Zadanie: POR
Porządki

Olimpiada Informatyczna

XXXII OI, etap II, dzień drugi. Plik źródłowy `por.*` **Dostępna pamięć: 256 MB.** *13.02.2025*

Bajtazar postanowił posprzątać swój pokój!

Ale nie idzie mu to zbyt dobrze. Jednym z problemów, które napotkał w trakcie sprzątania, jest ustawienie n książek na półce. Na półce jest n pozycji ponumerowanych od 1 do n i na każdej z nich znajduje się dokładnie jedna książka. Ponumerujmy książki od 1 do n w taki sposób, że docelowo książka 1 powinna stać na pozycji 1, książka 2 na pozycji 2, i tak dalej. Bajtazar potrafi szybko wyznaczyć liczbę inwersji w aktualnym ustawieniu. *Inwersją* nazywamy parę różnych pozycji i oraz j taką że $i < j$, na których stoją odpowiednio książki p_i oraz p_j, dla których zachodzi $p_i > p_j$.

Twoim zadaniem jest pomóc Bajtazarowi w ustawieniu książek na jego półce. Możesz wykonać co najwyżej m operacji. Jedna operacja polega na wybraniu dowolnych dwóch pozycji (być może takich samych), zamienieniu miejscami książek, które się tam aktualnie znajdują, a następnie uzyskaniu od Bajtazara informacji o liczbie inwersji w aktualnym ustawieniu. Po wszystkich Twoich zamianach książki na półce Bajtazara powinny być *posortowane*, to znaczy ustawione w kolejności $1, 2, \ldots, n$.

Komunikacja

To zadanie jest interaktywne. Należy napisać program, który będzie wykonywał operacje na książkach Bajtazara tak, aby na końcu były posortowane. W tym zadaniu komunikacja z procesem sprawdzającym odbywać się może albo przy użyciu biblioteki, albo przez standardowe wejście i wyjście. Ty wybierasz, którą opcję wolisz. **Nie należy używać obydwu metod jednocześnie.**

Przykładową implementację obydwu typów komunikacji można znaleźć w plikach dla zawodnika.

Twój program nie może otwierać żadnych plików. Program może korzystać ze standardowego wyjścia diagnostycznego (`stderr`), jednak pamiętaj, że zużywa to cenny czas. Jakiekolwiek nadmiarowe dane wypisane na standardowe wyjście mogą zostać potraktowane jako błędna odpowiedź.

Opcja 1: Komunikacja przy pomocy biblioteki

C++

Na początku programu należy napisać:

```
#include "porlib.h"
```

Biblioteka udostępnia następujące funkcje:

- `int DajN()` – Daje w wyniku liczbę książek n.
- `int Zamiana(int a, int b)` – zamienia książki na pozycjach a oraz b ($1 \leq a, b \leq n$); wynikiem funkcji jest liczba inwersji po wykonaniu zamiany. Gdy ta funkcja zwróci 0, to książki zostały posortowane i Twój program powinien zakończyć działanie, nie wykonując dalszych wywołań funkcji biblioteki.

Python

Na początku programu należy napisać:

```
from porlib import DajN, Zamiana
```

Biblioteka udostępnia następujące funkcje:

- `DajN()` – Daje w wyniku liczbę książek n.
- `Zamiana(a : int, b : int)` – zamienia książki na pozycjach a oraz b ($1 \leq a, b \leq n$); wynikiem funkcji jest liczba inwersji po wykonaniu zamiany. Gdy ta funkcja zwróci 0, to książki zostały posortowane i Twój program powinien zakończyć działanie, nie wykonując dalszych wywołań funkcji biblioteki.

Ogólne informacje

Funkcje biblioteczne można wywoływać w dowolnej kolejności, a tę zwracającą liczbę książek n można wywoływać wielokrotnie. Użycie którejkolwiek funkcji z niepoprawnymi argumentami będzie skutkować otrzymaniem werdyktu „błędna odpowiedź”. Podobnie wczytywanie ze standardowego wejścia i wypisywanie na standardowe wyjście może skutkować otrzymaniem werdyktu „błędna odpowiedź”..

Olimpiada Informatyczna finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025”.

Opcja 2: Komunikacja przez standardowe wejście/wyjście

W pierwszym wierszu standardowego wejścia znajdować się będzie jedna liczba całkowita n, oznaczająca liczbę książek. Następnie proces sprawdzający czeka na dwie liczby całkowite a i b, oznaczające pozycje do zamiany; należy je wypisać na standardowe wyjście, oddzielone pojedynczym odstępem i zakończone znakiem nowej linii. Następnie należy wczytać jedną liczbę całkowitą ze standardowego wejścia, oznaczającą liczbę inwersji po wykonanej zamianie.

Gdy wczytasz 0, to książki zostały posortowane i Twój program powinien zakończyć działanie, nie wykonując dalszych interakcji ze standardowym wejściem i wyjściem.

C++, wejście/wyjście strumieniowe

Należy standardowo załączyć odpowiedni nagłówek (`#include <iostream>`). Na końcu każdego wiersza przy wypisywaniu trzeba używać `std::endl`. Przykładowy początek komunikacji poniżej:

```
std::cin >> n;
std::cout << a << ' ' << b << std::endl;
std::cin >> number_of_inversions;
```

C++, wejście/wyjście przez `stdio`

Należy standardowo załączyć odpowiedni nagłówek (`#include <cstdio>`). Po wypisaniu każdego wiersza trzeba napisać `fflush(stdout)`. Przykładowy początek komunikacji poniżej:

```
scanf("%d", &n);
printf("%d %d\n", a, b);
fflush(stdout);
scanf("%lld", &number_of_inversions);
```

Python

Po wypisaniu każdego wiersza trzeba napisać `flush=True`. Przykładowy początek komunikacji poniżej:

```
n = int(input())
print(f"{a} {b}", flush=true)
number_of_inversions = int(input())
```

Przykładowy przebieg

Powiedzmy, że początkowe ustawienie książek to $1, 3, 2, 4, 6, 5$, a limit na liczbę zamian to $m = 1000$. Przebieg działania programu dla takich danych może być następujący.

Aktualne ustawienie przed akcją	Akcja	Argumenty	Zwrócona wartość
$1, 3, 2, 4, 6, 5$	DajN	–	6
$1, 3, 2, 4, 6, 5$	Zamiana	1, 2	3
$3, 1, 2, 4, 6, 5$	Zamiana	1, 2	2
$1, 3, 2, 4, 6, 5$	Zamiana	2, 3	1
$1, 2, 3, 4, 6, 5$	Zamiana	6, 5	0
$1, 2, 3, 4, 5, 6$	–	–	–

Testy przykładowe. Test 0 to test z przykładu powyżej. Poza tym:

`1ocen:` $n = 6$, $m = 1000$, początkowe ustawienie to $4, 2, 5, 1, 3, 6$;

`2ocen:` $n = 1000$, $m = 50\,000$, początkowe ustawienie to ciąg malejący;

`3ocen:` $n = 10\,000$, $m = 50\,000$, początkowe ustawienie składa się z dwóch rosnących łańcuchów; pierwsza połowa to liczby nieparzyste, druga to parzyste.

Olimpiada Informatyczna finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025”.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Ograniczenia	Punkty
1	$2 \leq n \leq 6$, $m = 200$	5
2	$2 \leq n \leq 100$, $m = 20\,000$	9
3	$2 \leq n \leq 1000$, $m = 25\,000$	23
4	$2 \leq n \leq 5000$, $m = 15\,010$	21
5	$2 \leq n \leq 10\,000$, $m = 19\,999$	42

Możesz założyć, że w każdym teście najpierw jest ustalana początkowa kolejność książek na półce, a dopiero później są obsługiwane operacje na książkach.

Dla zawodnika

W katalogu `dlazaw` znajdują się przykładowe (błędne) rozwiązania w C++ oraz Pythonie, które ilustrują obydwa modele komunikacji. Są też dostępne biblioteki do komunikacji oraz przykładowy program sprawdzający.

- `por.cpp` – przykładowe rozwiązanie demonstrujące komunikację przy pomocy `std::cin` oraz `std::cout`.
- `por.py` – przykładowe rozwiązanie demonstrujące komunikację przy pomocy `input()` oraz `print(...)`.
- `por2.cpp` – przykładowe rozwiązanie demonstrujące komunikację przy pomocy `porlib.h`.
- `por2.py` – przykładowe rozwiązanie demonstrujące komunikację przy pomocy `porlib.py`.

Aby skompilować przykładowe rozwiązania w C++ oraz program sprawdzający, możesz użyć komendy `make`, która tworzy pliki `por.e`, `por2.e`.

Program można uruchomić przy pomocy skryptu `run.sh`. Dla przykładu, wszystkie powyższe programy można uruchomić komendami:

- `./run.sh "./por.e"`
- `./run.sh "python3 por.py"`
- `./run.sh "./por2.e"`
- `./run.sh "python3 por2.py"`

Uruchomiony program wczytuje dane z pliku `input.in` w następującym formacie:

- w pierwszym wierszu: dwie liczby całkowite n i m,
- w drugim wierszu: permutacja liczb $1, \ldots, n$ opisująca początkowe ustawienie książek.

Liczby powinny być rozdzielone pojedynczymi odstępami.

Przykładowy program sprawdzający jest inny od tego używanego w SIO. W szczególności przykładowy program może nie sprawdzać poprawności wejścia ani argumentów wywołania funkcji. Gdy wyślesz rozwiązanie do SIO, zostanie ono sprawdzone na testach przykładowych za pomocą właściwego programu sprawdzającego.

Uwaga. W tym zadaniu nie są dostępne uruchomienia próbne.