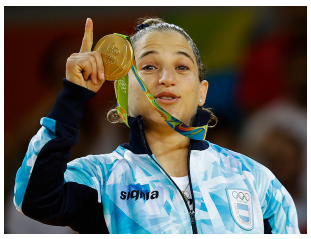
Trabajo Práctico 3: Árboles

Taller de Programación 10 de Septiembre de 2019

1 Presentación del ejercicio

1.1 Motivación y alacance del programa

La reconocida campeona argentina de Judo Paula Pareto se contactó con nosotrxs porque tiene intenciones de organizar torneos de judo amateur. Nos comentó que ella tiene preparados los equipos pero que no sabe como determinar las parejas de forma de minimizar la diferencia de edad.



Nos pide que hagamos un programa que reciba los equipos que ella muy amablemente nos provee y que escriba en un nuevo archivo las parejas armadas. Nos explicó también que como la convocatoria fue muy grande, en algunos equipos hay grandes diferencias etarias, por lo que probablemente se generen parejas muy dispares si queremos minimizar la diferencia de edad. Para solucionar esto, nos dio un listado de condiciones que deben cumplir las parejas.

1. Las parejas entre un menor y un mayor de 18 años no son aceptadas
2. Las parejas entre menores no pueden superar el año de diferencia de edad
3. Las parejas entre mayores no pueden superar los 2 años de diferencia de edad

1.2 Estructura de la información

Con la descripción pensamos que se puede construir la siguiente estructura para representar a cada unx de lxs participantes:

```
typedef struct {
   char *nombre, *apellido;
   int edad;
} _Judoca;
```

```
typedef _Judoca* Judoca;
```

Y la siguiente para las parejas:

```
typedef struct {
   Judoca participante1,participante2;
   int estadoPareja;
} _Pareja;

typedef _Pareja* Pareja;
```

Siendo estadoPareja un entero que indica si la pareja es válida o no pudiendo variar de 0 a 3. Convenimos que 0 representa que la pareja es válida y que cada uno de los números restantes se corresponde con las restricciones pedidas.

Dadas las restricciones que tenemos que cumplir, podríamos utilizar un Árbol Binario de Búsqueda General como el siguiente:

```
typedef struct _TNodo {
   void *dato;
   struct _TNodo *izq, *der;
} TNodo;

typedef TNodo* Arbol;
```

Para poder hacer un uso adecuado de la gestión de la memoria, debemos liberarla, para eso definimos una función:

```
void arbol_destruir ( Arbol arbolN , Destruir d ) ;
```

donde Destruir es:

```
typedef void (* Destruir ) ( void * dato ) ;
```

Esta función recibe un puntero a un dato y, libera la memoria pedida para el dato independientemente de su tipo. Es decir, puntualmente en nuestro caso, d es una función que libera la memoria de un tipo de dato que puede ser una estructura persona o una estructura pareja.

2 Objetivos

El programa deberá tomar un archivo de entrada con la información personal de lxs competidorxs y deberá generar una salida con las parejas armadas minimizando la diferencia etaria. En el caso que se forme una pareja que no cumpla con algunas de las reglas se deberá aclarar detalladamente la razón por la cual no las cumple.

3 Detalles de la implementación

El método main deberá permitir pasar el nombre del archivo de entrada y el nombre del archivo de salida. Es decir sería algo como:

```
./main nombreArchivoEntrada.txt nombreArchivoSalida.txt
```

3.1 Formato de archivo de entrada y salida

El archivo de entrada tendrá el nombre del primer equipo seguido de : y luego listadxs lxs competidorxs con su información personal. Luego vendrá el nombre del segundo equipo seguido de : y a continuación de eso, lxs competidorxs de ese equipo con su información personal. En la salida deberán listarse todas las parejas con sus debidas aclaraciones respecto a las restricciones.

Ejemplo de entrada:

```
Equipo1:
Robert,Mshvidobadze,30
Idalys,Ortiz,29
Amiran,Papinashvili,31
Daria,Bilodid,18
Lukhumi,Chkhvimiani,26
Madeleine,Malonga,25
...
Equipo2:
Clarisse,Agbegnenou,26
Ryuju,Nagayama,23
Tsukasa,Yoshida,23
Sharafuddin,Lutfillaev,28
Uta,Abe,19
Naohisa,Takato,26
...
```

Ejemplo de salida:

```
Daria Bilodid - Uta Abe - Pareja válida
Madeleine Malonga - Ryuju Nagayama - Pareja válida
Lukhumi Chkhvimiani - Tsukasa Yoshida - Pareja válida
Idalys Ortiz - Clarisse Agbegnenou - Pareja de mayores con diferencia mayor a 2 años
Robert Mshvidobadze - Naohisa Takato - Pareja de mayores con diferencia mayor a 2 años
Amiran Papinashvili - Sharafuddin Lutfillaev - Pareja de mayores con diferencia mayor a
2 años
```

4 Evaluación

Se pide que escriba un código que pueda implementar el emparejador con las reglas dadas y las estructuras propuestas. La definición de las estructuras y los prototipos de funciones deben estar declaradas en un archivo $.h$.

Se tomará como criterio de aprobación:

1. funcionamiento del programa;
2. claridad y eficiencia del código escrito;
3. uso de memoria dinámica (pedido y liberación de la misma);
4. modularización.