

Повишителен изпит по Увод в програмирането

СУ „Климент Охридски“, ФМИ, спец. Компютърни науки, 26.08.2019 г. Вариант А

Задача 1. Дадени са символни низове, съставени от малки латински букви и интервали ($\Sigma = \{ 'a', 'b', \dots, 'z', ' ' \}$). „Кодиране“ на низ s наричаме функция $\text{code}_s: \Sigma \rightarrow \Sigma$, която замества всяка буква от s с произволна буква от Σ , а буквите от Σ , които не се срещат в s , оставя непроменени. Да се дефинира функция

`void encode ([подходящ тип] s1, [подходящ тип] r1, [подходящ тип] s2, [подходящ тип] r2),`

където $s1$, $r1$, $s2$ и $r2$ са символни низове. Функцията да намира кодирането, което преобразува $s1$ в $r1$ и, ако такова съществува, да преобразува със същото кодиране низа $s2$, като резултата да запише в $r2$. Ако такова кодиране не съществува, в $r2$ да се запише празен низ.

Примери:

`encode("hello", "worrrd", "love", result)` ще запише в `result` низа `"rdvo"`

`encode("hello", "world", "love", result)` ще запише в `result` празния низ

Анализ на задачата:

Символните низове $s1$ и $r1$ ни дават информация за това коя буква с коя се заменя при кодирането. Ако вземем примера с `"hello"` и кодирания му вариант `"worrrd"` - на първо място забелязваме, че са с еднакви дължини, тъй като ако не бяха, то нямаше да съществува такова кодиране. На второ място забелязваме, че `'h'` се замества с `'w'`, `'e'` се замества с `'o'`, `'l'` се замества с `'r'` и т.н. и тук е момента, в който трябва да си дадем сметка, че ако веднъж дадена буква се замества с някоя друга, то ако по натам в низа се появи отново същата буква - в съответния кодиран низ трябва на нейната позиция отново да е буквата с която се е заместила първия път, иначе такова кодиране отново няма да съществува. Това ни навежда на мисълта, че ще е най-добре да отстраним този казус с това дали съществува или не подобно кодиране с една булева функция която връща `true` или `false` според това дали съществува или не подобно кодиране. След като вече разполагаме с такава функция, ако тя връща `true`, може да продължим с кодирането, защото вече ще сме сигурни, че за всяка буква от първоначалния низ съответства точно една буква от азбуката, тоест функцията `encode` не е нито *инекция* нито *сюрекция*. За оптимизация, докато правим проверката ще съхраняваме в масив с дължина $26+1$ (броя на буквите + празната клетка) на съответната позиция - кодиращия и символ. Например на позиция с индекс 0 в масива (първата) отговаря буквата `'a'`, на позиция с индекс 1 (втората) - отговаря буквата `'b'` и т.н., като накрая сме си оставили едно място и за празната клетка.

Задача 2. Дадена е матрица от символи `char C[3][3]`, и масив от n символни низа `words`, като всеки символен ни е с големина най-много 9 символа. Казваме, че една дума w може да се прочете в матрицата C , ако може да се намери последователност от съседни (споделящи *точно една* стена) клетки в C , чиито букви взети в този ред образуват думата w . Да се напише функция:

`bool crossword ([подходящ тип] C, [подходящ тип] words, unsigned n)`

Функцията да връща истина тогава и само тогава, когато всяка от думите в масива `words` може да се прочете в матрицата.

Пример:

c	a	t
w	h	o
w	i	n

При матрицата по-горе, функцията ще върне истина за масива с низове "cat", "what" и "not", но не и за масива с низове "cat", "hot" и "chat".

Коментар:

Така както е зададено условието на задачата, не е еднозначно дефинирано дали при прочитането на дадена дума чрез матрицата са допустими *застъпвания*, където „застъпване“ ще дефинираме като преминаване през една и съща буква повече от един път (*спазвайки посочените в условието правила*). В кода имплементиран по-горе както се вижда никъде не маркираме дадена буква след прочитането ѝ и следователно сме допуснали б.о.о., че са позволени застъпвания. Ако искаме да забраним застъпванията ще трябва когато намерим съвпадение с началната буква на дадена дума, във функцията, която търси съвпадение на цялата дума да маркираме буквата като *прочетена*, след което да я отмаркираме.

В решението сме допуснали че в задачата не се допускат застъпвания (промени единствено на редове 25, 26 и 32).