

## Tarea 5 - Programación y algoritmos

### Giovanni Gamaliel López Padilla

## Problema 1

### Problema 1a

Encuentre si la palabra 'p' esta presente en un string 'str', de otra manera regresa el número de veces que se encontro.

Se descargaron 20 parrafos de un documento de prueba llamado 'Lorem ipsum' [1]. Este documento es guardado en la carpeta **Problema.1a** con el nombre **test\_text.txt**. Se creo una función llamada **find\_word**. Esta función recibe como parámetros un puntero del tipo FILE y la palabra a buscar en el texto. El algoritmo que sigue la función **find\_word** es el siguiente:

```

1  // input text, word
2  // output count
3  count = 0
4  letter_behind = ' '
5  letter_file = read_character_from_file(text)
6  letter_word = word[0]
7  letter_final_word = word[-1]
8  while (letter != EOF)
9  {
10     if (letter_file == letter_word)
11     {
12         if (letter_behind != letter_word)
13         {
14             letter_behind = letter_file
15             letter_file = read_character_from_file(text)
16             i = 1
17             while (letter_file == word[i])
18             {
19                 letter_file = read_character_from_file(text)
20                 i += 1
21             }
22             if (i == word_size and letter_is_not_a_character)
23             {
24                 count += 1
25             }
26         }
27     }
28     letter_file = read_character_from_file(text)
29 }
```

En la linea 10 se comprueba si el carácter leído coincide con el primer carácter de la letra. Si coincide entonces se comprobará si los siguientes caracteres coinciden. En la linea 12 se comprueba si el carácter anterior a la primera coincidencia no es una letra, esto para no encontrar coincidencias en palabras distintas a las buscadas. En la linea 17 comienza la verificación de los demás caracteres, si coincidieron todos entonces la variable i debe ser del mismo valor que el tamaño de la palabra. En la linea 22 se comprueba si la variable i es del mismo tamaño que la palabra y si la letra siguiente no es una letra.

El programa se encuentra en la carpeta **Problema.1a**, el programa se compilo con el siguiente comando:

```
1 gcc -Wall -Wextra -Werror -pedantic -ansi -o main.out main.c -std=c11
```

Se usaron las palabras lorem, sapien, maecenas, aliquam, quam, tor y tor para probar el programa. Los resultados son los mostrados en la tabla 1.

Palabra	Resultados
lorem	9
sapien	10
maecenas	6
aliquam	18
quam	8
tor	0
tortor	6

**Tabla 1:** Resultados del programa 1a.

La manera de ejecutar el programa es la siguiente: Para hacer la búsqueda de la palabra lorem en el archivo `test_text.txt` se usa el siguiente comando.

```
1 ./main.out test_text.txt lorem
```

Se creo un script en bash el cual ejecuta el programa con las palabras mencionadas en la tabla 1. El script contiene las siguientes ejecuciones:

```
1 make clean
2 make
3 ./main.out test_text.txt lorem
4 ./main.out test_text.txt sapien
5 ./main.out test_text.txt maecenas
6 ./main.out test_text.txt aliquam
7 ./main.out test_text.txt quam
8 ./main.out test_text.txt tor
9 ./main.out test_text.txt tortor
```

## Problema 1b

## Problema 1a

Separe un string en tokens de acuerdo a un carácter especial dado como entrada. Debe regresar un arreglo que apunte a cada uno de los tokens encontrados.

## Problema 1c

Concatene dos strings creados con memoria dinámica. Regrese el resultado en el primer string.

## Problema 2

Programa que calcule el moving average y la mediana de una matriz de tamaño  $L_1 \times L_2$ . Reemplazar los valores de la matriz en su valor central.

## Referencias

[1] Lorem ipsum text. <https://es.lipsum.com/>.