość ocen Jasia i kolegi. W drugim wierszu znajdzie się  $n_j$  liczb, oznaczających oceny Jasia, gdzie każda liczba zawiera się w zbiorze  $\{1, 2, 3, 4, 5, 6\}$ . W trzecim wierszu znajdzie się  $n_k$  liczb, oznaczających oceny kolegi, zawierające się w tym samym zbiorze, co oceny Jasia.

### Wyjście

W pierwszym wierszu wyjścia powinna znaleźć się liczba, oznaczająca ilość zmian jakie Jaś wprowadzi w dzienniku.

### Przykład

Wejście:

3 4

2 2 3

4 5 4 3

Wyjście:

2

### **Wyjaśnienie do przykładu**

Jasiu może zmienić swoją dwójkę na szóstkę i piątkę kolegi na jedynekę.