



Tarea 2

Tabla de posiciones

Entrega: miércoles 2 de mayo a las 21:00 hrs.

1. Introducción

Un cliente muy apasionado de la programación en C y el fútbol le encargó desarrollar un programa en C para hacer un análisis de tablas de posiciones de fútbol en distintos campeonatos (mundial, campeonato nacional, copa Milo, etc.). Para esto, su programa recibirá un archivo de muchas líneas de texto según el siguiente formato:

```
pais pj puntos pg pe dg gf
```

Donde:

1. pj: Partidos Jugados
2. pg: Partidos Ganados
3. pe: Partidos Empatados
4. dg: Diferencia de Goles
5. gf: Goles a Favor
6. gv: Goles marcados como visita

Su programa deberá ejecutarse de la siguiente manera:

```
./tarea2 archivo [-w FILE -o ORDEN -f FILTRO -g -n NUM]
```

Donde:

1. **-w FILE:** La flag w, cambiará el modo en el que se presentan los datos, por defecto se deberán imprimir a la consola, pero si se incluye esta flag, se deberán escribir en el archivo **FILE**. Ej:

```
./tarea2 -w salida.txt
```



2. **-o ORDER:** Esta flag indica el criterio de ordenamiento. Por defecto deberá ser de mayor a menor según los puntos, pero si esta presente, deberá ser según el criterio que se indique en **ORDER**. El formato de **ORDER** es uno o más pares *COLUMNA ORIENTACION*, donde *COLUMNA* es el nombre de la columna, tal como se indica en la descripción del archivo, y *ORIENTACION* puede ser *asc* (de menor a mayor) ó *desc* (de mayor a menor). Cuando se indique más de una columna, la primera será el criterio principal de ordenamiento, y siempre la que viene después será el criterio de desempate en caso de que todo lo anterior sea igual. La *COLUMNA* también puede ser *pp*, *gc* y *gl* que son partidos perdidos, goles en contra y goles de local, respectivamente. Estos valores se calculan a partir de los demás datos entregados ($pp = pj - pg - pe$, $gc = gf - dg$, $gl = gf - gv$). Ej:

```
./tarea2 -o pts desc dg desc gf desc  
./tarea2 -o gc asc
```

3. **-f FILTRO:** Esta flag indica el criterio que se utilizará para filtrar los datos, por defecto no se aplica ningún filtro. El formato de **FILTRO** es *COLUMNA CRITERIO*, donde *COLUMNA* puede ser el nombre de una columna como se señala más arriba, o uno de los valores adicionales *pp*, *gc* y *gl* ya explicados. Por otro lado, el *CRITERIO* es una expresión matemática, de entre las siguientes:

- >NUM
- <NUM
- =NUM
- >=NUM
- <=NUM

Ej:

```
./tarea2 -f pp <3  
./tarea2 -f pe >=5
```

4. **-g:** La flag g indica que el modo de impresión deberá ser gráfico. El modo de impresión por defecto es numérico, es decir, si quiero imprimir que el equipo A posee 10 puntos lo haría de la siguiente manera:

10



Pero si lo quiero imprimir de la manera gráfica seria:

```
0000000000
```

Ej:

```
./tarea2 -g
```

5. **-n NUM:** El flag n indica la cantidad máxima (**NUM**) de países a ser impresos, por defecto se deben imprimir todos. Ej:

```
./tarea2 -n 10
```

Estas flags pueden aparecer en cualquier orden y ser procesadas correctamente.
Como requerimientos de programación, su cliente le ha pedido dos cosas:

- Pensando en futuros criterios de ordenamiento que puedan incluirse en nuevas versiones del software, y en la limpieza del código, cualquier comparación debe hacerse mediante una función que reciba como parámetro los índices de las filas que ocupen los equipos en el archivo y retornar el índice del que deba ir primero según el criterio dado o -1 si hay empate (funciones de la forma `int comparacion(int, int)`, y `comparacion(2,4)` sólo puede retornar 2, 4 o -1).
- Aprovechando lo anterior, deberá programar una sola función para ordenar, que reciba como parámetro un puntero a la función que compara según el criterio pedido.

2. Consideraciones Generales

Su programa debe ser robusto frente a datos erróneos. Esto quiere decir que su programa no debe caerse en caso de que el usuario haga operaciones matemáticas no válidas (p.ej., dividir por cero si fuera el caso). Sin embargo, pueden asumir que el usuario será amigable, en el sentido de que cuando le soliciten un número este no ingresará letras, por ejemplo.

3. Evaluación y Entrega

Formato de entrega: Subir un solo archivo comprimido con todo el código fuente de su programa al módulo de tareas de la página del curso en <https://saf.uandes.cl/ing/>, con el nombre de archivo



“NOMBRE-APELLIDO-Tarea1.tar.gz” el cual deberá contener los archivos de código fuente en C (archivos .c y .h), reemplazando “NOMBRE” y “APELLIDO” según corresponda (ej. ALAN-TURING-Tarea1.tar.gz). Los archivos compilados no serán tomados en cuenta, si se llega a subir solo un archivo compilado, este será ignorado, y será evaluado con nota 1.

En el momento de compilación, se recomienda incluir las siguientes flags al compilador:

```
-Wall -Wextra -Wundef -Werror -Wuninitialized -Winit-self
```

Aquellas tareas que no compilen de la forma normal (gcc FLAGS -o salida entrada) serán evaluadas con nota 1. Las que compilen, pero se caigan durante la ejecución, serán evaluadas con nota máxima 3.

Su programa será evaluado con múltiples casos de prueba y deberá ser capaz de ejecutarlos todos de manera correcta. De fallar en algún caso de prueba serán descontados los puntos correspondientes a dicho caso.

4. Consideraciones de Trabajo

El trabajo en esta tarea es estrictamente individual, por lo que cuide su tarea para que no sea copiada parcial o íntegramente por otros. Todas las tareas entregadas serán comparadas por un sistema automático de detección de plagio. Cualquier copia será penalizada, recibiendo el mismo castigo tanto quien copia como quien permite que le copien. También es considerada copia cualquier ayuda externa recibida directamente en la tarea, sin importar si proviene de un alumno del curso, de la universidad, o de otro lugar. El castigo será establecido por el Consejo de la Facultad, siendo como mínimo un 1,0 de promedio en el curso.