

Zadanie 1: CIĄGI BINARNE (40 punktów)

Ciągi binarne to ciągi złożone z zer i jedynek. Ciągi binarne odpowiadają liczbom naturalnym zapisanym w systemie binarnym, np.

11001 odpowiada liczbie $(11001)_2 = 25$

Niech C_1 i C_2 będą ciągami binarnymi i niech l_1 i l_2 będą odpowiednio odpowiadającymi im liczbami. Powiemy, że $C_1 \leq C_2$ jeśli $l_1 \leq l_2$. Będziemy rozważać jedynie ciągi zaczynające się od 1, z wyjątkiem ciągu jednoelementowego 0 odpowiadającego liczbie 0.

Zadanie

Napisz program, który:

1. Wczyta z pliku wejściowego `bin.txt` zbiór ciągów binarnych,
2. Dla każdego ciągu C z pliku `pytania.txt` wypisze do pliku `odp.txt` słowo TAK, gdy ciąg C znajdował się w pliku `bin.txt` lub słowo NIE w przeciwnym przypadku.

Możesz założyć (choć nie jest to konieczne), że ciągi w pliku `bin.txt` podane są w kolejności niemalejącej, tzn. jeśli ciąg C pojawia się w tym pliku wcześniej niż ciąg B , to $C \leq B$. Nie można natomiast niczego takiego zakładać na temat ciągów z pliku `pytania.txt`.

Wejście

W pierwszym wierszu pliku `bin.txt` znajduje się liczba całkowita n , spełniająca nierówność $1 \leq n \leq 10^6$. W kolejnych n wierszach znajdują się ciągi binarne, każdy zapisany w postaci słowa złożonego z zer i jedynek.

W pierwszym wierszu pliku `pytania.txt` znajduje się liczba całkowita m , spełniająca nierówność $1 \leq n \leq 10^7$. W kolejnych m wierszach znajdują się ciągi binarne, każdy zapisany w postaci słowa złożonego z zer i jedynek.

Uwaga! Ciagi binarne w obu plikach mogą być **dowolnie długie**.

Wyjście

Plik `odp.txt` powinien zawierać m wierszy. W wierszu o numerze k powinno znaleźć się słowo TAK, jeśli ciąg binarny z k -tego wiersza pliku `pytania.txt` znajduje się w pliku `bin.txt` lub słowo NIE w przeciwnym przypadku.

Przykład

Dla pliku `bin.txt`:

[illegible]

i pliku pytania.txt:

[illegible]

Poprawny plik odp.txt ma postać:

TAK
NIE
TAK
NIE

Zasady punktowania

Punktowana będzie przede wszystkim poprawność i efektywność użytej metody. Oznacza to, że słabiej będą punktowane te rozwiązania, które nie działają poprawnie dla wszystkich danych wejściowych (np. gdy niektóre ciągi są zbyt długie). Podobnie słabsze oceny uzyskają programy o dużej złożoności czasowej. Dlatego przed rozpoczęciem programowania należy dokładnie zanalizować wymyślony algorytm (złożoność czasowa i pamięciowa).

Proszę także zwrócić uwagę na czytelność kodu i stosowanie komentarzy!

Forma rozwiązania zadania

Rozwiązanie **powinno** być napisane i przetestowane na komputerze. Program będący rozwiązaniem powinien być „aplikacją konsolową”, co oznacza, że nie powinien zawierać żadnych okien, powinien otwierać się w trybie tekstowym, czytać dane z plików wejściowych, obliczać rozwiązanie (być może równocześnie z czytaniem plików wejściowych) i zapisywać je w pliku wyjściowym.

Najlepiej napisać takie rozwiązanie w Delphi, można jednak użyć Turbo Pascala (program należy wtedy kompilować w trybie chronionym ze względu na możliwość dużego zapotrzebowania na pamięć operacyjną).

Program w Delphi będzie wyglądał tak samo jak ten w Pascalu, musi zawierać jednak napis `{$apptype console}`. Należy w tym celu wykonać następujące kroki (dla Delphi 5):

1. Wybrać z menu głównego Delphi „New...”.
2. W okienku, które się pojawi wybrać „Console Application” (utworzony zostanie nowy projekt, składający się tylko z jednego pliku).
3. Zapisać projekt pod nazwą `bin.dpr` i napisać w tym pliku swoje rozwiązanie.

Natomiast dla wcześniejszych wersji Delphi:

1. Utworzyć nowy projekt.
2. Usunąć z projektu plik formatki `Unit1.pas`.
3. Zapisać projekt pod nazwą `bin.dpr` i zmienić treść pliku `bin.dpr` na następującą:

```
program bin;  
uses  
    SysUtils;  
{ $apptype console }  
begin  
end.
```

4. Napisać rozwiązanie zadania w pliku `bin.dpr`.

Wskazówka techniczna kolejne znaki ciągu binarnego można czytać za pomocą polecenia `Read(f, ch)`, gdzie `f` jest plikiem tekstowym a `ch` jest zmienną znakową typu `Char`. Znaki czytamy dopóki nie napotkamy na koniec linii, tzn. dopóki `EOLn(f) = False`.

Rozwiązania można przynieść na dyskietce, lub przesłać pocztą elektroniczną jako załącznik na adres: kowalik@mimuw.edu.pl.

UWAGA! Proszę podpisywać rozwiązania (umieścić imię i nazwisko w pliku z rozwiązaniem) – także te wysyłane e-mailem !