



Zadanie: SUM

Sumowanie

PA 2017, runda finałowa. Dostępna pamięć: 256 MB. Limit czasu: 1 s.

17.12.2017

Twoje zadanie jest proste – masz dodać do siebie dwie liczby! Niestety, ponieważ na tym konkursie byłoby ono zbyt proste, musimy je nieco utrudnić.

Na potrzeby tych zawodów (i tylko na ich potrzeby) stworzyliśmy nowy, innowacyjny język skryptowy *PAScript*, którego głównym zastosowaniem jest męczenie finalistów Potyczek Algoritmicznych. Skrypt składa się z k wierszy; i -ty jego wiersz zawiera dwa niepuste słowa a_i, b_i złożone z drukowalnych znaków ASCII (o numerach od 33 do 126). Interpreter naszego języka pobiera z wejścia słowo s , a następnie wykonuje następujący pseudokod:

```
while true:
    foundReplacement = false
    for i = 1, 2, ..., k:
        if a(i) jest podsłowem s:
            zastąp najwcześniejsze wystąpienie a(i) przez b(i)
            foundReplacement = true
            break
    if not foundReplacement:
        break
print(s)
```

Twoim zadaniem jest znaleźć efektywny skrypt potrafiący dodawać do siebie liczby binarne. Wejściem do Twojego skryptu będzie napis postaci `liczba binarna + liczba binarna`, bez spacji ani zer wiodących w zapisie liczb. Przykładowym poprawnym wejściem skryptu jest `10001+1001`. Liczby na wejściu są całkowite nieujemne. Długość wejścia nie przekroczy 100 znaków.

Twój skrypt dla każdego takiego napisu powinien wypisać sumę tych liczb, również w postaci binarnej i bez zer wiodących. Dla powyższego przykładu, poprawnym wynikiem będzie `11010`.

Twój skrypt musi być również odpowiednio krótki i efektywny. Dokładniej, musi spełniać następujące warunki:

- $k \leq 50$ (skrypt nie może mieć więcej niż 50 reguł);
- $1 \leq |a_i|, |b_i| \leq 8$ (żadne słowo w skrypcie nie może być puste ani mieć więcej niż 8 znaków);
- Na każdym wejściu o długości maksymalnie 100 znaków interpreter wykona zewnętrzną pętlę maksymalnie 100 000 razy oraz w trakcie wykonania długość napisu s nie przekroczy 2017 znaków.

Wyjście

Pierwszy wiersz wejścia powinien zawierać pojedynczą liczbę całkowitą k ($1 \leq k \leq 50$) – liczbę reguł w skrypcie. Każdy z kolejnych k wierszy powinien składać się z dwóch niepustych słów a_i, b_i oddzielonych pojedynczym odstępem. Wiersze te powinny opisywać kolejne reguły skryptu.

Przykład

Przykładowe wyjście w poprawnym formacie:

```
6
00 0
01 1
10 1
11 1
0 NIE
1 TAK
```

Uwaga: wypisany program nie rozwiązuje powyższego zadania; pokazuje tylko format wyjścia. Skrypt ten dla napisu złożonego z zer i jedynek stwierdza, czy na wejściu podano co najmniej jedną jedynekę.