

# Praca zdalna

XVI OIJ, zawody II stopnia, zawody próbne  
11 marca 2022

Kod zadania: **pra**  
Limit czasu: **0.5 s (C++) / 6.5 s (Python)**  
Limit pamięci: **256 MB**



Bajtazar z racji pandemii pracuje zdalnie. Ponieważ wszyscy jego współpracownicy pracują ze swoich rodzinnych krajów z różnymi strefami czasowymi, coraz trudniej jest ustalić wspólny termin comiesięcznego spotkania. Spotkanie to powinno rozpocząć się o pełnej godzinie i trwać dokładnie godzinę.

Każdy pracownik ma kalendarz, w którym jest zaznaczony przedział czasu, w którym może wziąć udział w spotkaniu:  $i$ -ty pracownik zaczyna pracę o godzinie  $A_i$ , a kończy pracę **po** godzinie  $B_i$ . Oznacza to, że pracownik  $i$  może wziąć udział w spotkaniu o dowolnej godzinie od  $A_i$  do  $B_i$  (włącznie).

Każdy z pracowników jest także gotowy zostać po godzinach albo zacząć wcześniej niż to co zadeklarował. Nikt nie zrobi jednak tego za darmo: za każdą godzinę spędzoną dłużej w pracy należy zapłacić pracownikowi bajtdolara.

Wyznacz termin, w którym można zorganizować spotkanie tak, żeby każdy z pracowników (być może za dodatkową opłatą) mógł w nim uczestniczyć, a opłata za nadgodziny była jak najmniejsza.

## Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna  $N$  ( $1 \leq N \leq 100\,000$ ) oznaczająca liczbę pracowników. Następnie w  $N$  wierszach znajdują się dwie liczby całkowite  $A_i$  oraz  $B_i$  ( $0 \leq A_i \leq B_i \leq 10^9$ ) oznaczające okienko pracy  $i$ -tego pracownika, przy czym  $A_i$  to godzina rozpoczęcia pracy, natomiast  $B_i$  to godzina po której opuści on pracę.

## Wyjście

Na standardowe wyjście należy wypisać dwie liczby w pojedynczym wierszu – liczbę  $T$  oznaczającą moment rozpoczęcia spotkania oraz liczbę  $K$  oznaczającą sumaryczną opłatę za nadgodziny.

Jeśli istnieje wiele możliwych rozwiązań, Twój program może wypisać dowolne z nich.

## Ocenianie

Jeżeli jedynie jedna z dwóch wypisanych liczb będzie poprawna, otrzymasz za dany test 50% punktów przypisanych dla danego testu. Aby otrzymać te punkty, Twój program musi wypisać dwie liczby całkowite nie większe od  $10^{18}$  oraz musi zmieścić się w limicie czasu i pamięci i zakończyć uruchomienie bez błędu.

Możesz rozwiązać zadanie w kilku prostszych wariantach – niektóre grupy testów spełniają pewne dodatkowe ograniczenia. Poniższa tabela pokazuje, ile punktów otrzyma Twój program, jeśli przejdzie testy z takim ograniczeniem.

Dodatkowe ograniczenia	Liczba punktów
$A_i = B_i$	18
$N \leq 1000, A_i, B_i \leq 1000$	24
$N \leq 1000$	48
$A_i, B_i \leq 100\,000$	76

## Przykłady

Wejście dla testu pra0a:

```
3
1 5
3 4
7 8
```

Wyjście dla testu pra0a:

```
4 3
```

**Wyjaśnienie do przykładu:** Za 3 bajtodolary ostatni pracownik jest skłonny rozpocząć pracę o godzinie 4, wtedy spotkanie o godzinie 4 pasuje każdemu pracownikowi.

Wejście dla testu pra0b:

```
2
1 7
10 20
```

Wyjście dla testu pra0b:

```
7 3
```

Wejście dla testu pra0c:

```
5
1 2
3 3
4 5
1 3
3 5
```

Wyjście dla testu pra0c:

```
3 2
```

### Pozostałe testy przykładowe

- test pra0d:  $N = 1000$ ,  $A_i = B_i = i$  dla  $i = 1, 2, \dots, 1000$ .

