

Ejercicio 2

Este ejercicio recicla el struct del programa anterior, y utiliza el archivo generado anteriormente.

Los procedimientos leerVector() y leerArchivos() son los mismos del programa anterior. Se le agregó el vector declarado en el main como parámetro a leerArchivos(). De esta forma, al leer el archivo se modifica el vector, que se usará para el resto del programa.

Función main: Lee el archivo, lo guarda en un vector y lo muestra en pantalla. A continuación se pregunta al usuario si desea sortear los grupos. De ser afirmativa la respuesta, ejecuta la función sorteaGrupo(equipos). De lo contrario, cierra el programa.

generarAleatorio(contadorAleatorio) y chequeoAleatorio(aleatorio,contadorAleatorio) Ambas funciones reciben el vector contadorAleatorio como parámetro. A través de la función rand, generarAleatorio crea 8 números y se los carga a contadorAleatorio. A continuación pasa el vector a chequeoAleatorio que a través de un bucle, se asegura de que no haya números repetidos. Además le pasa el valor aleatorio siguiente para que lo comparte con el resto de los valores. Si el valor no se repite, los asigna al salir del while y repite el proceso hasta llenar el vector.

void sorteaGrupo(equipo equipos[]) Este procedimiento recibe el vector de tipo equipo, y procede a realizar el sorteo de la fase de grupos.

```
¿Desea sortear grupos? <S o N>
-
```

Paso 1. En orden de salida, los 8 equipos del bolillero serán cabecera de los grupos A,B,C,D,E,F,G y H respectivamente.

```
Comenzando primer etapa del sorteo...
Primer bolillero sorteado satisfactoriamente. Presione una tecla para continuar..
-
```

Paso 2. En el paso dos, los únicos equipos que pueden llegar a generar conflicto son los pertenecientes a Conmebol. Por lo tanto, al saber la posición de Brasil y Argentina, al caer un equipo en esas posiciones, bastaría con preguntar si su confederación es Conmebol. De ser así, se repite el paso 2.

```
Comenzando segunda etapa del sorteo...
Segundo bolillero sorteado satisfactoriamente. Presione una tecla para continuar..
--
```

Paso 3. Se sortean los números, y utilizando el switch (en cada case) se recorren los equipos conformados con anterioridad para ver si coinciden con el equipo evaluado en confederación. De ser así, evalúa si esa confederación es Uefa, ya que tiene tolerancia de 2 equipos por grupo. En caso de detectar un error, repite el paso.

Paso 4. Idem paso 3.

Luego de ejecutar los 4 pasos, muestra los resultados

```
RUS AFC A
ALE UEFA B
BRA CONMEBOL C
POR UEFA D
ARG CONMEBOL E
BEL UEFA F
POL UEFA G
FRA UEFA H
ESP UEFA F
PER CONMEBOL D
SUI UEFA C
ENG UEFA G
COL CONMEBOL B
MEX CONCACAF E
URU CONMEBOL H
CRO UEFA A
DEN UEFA H
ISL UEFA G
CRC CONCACAF A
SWE UEFA D
TUN CAF C
```

En la fase de sorteo de grupos se utilizan dos funciones: `bool chequeoConfederación(equipo equipos, char grupo, int pos)` que devuelve verdadero o falso, según se encontró que se repite la confederación o no, y `bool chequeoUefa(equipo equipos, char Grupo)` que está anidada a `chequeoConfederacion` cumpliendo la función de verificar, al ser el equipo de confederación UEFA, que en ese grupo no hubiese más de uno.