B1- C-Pool

B-CPE-042

match - nmatch

Characters matching

v1.4

match - nmatch

Characters matching

repository name: : CPool_match-nmatch
repository rights: : ramassage-tek
language: : C
group size: : 1

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- Don't push your **main** function into your delivery directory, we will be adding our own. Your files will be compiled adding our **main.c** and our **my_putchar.c** files.
- You are only allowed to use the **my_putchar** function to complete the following tasks, but don't push it into your delivery directory, and don't copy it in *any* of your delivered files.
- If one of your files prevents you from compiling with * **.c**, the Autograder will not be able to correct your work and you will receive a 0.

You may use your Makefile lib, but it must be stored in the following directories (like any normal Pool day): **CPool_match-nmatch/lib/my** and **CPool_match-nmatch/include** (my.h file).

You are not to use any system functions, except *write*.

Match

The purpose of this function is to **find out if two strings match**, that is to say when they are identical.
If the second string contains an asterisk ('*'), this asterisk can be replaced with any character string (even an empty one) so that the two strings become identical.
For instance, *main.c* and **.c* match because it is possible to replace the asterisk with the *main* string, making them identical.
This second string can contain an unlimited number of asterisks.

The function must be prototyped as follows:

```
int match(char *s1, char *s2);
```

The function returns **1** if the strings match, and **0** otherwise.
Delivery: CPool_match-nmatch/match.c

Nmatch

The purpose of this function is to **count the number of times two strings match**.
When there are two or more asterisks, several string combinations are possible. Your function must calculate the total number of such combinations.
For instance, *abcbd* and **b** match two times: (*a*, *cbd*) and (*abc*, *d*)
abc and *a*** match three times: (*nothing*, *bc*) ; (*b*, *c* and (*bc*, *nothing*).

The function must be prototyped as follows:

```
int nmatch(char *s1, char *s2);
```

The function returns the number of combinations that match.
Delivery: CPool_match-nmatch/nmatch.c