

Lexicographic sorting based on a table of words

Propose an implementation in C language of the sort by base applied to an array of words, that is strings of characters formed only with unaccented letters of the alphabet.

The principle of sorting by base of a table of words can be summarized as follows:

Input: an array T of nbwords words and the longmax the length of the largest word in T

Output (side effect): array T sorted in ascending lexicographic order

Principle:

for k ranging from 1 to longmax **do**

Perform a sort by count of T on the Kth character from the end of the words

end for

This principle is based on the sorting by enumeration of the words of the table that is carried out on each of the characters in starting with the rightmost characters.

For this class where words are handled, the base used for sorting by base is therefore equal to 27: the 26 lowercase letters of the alphabet and the null character "\0" which is the equivalent of 0 in integers. We will therefore only handle a few words in letters lowercase.

Regarding the sorting by enumeration of the words of the table, we cannot directly use the stated principle in the course because it is not a question of counting the number of identical words (in the example of the course we count the number of equal integers) but the number of words whose kth letter from the end is the same.

Input: an array t of nbwords words and an integer k denoting the kth letter from the end of the words on which we sort by count

Output (side effect): the array t sorted in ascending order on the kth letter

Variables used:

Res -> an array of words. Since sorting by enumeration cannot be carried out in place, we use this table Res in the same size than t to perform the sort.

Numbers: a table of 27 integers, indexed by the rank of the letters of the alphabet (plus the index 0 for the null character) used to count the number of words with the same kth letter.

Indices: a table indexed by the rank of the letters of the alphabet and such as Indices[r] contains the index in the table result where the sequence of words whose kth letter is of rank r begins.

Principle:

1. Effectifs[r] must contain the number of words in the table whose Kth letter is that of rank r in the alphabet:

For i ranging from 0 to nbwords -1 **do**

 Calculate the rank r of the kth letter of the word t[i]

 Effectifs[r] := Effectifs[r] +1

2. Once we have the establishment plan we must calculate, for each letter of rank r in the alphabet, the index in the result table where the sequence of words with the k th letter of rank r begins.

The words whose k th letter is null are to be placed at the start of the result array

Words whose k th letter is of rank r are to be placed from the index $\text{Indices}[r-1] + \text{Numbers}[r-1]$

$\text{Indices}[0] = 0$

For r ranging from 1 to 26 do

$\text{Indices}[r] = \text{Indices}[r-1] + \text{Numbers}[r-1]$

End for

3. Using the Indices table, we place each word in the table t in its place, according to its k th letter, in the table Res.

4. Then we copy the table Res into the table t

An example

Consider the following words that we want to sort in lexicographical order:

jour, le, mot, trois, maison, chat, longtemps, je, arbre, cheval.

The figure given in the appendix shows the start of the lexicographic sorting process with the calculation of the Numbers tables and Clues for the step taking into account the last letter of the words. We have visualized by - the null characters. Analyze example and continue sorting to the end.

TO DO :

You must implement in C language the lexicographic sorting of an array of words entered on the keyboard. You will need to use the [string.h](#) library in particular to calculate the length of words, but also to copy the strings in arrays because in C we cannot simply assign strings, we have to copy them with the [strcpy](#) function.

Be careful, this lexicographic sorting can only work with character strings containing only letters not accentuated but we do not ask to verify the entry of words.

0 jour	-----jour	-----le	---cheval	-----je
1 le	-----le	-----je	----chat	-----le
2 mot	-----mot	----arbre	---trois	----arbre
3 trois	---trois	---cheval	-----je	----chat
4 maison	---maison	---maison	-----le	-----mot
5 chat	----chat	----jour	---maison	longtemps
6 longtemps	longtemps	---trois	-----mot	---trois
7 je	-----je	longtemps	longtemps	----jour
8 arbre	----arbre	-----mot	----arbre	---maison
9 cheval	---cheval	----chat	----jour	---cheval

Effectifs	-:0	Indices	-:0
a:0		a:0	
b:0		b:0	
c:0		c:0	
d:0		d:0	
e:3		e:0	
f:0		f:3	
g:0		g:3	
k:0		k:3	
l:1		l:3	
m:0		m:4	
n:1		n:4	
o:0		o:5	
r:1		r:5	
s:2		s:6	
t:2		t:8	
u:0		u:10	