



## Wstążki dla dziewczynek



### Wskazówka

Wczytujemy ilość wstążek ( $n$ ) oraz długości wstążek czerwonych ( $a[]$ ) oraz niebieskich ( $b[]$ ). Dla każdego koloru oddzielnie znajdujemy najmniejszą długość:  $x$  dla czerwonych,  $y$  dla niebieskich.

Zmienna *steps* (na początku wyzerowana) przechowuje nam ilość potrzebnych operacji. Przeładowujemy kolejne pary wstążek: na czerwonej musimy wykonać  $a[i] - x$  operacji, a na niebieskiej należy wykonać  $b[i] - y$  operacji. Zmienną *steps* powiększamy o większą z tych dwóch liczb.

### Kod w języku C++

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;
    cin >> n;
    int a[1000], b[1000], x, y;
    x = y = 1000000;
    for(int i = 0; i < n; i++)
    {
        cin >> a[i];
        x = min(x, a[i]);
    }
    for(int i = 0; i < n; i++)
    {
        cin >> b[i];
        y = min(y, b[i]);
    }
    int steps = 0;
    for(int i = 0; i < n; i++)
        steps += max(a[i] - x, b[i] - y);
    cout << steps << endl;
}
```

Wyszukiwanie najmniejszych elementów można wykonać przy pomocy funkcji `min_element()` – wtedy nie będzie potrzebne użycie do tego pętli.

## Kod w języku Python

Możemy posłużyć się funkcją `min()` dla list `a[]` oraz `b[]` w całości:

```
n = int(input())
a = list(map(int, input().split()))
b = list(map(int, input().split()))
x = min(a)
y = min(b)
steps = 0
for i in range(n):
    steps += max(a[i] - x, b[i] - y)
print(steps)
```