

Zadanie «Liczby» (lic)

Napisać program `lic` do którego z wiersza poleceń przekazywane są dwa argumenty będące nazwami plików. Przykładowe wywołanie:

```
lic.exe plik1.txt plik2.txt
```

Pliki są plikami tekstowymi zawierającymi liczby całkowite z przedziału $1, \dots, 1000$, gdzie jeden wiersz zawiera tylko jedną liczbę.

Program powinien wyznaczyć liczby, które występują w `plik1.txt` (plik, którego nazwa jest przekazana w pierwszym argumencie) i nie występują w `plik2.txt` (plik, którego nazwa jest przekazana w drugim argumencie). Wyznaczone liczby należy wypisać na standardowym wyjściu. Liczby należy wypisać posortowane rosnąco, każda liczba może być wypisana tylko jeden raz. Każdy z plików powinien zostać odczytany tylko jeden raz.

Specyfikacja wejścia

Obydwa pliki mają identyczny format i spełniają następujące założenia:

- pliki są plikami tekstowymi,
- pliki zawierają wyłącznie liczby całkowite z przedziału $1, \dots, 1000$,
- każdy wiersz zawiera tylko jedną liczbę,
- ilość liczb w plikach nie jest znana,
- liczby w pliku mogą się powtarzać,
- porządek liczb w pliku jest nieokreślony,
- jeden z plików albo obydwa mogą być puste.

Specyfikacja wyjścia

Program powinien wypisać na standardowym wyjściu liczby, które występują w pliku, którego nazwa została podana w pierwszym argumencie i nie występują w pliku, którego nazwa została podana w drugim argumencie. Liczby należy wypisać w formacie:

- liczby należy wypisać w porządku rosnącym,
- każda liczba powinna być wypisana tylko jeden raz,
- każdy wiersz powinien zawierać tylko jedną liczbę,
- w przypadku braku rozwiązania należy wypisać tekst: `BRAK ROZWIĄZANIA`.

Przykładowe wejście A

wywołanie: `lic.exe plikA1.txt plikA2.txt`

zawartość plikA1.txt:

10
5
55
1
10
15
5
20
1

zawartość plikA2.txt:

35
15
8
31
5
20
15
20

Przykładowe wyjście A

1
10
55

Przykładowe wejście B

wywołanie: `lic.exe plikB1.txt plikB2.txt`

zawartość plikB1.txt:

8
12
7
8
13
8

zawartość plikB2.txt:

13
8
13
7
12
7

Przykładowe wyjście B

BRAK ROZWIĄZANIA

Przykładowe wejście C

wywołanie: `lic.exe plikC1.txt plikC2.txt`

zawartość `plikC1.txt`: pusty plik

zawartość `plikC2.txt`: pusty plik

Przykładowe wyjście C

BRAK ROZWIĄZANIA

Przykładowe wejście D

wywołanie: `lic.exe plikD1.txt plikD2.txt`

zawartość `plikD1.txt`:

8
12
7
8
13
8

zawartość `plikD2.txt`: pusty plik

Przykładowe wyjście D

7
8
12
13

Przykładowe wejście E

wywołanie: `lic.exe plikE1.txt plikE2.txt`

zawartość plikE1.txt:

15
5
5
20
5
10
20

zawartość plikE2.txt:

30
10
15
20
10
25

Przykładowe wyjście E

5

Ustalenia techniczne

1. Rozwiązaniem zadania są:

- program konsolowy napisany w języku C/C++ – Kod źródłowy programu powinien być zawarty wyłącznie w jednym pliku o nazwie `lic.c` (dla języka C) lub `lic.cpp` (dla języka C++). W pierwszej linii pliku źródłowego należy umieścić w komentarzu *indywidualny kod uczestnika* (IKU). Nie jest dopuszczalne umieszczanie w kodzie jakichkolwiek innych danych umożliwiających zidentyfikowanie uczestnika (także we właściwościach pliku).

Plik należy spakować do pliku `IKU-lic.zip`, gdzie IKU jest *indywidualnym kodem uczestnika*.

2. Należy przyjąć, że dane wejściowe mają poprawny format (opisany w treści zadania), nie ma konieczności sprawdzania poprawności danych wejściowych. Wyjście powinno mieć format opisany w treści zadania.
3. Należy przyjąć, że pliki z danymi wejściowymi istnieją.
4. W programach można korzystać wyłącznie ze standardowych bibliotek języka C/C++.
5. W programach nie można korzystać z rozwiązań i mechanizmów nieprzenośnych (np. zależnych od systemu operacyjnego).
6. Programy nie mogą:
 - tworzyć nowych procesów lub wątków,
 - uruchamiać innych programów,
 - używać funkcji sieciowych (np. `socket`, `send` itp.),
 - oczekiwać na interakcję użytkownika.
7. Zadanie należy przesłać przez stronę konkursu «Złoty Indeks» Platformy Zdalnej Edukacji korzystając z łącza do przesyłania rozwiązań zadania «lic».
8. Zadanie jest oceniane w skali 0-15 punktów. Składowe oceny:
 - obsługa argumentów przekazywanych z wiersza poleceń: 0–2 punkty,
 - odczyt danych z pliku: 0–1 punkt,
 - jednokrotny odczyt danych z plików: 0–2 punkty,

- wyznaczenie liczb wynikowych: 0–5 punktów,
- wypisanie liczb wynikowych: 0–1 punkt,
- wypisanie liczb posortowanych: 0–2 punkty,
- wypisanie każdej liczby jeden raz: 0–2 punkty.