

# La misión de La'ebb

Algoritmos y programación II - 1C2023

Trabajo práctico Nº1

Individual



## Introducción

La'ebb es la mascota del último mundial y entre sus tareas esta llevar registro de todos los partidos que suceden en el mismo. A medida que pasaban los días se dio cuenta que no llegaba a registrar todos los partidos y comenzaron a amontonarse los resultados. Ahora que finalizó el mundial las personas quieren poder volver a ver a los distintos resultados de su equipo, pero La'ebb solo llegó a armar las tablas hasta cuartos de final. Por suerte se encontró con una catedra de programación dispuesta a darle una mano.

## Enunciado

Para completar la misión de La'ebb vamos a disponer de 2 archivos uno con los equipos que participaron del evento y otro con los resultados de cada partido.

### Equipos

El archivo de equipos se llamará **equipos.txt** y tendrá el formato:

**país grupo**

Por ejemplo:

```
Argentina C
Catar A
Ghana H
...
```

Este archivo se deberá leer línea por línea y cargar sus datos en un vector dinámico que se encuentre ordenado alfabéticamente por país utilizando inserción.

Si un país se compone de más de una palabra se pondrá con guion bajo y el mismo deberá ser removido a la hora de guardarlo. Ejemplo: Países\_bajos y al guardarlo Países bajos.

### Resultados

El archivo con los resultados se llamará **resultados.csv** y tendrá el formato:

**grupos**

**equipo1,goles ,equipo2,goles**

....

**fase**

**equipo1,goles,resultado\_penales,equipo2,goles,resultado\_penales**

Por ejemplo:

```
grupos
Ecuador,1,Senegal,2
Tunez,1,Francia,0
octavos
Marruecos,0,3,España,0,0
...
```

Si un país tiene más de una palabra se escribirá normalmente (con un espacio en el medio de ambas palabras), por ejemplo:

**Países bajos**,goles,resultado\_penales,equipo2,goles,resultado\_penales

**NOTA:** si no se juegan penales se encontrará **-1** en el lugar correspondiente.

Con esta información se deberá calcular la cantidad de puntos que consiguió cada equipo por fase usando la siguiente lógica:

Resultado	Puntos
Victoria	3
Empate	1
Derrota	0
Victoria por penales	2
Derrota por penales	1

El archivo debe leerse línea por línea y no se pueden almacenar las mismas.

Una vez leídos ambos archivos se deberá armar un programa que cuente con el siguiente menú:

1. Listar equipos.
2. Mostrar los equipos en primer segundo y tercer lugar.
3. Buscar equipo por nombre.
4. Mostrar por fase los países ordenados por puntaje.
5. Salir.

#### *Listar equipos*

Se deberán listar todos los equipos en el orden en el cual estén guardados con su nombre y grupo al que pertenecieron.

#### *Mostrar los equipos en primer segundo y tercer lugar*

Se deberá mostrar el nombre del equipo ganador, el segundo puesto y el tercer puesto del mundial con los puntos totales que obtuvo cada uno en el torneo.

**NOTA:** la victoria no es necesariamente de aquel con más puntos sino de aquel que gana la fase “final”, el tercer puesto es de aquel que gana la fase “tercer puesto”

#### *Buscar equipo por nombre.*

Se le deberá pedir al usuario que ingrese el nombre de un equipo, si el mismo no se encuentra entre los equipos participantes se le avisará que no hubo coincidencias.

Si hay una coincidencia se le mostrará el nombre, grupo al que perteneció, fase hasta la que llegó y título obtenido en caso de obtener alguno (campeón, subcampeón, tercer puesto o cuarto puesto).

#### *Mostrar por fase los países ordenados por puntaje.*

Se deberá mostrar para la fase de grupos por cada grupo el nombre de cada selección ordenadas por el puntaje obtenido (de mayor a menor puntaje). En el resto de las fases no se diferenciará a que grupo perteneció cada selección, se mostrarán todas juntas ordenadas por el puntaje obtenido en dicha fase.

## Salir

Despedirse del usuario y finalizar el programa liberando cualquier memoria que se haya pedido.

## Aclaraciones importantes

- **NO** se puede usar ninguna de las funciones de la biblioteca de strings todo deberá ser desarrollado por ustedes. Salvo los métodos: **getline()** para archivos y **tolower()** a la hora de solicitar datos al usuario.
- Los vectores deben crearse utilizando memoria dinámica. La misma se debe liberar al final del programa. **NO se permite el uso de la biblioteca STD, en particular, <vector>.**
- **NO** se pueden recorrer los archivos mas de **UNA** vez
- El trabajo es de carácter **individual**.
- Se valorará, entre otras cosas, la eficiencia del algoritmo.
- No se deben subir archivos de configuración de los IDEs (solo subir .cpp y .h).
- El trabajo debe compilar con los flags **-Wall -Werror -Wconversion**.

## ¿Qué se evaluará?

- Compilación: sin warnings ni errores.
- Funcionalidad.
- Eficiencia espacial.
- Eficiencia temporal.
- Buenas prácticas de programación (nombres descriptivos, indentación, etc.).
- Separación de operadores.
- Modularización.
- Precondiciones y postcondiciones.
- Manejo de archivos (errores de apertura, correcto procesamiento, cierre)
- Uso de memoria dinámica

## Normas de entrega

Se deberá subir al campus un único archivo comprimido (.zip) en la sección TPs.

Este archivo deberá tener un nombre formado de la siguiente manera:

**Padron\_Apellido\_TP1**

Por ejemplo:

110204\_Perez\_TP1.zip

El archivo deberá contener solo los archivos fuente. Es decir, solo .cpp y .h. **NO** subir los archivos de configuración de sus IDEs. (por ejemplo: CMakeList y cmake-build para Clion, .vscode para VisualStudioCode).

La fecha de **entrega** vence el **miércoles 5/4** a las 23.55hs.

**Puntaje:** 40 Puntos.