# Zadanie: PAR
# Para liczb

POTYCZKI ALGORYTMICZNE

**Potyczki Algorytmiczne 2026, runda próbna. Limity: 256 MB, 1 s.** *10.03.2026*

Algosia i Bajtek grają w Mikro-Totka, i opracowali niezawodną metodę typowania liczb, która zawsze kończy się wygraną.

W Mikro-Totku gracz musi wytypować dwie różne liczby całkowite $A$ i $B$ z przedziału $[1, 1000]$. Algosia zastosowała opracowaną wspólnie z Bajtkiem niezawodną metodą, i wytypowała liczby $A$ i $B$, które należy w dzisiejszym losowaniu zgłosić. Teraz musi je przekazać Bajtkowi, żeby zgłosił je w punkcie Mikro-Totka (sama będzie zajęta rozszerzaniem ich metody na inne gry losowe).

Niestety, cała klasa Algosi i Bajtka zauważyła, że coś jest na rzeczy (tona lizaków, którą zamówił Bajtek mogła mieć z tym coś wspólnego). Teraz wszyscy pilnie przysłuchują się każdemu słowu, które Algosia i Bajtek wymieniają.

Algosia z Bajtkiem wymyślili, jak zmylić wszystkich podsłuchiwaczy. Algosia poda Bajtkowi dwie liczby *inne* niż wybrane przez nią $A$ i $B$! A Bajtek domyśli się, jakie liczby Algosia pierwotnie wybrała. Oczywiście pewną trudnością jest to, jak dokładnie Bajtek ma domyślić się liczb Algosi. Tu właśnie z pomocą przychodzisz Ty.

Zaimplementuj algorytm, którego ma użyć Algosia by wybrać liczby $C$ i $D$ ($C \neq A$, $C \neq B$, $D \neq A$, $D \neq B$, $C \neq D$, $1 \leq C, D \leq 1000$), oraz algorytm, którego ma użyć Bajtek, żeby otrzymawszy liczby $C$ i $D$ odtworzyć pierwotne liczby $A$ i $B$.

## Interakcja

**To jest zadanie interaktywne.** Twój program zostanie uruchomiony w dwóch kopiach — jednej dla Algosi, a drugiej dla Bajtka. Każde z uruchomień w pierwszym wierszu wejścia otrzyma słowo "Algosia" lub "Bajtek", które określa, za działanie której z osób odpowiada ta kopia programu

**Wersja dla Algosi** W drugim wierszu wejścia dane będą dwie liczby całkowite $A$ i $B$ ($1 \leq A, B \leq 1000$, $A \neq B$). Twój program powinien wczytać je, i wypisać na standardowe wyjście jeden wiersz, zawierający dokładnie dwie liczby $C$ i $D$, spełniające $1 \leq C, D \leq 1000$, $C \neq A$, $C \neq B$, $D \neq A$, $D \neq B$, $C \neq D$.

**Wersja dla Bajtka** W drugim wierszu wejścia znajdą się dwie liczby całkowite $C$ i $D$, które wybrała Algosia. Twój program powinien je wczytać, a następnie wypisać na standardowe wyjście liczby $A$ i $B$, które muszą być tymi samymi liczbami, które wczytała wersja programu dla Algosi, w tej samej kolejności.

**Po wypisaniu odpowiedzi należy opróżnić bufor wyjścia, na przykład poprzez wywołanie polecenia `cout.flush()`, lub `fflush(stdout)` jeżeli używasz `printf`.** Należy przestrzegać podanego wyżej formatu wejścia i wyjścia, łącznie z białymi znakami i podziałem na wiersze — w przeciwnym przypadku program może zakończyć się werdyktem *Przekroczenie limitu czasu* zamiast *Błędna odpowiedź*.

### Przykładowa interakcja

| **Sprawdzarka do Algosi** | **Algosia do sprawdzarki** | **Sprawdzarka do Bajtka** | **Bajtek do sprawdzarki** | **Wyjaśnienie** |
|---|---|---|---|---|
| `Algosia` | | `Bajtek` | | Sprawdzarka informuję kopie programów Algosi i Bajtka o ich tożsamości |
| 20 26 | | | | Algosia ustaliła, że należy zgłosić liczby 20 i 26. |
| | 1 2 | | | Algosia podaje Bajtkowi liczby jeden i dwa. Bajtek na pewno się domyśli, o co chodzi. |
| | | 1 2 | | Ta sama para liczb, w tej samej kolejności, zostaje przekazana Bajtkowi. |
| | | | 20 26 | To było proste! Bajtek od razu wiedział o jakie liczby chodzi. |

**Test przykładowy:** Liczby, które należy zgłosić to 20 i 26.

## Eksperymenty

W sekcji *Pliki* dostępny jest przykładowy interaktor `parsoc.cpp`. Może się on nieznacznie różnić od tego używanego do sprawdzania rozwiązań. Należy go uruchamiać komendą:

```
python3 parrun.py [rozwiazanie] < [test] > [wyjscie]
```

przy czym plik `parsoc.cpp` musi być w tym samym folderze. Format, w którym interaktor przyjmuje wejście jest opisany w komentarzu w pliku `prasoc.cpp`.