

Zadanie 2. Sufiksy

Słowo definiujemy jako ciąg złożony z małych liter alfabetu angielskiego.

Niech $s[1..n]$ będzie słowem o długości $n > 0$.

Sufiksem słowa s nazywamy każde jego podślowo kończące na ostatniej pozycji słowa s .

Sufiks $s[k..n]$ nazywamy k -tym sufiksem.

Przykład 1.

słowo $s[1..10] = \text{mascarpone}$ ma następujące sufiksy:

k	$s[k..n]$
1	mascarpone
2	ascarpone
3	scarpone
4	carpone
5	arpone
6	rpone
7	pone
8	one
9	ne
10	e

Uporządkowanie alfabetyczne wszystkich sufiksów słowa *mascarpone* daje następującą kolejność ich numerów (od najmniejszego): 5, 2, 4, 10, 1, 9, 8, 7, 6, 3:

k	$s[k..n]$
5	arpone
2	ascarpone
4	carpone
10	e
1	mascarpone
9	ne
8	one
7	pone
6	rpone
3	scarpone

Poniżej zapisano funkcję **czy_mniejszy($n, s, k1, k2$)**. Wynikiem funkcji jest wartość PRAWDA, gdy sufiks $s[k1..n]$ jest mniejszy w porządku alfabetycznym od sufiksu $s[k2..n]$ oraz FAŁSZ w przeciwnym przypadku.

Specyfikacja

Dane:

- n – długość słowa,
- $s[1..n]$ – słowo zapisane jako tablica znaków (numerowanych od 1),
- $k1$ – numer pierwszego sufiksu ($1 \leq k1 \leq n$),
- $k2$ – numer drugiego sufiksu ($1 \leq k2 \leq n, k1 \neq k2$).

Wynik:

PRAWDA jeśli sufiks $s[k1..n]$ jest mniejszy w porządku alfabetycznym od $s[k2..n]$, albo FAŁSZ – w przeciwnym wypadku.

czy_mniejszy ($n, s, k1, k2$)

$i \leftarrow k1$

$j \leftarrow k2$

dopóki ($i \leq n$ oraz $j \leq n$) **wykonuj**

jeżeli ($s[i] == s[j]$)

$i \leftarrow i + 1$

$j \leftarrow j + 1$

w przeciwnym razie

jeżeli ($s[i] < s[j]$)

zakończ z wynikiem PRAWDA

w przeciwnym razie

zakończ z wynikiem FAŁSZ

jeżeli ($j \leq n$)

zakończ z wynikiem PRAWDA

w przeciwnym razie

zakończ z wynikiem FAŁSZ

Zadanie 2.1. (0–2) 📄

Pierwsze dwie instrukcje *jeżeli* w funkcji *czy_mniejszy* wykonują porównania dwóch znaków słowa *s*.

Przykład:

dla danych $s = \text{mascarpone}$, $k1 = 5$, $k2 = 2$ algorytm wykona 3 porównania:

- (pierwsza instrukcja *jeżeli*) – sprawdzenie, czy $s[5] = s[2]$
- (pierwsza instrukcja *jeżeli*) – sprawdzenie, czy $s[6] = s[3]$
- (druga instrukcja *jeżeli*) – sprawdzenie, czy $s[6] < s[3]$

Podaj przykład słowa *s*, o długości ≤ 10 oraz liczb $k1$, $k2$, $k1 \neq k2$, dla których funkcja *czy_mniejszy* wykona dokładnie 6 porównań w pierwszej instrukcji *jeżeli*.

$s =$ _____, $k1 =$ _____, $k2 =$ _____

Zadanie 2.2. (0–2)

W plikach *slova1.txt*, *slova2.txt* i *slova3.txt* znajdują się po trzy wiersze:

- w pierwszym wierszu każdego pliku zapisana jest liczba całkowita dodatnia n , oznaczająca długość słowa
- w drugim wierszu zapisane jest n -literowe słowo *s*, składające się z małych liter alfabetu angielskiego a-z
- w trzecim wierszu zapisane są dwie liczby $k1$ i $k2$, oddzielone spacją.

Napisz program z zaimplementowaną funkcją *czy_mniejszy*. Jako wynik Twój program powinien wypisywać TAK lub NIE, w zależności od wyniku funkcji *czy_mniejszy*. Odpowiedzi dla poszczególnych plików zapisz w pliku *wyniki2_2.txt*.

Dla przykładowego pliku *sufiks_1.txt*, Twój program powinien dać odpowiedź: TAK, a dla przykładowego pliku *sufiks_2.txt* – odpowiedź: NIE.

Zadanie 2.3. (0–3)

Dana jest dodatnia liczba całkowita n oraz słowo $s[1..n]$. Naszym celem jest obliczenie wartości elementów tablicy $T[1..n]$ zawierającej numery sufiksów słowa $s[1..n]$ uporządkowanych w porządku alfabetycznym.

Przykład:

dla słowa *mascarpone* wynikowa tablica T to [5, 2, 4, 10, 1, 9, 8, 7, 6, 3],

dla słowa *kalafiorowa* wynikowa tablica T to [11, 4, 2, 5, 6, 1, 3, 7, 9, 8, 10].

Z wykorzystaniem funkcji `czy_mniejszy($n, s, k1, k2$)` zapisz w wybranej przez siebie notacji (w postaci pseudokodu lub w wybranym języku programowania) algorytm, który obliczy wartości elementów tablicy T zawierającej numery sufiksów zgodnie z porządkiem alfabetycznym sufiksów słowa s .

Uwaga: w zapisie możesz wykorzystać tylko operacje arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie, dzielenie całkowite, reszta z dzielenia), odwoływanie się do pojedynczych elementów tablicy, porównywanie liczb lub znaków, instrukcje sterujące i przypisania lub samodzielnie napisane funkcje zawierające wyżej wymienione operacje.

Specyfikacja

Dane:

- n – liczba całkowita dodatnia, długość słowa
- $s[1..n]$ – słowo zapisane jako tablica znaków (numerowana od 1)

Wynik:

- $T[1..n]$ – tablica T taka, że $T[i]$ -ty sufiks słowa s jest mniejszy w porządku alfabetycznym od $T[i + 1]$ -go sufiksu słowa s dla każdego $1 \leq i < n$.

Algorytm:

Zadanie 2.4. (0–3)

W pliku `slova4.txt` znajduje się 10 wierszy. Każdy wiersz zawiera liczbę n ($1 \leq n \leq 100$) oraz n -literowe słowo s składające się z małych liter alfabetu angielskiego. Dane w wierszu są oddzielone znakiem odstępu.

Napisz program, który dla każdego słowa s z pliku wypisze jego sufiks najmniejszy w porządku alfabetycznym.

Przykład:

Sufiksem najmniejszym w porządku alfabetycznym dla słowa *mascarpone* jest *arpone*, a dla słowa *truskawki* sufiksem najmniejszym w porządku alfabetycznym jest *awki*.

Dla przykładowego pliku `sufiks_4.txt`, zawierającego tylko 4 wiersze (ze słowami: *banan*, *mascarpone*, *abcaabbaabbccba*, *maturazinformatyki*), Twój program powinien dać odpowiedź:

an

arpone

a

aturazinformatyki