

RECURSION - elfish()

THE TEMPLATE

You are given a main function with helper functions that call the functions `elfish(char* strWord)` and `Xish(char* strX, char* strWord)` several times and check their boolean output. You are not meant to change these functions, but only to implement the functions `elfish` and `Xish`.

THE ASSIGNMENT

Part 1: Elfish (15 points)

Use recursion to implement the function `bool elfish(char* strWord)`. The function returns `true` if the word in `strWord` is `elfish`, based on the following definition:

- A word is elfish if it contains all the letters in the word “elf”.

Thus, `elfish()` should return `true` if `strWord` contains *at least one ‘E’, at least one ‘L’ and at least one ‘F’*. The order doesn’t matter, but you must check for **both upper case letters and lower case**. So the string “Elf” is elfish, as is the string “eLf” and the strings “elf” and “ELF”. The string “Elephant”, however, is not elfish, and should thus return `false`.

Part 2: X-ish (10 points)

Use recursion to implement a more general version of elfish called X-ish, `bool Xish(char* strX, char* strWord)`. The function returns `true` if the word in `strWord` is X-ish, based on the following definition:

- A word is X-ish if it contains all the letters in the string X.

Here you do *not* need to *worry about upper and lower case*, only return `true` if *all the exact characters* in the string `strX` are also somewhere in the string `strWord`.

Remember that you are free to make any helper functions that you like, like making a separate recursive function called by the base functions, maybe to have access to more function variables than

in the base functions. You can also change the base functions in any way, just making sure that the `main()` function can call them without any modifications to the code in the main function or the check functions.

Using the string, `char*` as a list of `char`:

Also remember that the `char*` that is sent into your functions is simply *a dynamic array of `char`*. You can view this as a **list**. In order to check the *first item* in that list you can dereference the pointer, either with `*strWord` (what the pointer points to, the first character) or `strWord[0]`. The **last char** in each string will always be `'\0'`.

In order to access a list that starts at the next item after the current `char*` you can use `(strWord+1)` or `&(strWord[1])`. Both are equivalent, only differ in syntax. This is a `char*` that is also a list of `char`, or a string, *that starts one item later in the list*.

SNIÐMÁTID

Gefið er `main` fall sem kallar ítrekað á föllin `elfish(char* strWord)` og `Xish(char* strX, char* strWord)` og athugar boolean úttakið úr þeim. Þér er ekki ætlað að breyta þessum föllum heldur einungis að útfæra föllin `elfish` og `Xish`.

ÚTFÆRSLA

Hluti 1: `Elfish` (15 stig)

Notið endurkvæma forritun til að útfæra fallið `bool elfish(char* strWord)`. Fallið skilar `true` en orðið í `strWord` er “elfish”, skv eftirfarandi skilgreiningu::

- Orð er “elfish” ef það inniheldur alla stafina í orðinu “elf”.

Þannig ætti `elfish()` að skila `true` ef `strWord` inniheldur *a.m.k. eitt ‘E’, a.m.k. eitt ‘L’ og a.m.k. eitt ‘F’*.

Röðin skiptir ekki máli en þú verður að **athuga bæði hástafi og lágstafi**. Þannig er strengurinn “Elf”

“elfish”, líkt og strengurinn “eLf” og strengirnir “elf” og “ELF”. Strengurinn “Elephant” er aftur á móti ekki “elfish” og ætti því að skila false.

Hluti 2: X-ish (10 stig)

Notið endurkvæma forritun til að útfæra almennari útgáfu af elfish sem er kölluð X-ish, **bool Xish(char* strX, char* strWord)**. Fallið skilar **true** ef orðið í **strWord** er “X-ish”, skv. eftirfarandi skilgreiningu:

- Orð er “X-ish” ef það inniheldur alla stafina í orðinu X.

Hér þarf *ekki* að hafa áhyggjur af *hástöfum og lágstöfum*, einungis að skila **true** ef allir stafirnir, *nákvæmlega eins og þeir eru í strengnum strX*, eru e-s staðar í strengnum **strWord**.

Mundu að þú mátt búa til hvaða hjálparföll sem henta, t.d. að búa til sér endurkvæmnisfall sem aðalfallið kallar í, ef þú vilt hafa aðgang að fleiri færíbreytum en aðalfallið býður upp á. Það má líka breyta skilgreiningunni á aðalfallinu, en þó einungis þannig að **main()** fallið kalli rétt á þau án nokkurra breytinga á **main()** og **check** föllunum.

Using the string, char* as a list of char:

Mundu einnig að **char*** er einfaldlega *kviklegt fylki af char*. Þú getur notað þetta eins og **lista**, með aðgang að fremsta stakinu og afgangnum af listanum. Til að kíkja á fremsta gildið í listanum er hægt að nota hvort sem er ***strWord** (það sem bendirinn bendir á, fremsta stakið) eða **strWord[0]**. Síðasti stafurinn í strengnum verður alltaf ‘\0’.

Til þess að vinna með lista sem byrjar á næsta staki á eftir núverandi fremsta staki er hægt að nota hvort sem er **(strWord+1)** (færir bendinn fram einn í listanum) eða **&(strWord[1])** bendir á næsta stak í listnum. Bæði gera nákvæmlega það sama, bara mismunandi syntax. Þetta skilar líka **char***, sem er *listi af char* (eða strengur) sem *byrjar einu staki aftur í listanum*.