

Segundo Parcial de Laboratorio

Algoritmos y Estructura de Datos II

TEMA B

Ejercicio 1

Completar en el archivo `palindrome.c` el código necesario para que funcione el algoritmo que verifica si una frase es *palíndromo*. Un palíndromo es:

"(...) una palabra o frase que se lee igual en un sentido que en otro (por ejemplo; Ana, Anna, Otto). Si se trata de números en lugar de letras, se llama capicúa." (extraído de [wikipedia](https://es.wikipedia.org/wiki/Pal%C3%ADndromo)).

Lo importante es tener en cuenta las letras y no así los espacios en el análisis, entonces la frase **"ana lava lana"** es palíndromo.

Deben completar en el código la creación de una cadena auxiliar que contenga la frase original sin los espacios. En particular se debe programar la función:

```
static char * remove_blanks(char *str, size_t *no_blanks_count)
```

que dada una cadena `str` debe devolver una nueva cadena (en nueva memoria) cuyo contenido debe ser el resultado de quedarse con los elementos que se encuentran en `str` que **no son espacios** (que no son el caracter `' '`). Adicionalmente la función debe dejar en `*no_blanks_count` la cantidad de elementos que no son espacios (o en otras palabras, el tamaño de la cadena resultante). Por ejemplo:

```
phrase = "ana lava lana";  
phrase_no_blanks = remove_blanks(phrase, &size_no_blanks);  
{phrase_no_blanks --> "analavalana" && size_no_blanks == 11}
```

El programa resultante no debe dejar *memory leaks*.

Una vez compilado el programa puede probarse ejecutando:

```
$ ./palindrome "ana lava lana"
```

Obteniendo como resultado:

```
es palíndromo!
```

Otro ejemplo de ejecución:

```
$ ./palindrome "una frase cualquiera"
```

que genera la siguiente salida

```
no es palíndromo
```

Más palíndromos de prueba

1. *"a mi loca colima"*
2. *"amar da drama"*
3. *"oso baboso"*
4. *"dabale arroz a la zorra el abad"*
5. *"luz azul"*
6. *"12321"*
7. *"1221"*