

# TICS311: Tarea #2

## Universidad Adolfo Ibañez

Ahmad Armoush  
ahmad.armoush@edu.uai.cl

Danilo Bórquez Paredes  
danilo.borquez.p@uai.cl

Wilson González  
-

Miguel Romero  
miguel.romero.o@uai.cl

29 de marzo de 2022

## Objetivos

- Manejo de Arreglos en C
- Algoritmos de ordenamiento

## 1. Under Attack!

¡Su programa anterior fue un éxito! Gracias a su trabajo las personas de su equipo pueden enviarse mensajes codificados sin la preocupación de que una persona externa pueda leer el contenido del mensaje. Obviamente las claves utilizadas para la codificación deben ser bien aseguradas, pero ese es otro cuento...

Hace poco se han estado interceptando mensajes del bando rival. Utilizando ingeniería inversa lograron decodificar un mensaje enemigo como el mostrado en la siguiente figura:

```
mensaje.txt

first_name,danger_category,attack_prob
Kearney Carding,1,0.2733129
Antonin Revans,1,
Conney Lochrie,5,
Anton Sellack,2,
Lory Brydie,3,0.3140912
```

Al parecer el bando enemigo tiene una lista de posibles atacantes con sus respectivas categorías de peligro, y una probabilidad de ataque.

Nuestro jefe le está dando prioridad a este asunto, por lo cual le ha pedido su ayuda para lograr clasificar de buena manera los peligros que puedan generar los individuos de la lista recién mostrada, que puede llegar a tener miles de nombres...

### 1.1. Objetivo del trabajo

Generar una lista con los  $N$  nombres más peligrosos encontrados en el mensaje. La información que se tiene de cada nombre es la siguiente:

- `danger_category` indica el nivel de peligrosidad. Mientras más alto este número, el nombre es más peligroso
- `attack_prob` corresponde a la probabilidad de que el nombre nos ataque. Mientras más alto este valor, mayor es la probabilidad. Notar que este valor puede estar vacío, lo cual indica que no se tiene información de la probabilidad de ataque.

### 1.1.1. Criterios de peligrosidad

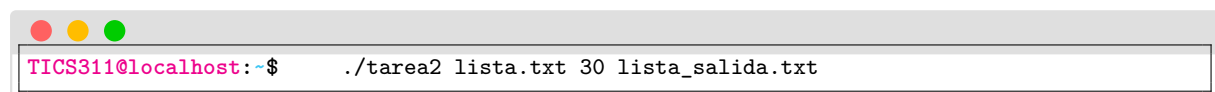
Para saber si un nombre es más peligroso se seguirán los siguientes criterios<sup>12</sup>:

- Un número mayor de `danger_category` implica mayor peligrosidad
- Un número mayor de `attack_prob` implica mayor peligrosidad
- Si existe un nombre sin `attack_prob`, entonces su peligrosidad será medida dependiendo de la categoría:
  - Si se encuentra en las categorías 1 o 2, entonces no se considera peligroso, quedando al final de la lista **de su respectiva categoría**.
  - si se encuentra en las categorías 4 o 5, entonces se considera muy peligroso, quedando al principio de la lista **de su respectiva categoría**.
  - si se encuentra en la categorías 3, entonces se considera medianamente peligroso, quedando en la mitad de la lista **de su respectiva categoría**
- Si existen dos nombres que empatan en peligrosidad (`danger_category` y `attack_prob`), entonces se ordenarán por orden alfabético.

### 1.2. Entrada del programa

Su programa recibirá como entrada sólo tres parámetros, correspondientes al nombre del archivo que contiene la lista de nombres, la cantidad de nombres que se quieren priorizar, y el archivo de salida que contendrá la lista generada.

A continuación un ejemplo de ejecución:



```
TICS311@localhost:~$ ./tarea2 lista.txt 30 lista_salida.txt
```

El ejemplo mostrado más arriba utilizará el archivo *lista.txt* que contiene la lista de nombres, y generará una lista con 30 de esos nombres, los cuales escribirá en el archivo *lista\_salida.txt*. El archivo de entrada siempre tendrá **a lo menos** la cantidad de nombres que se piden.

### 1.3. Salida del programa

El programa debe generar un archivo con los nombres (y sólo los nombres!) más peligrosos de la lista entregada

## 2. Sobre la entrega

- La tarea debe ser hecha en lenguaje de Programación C.
- Cada grupo puede ser de 2 o 3 personas.
- La tarea se debe entregar el día **Domingo 13 de Abril** a las 23:59.
- Por cada día de atraso se descontará 1 punto, comenzando a las 00:00 horas del siguiente día. Por ejemplo si entrega la tarea a las 00:00 del siguiente día, la nota máxima que puede obtener es un 6.0
- Para la corrección se utilizará un compilador gcc v 5.1 o superior
- La entrega se realiza por la plataforma Webcursos<sup>3</sup>

<sup>1</sup> Cuando se habla de **Su respectiva categoría**, implica que el nombre quedará al lado de otros nombres de la misma categoría

<sup>2</sup> Cuando el nombre queda en la mitad de su categoría, se refiere a que queda en la mitad de los nombres ordenados que comparten su categoría

<sup>3</sup> <http://webc.uaia.cl>

- El archivo a entregar debe ser un zip que contenga una carpeta en su interior (y sólo una carpeta) con el nombre **tarea2**. Dentro de esa carpeta debe haber un Makefile<sup>4</sup> y por lo menos un main.c. Además el nombre de su zip debe ser grupoX-tarea2.zip, donde X es el número de su grupo.
- IMPORTANTE: el directorio **tarea2** debe contener el Makefile que generará el archivo ejecutable **tarea2**.

---

<sup>4</sup>Este archivo deben generarlo. Información útil pueden encontrarla en <https://stackoverflow.com/questions/1484817/how-do-i-make-a-simple-makefile-for-gcc-on-linux>. Pueden probar su Makefile en un computador con MAC o Linux, o en VSCode en la terminal con UBUNTU. También puede probarse en la shell de <https://repl.it/>