



ENTREGA 2 VERSIÓN REVISADA

Esta segunda entrega no forma parte del proyecto semestral y no será continuado.

CREACIÓN DE GRUPO

Esta tarea se realizará en grupos de a 2 alumnos. En caso de no tener un grupo, se sugiere buscar o publicar un aviso “busco grupo” en el foro de la página del curso. Tareas posteriores pueden realizarse con un nuevo grupo.

CARRERA DE CAMELLOS

Hace unos días el camellódromo de Chantiago lució su máximo esplendor en el Derby del Gran Camejockey de las 3210 cameyardas. Tristemente, algo salió terriblemente mal y el sistema para determinar los ganadores en las diferentes etapas falló. Los apostadores, indignados, han presentado un masivo reclamo en busca de una solución para poder cobrar sus premios. No es menor, si se considera que el total de premios a repartir se estima cercano a los 37 chilonos de dólares.

Recientemente, tras un intenso trabajo, una ingeniera llamado Lorenza de la Ricca con mucha chispeza dio con una posible solución – pero lamentablemente sus habilidades de programación no le permiten completarla. Aquí es donde entras tú, para ayudarla.

PREMIACIÓN

Antes de continuar, es necesario que sepas algo sobre la carrera y las apuestas del Derby del Gran Camejockey de las 3210 cameyardas.

La carrera consiste en 5 vueltas a la pista y el record actual es de 376,2 segundos (algo más de 6 minutos). Durante las primeras 3 vueltas los camellos reúnen puntos, de acuerdo al lugar que obtienen: 6 puntos para el primer lugar, 4 puntos para el segundo y 1 punto para el tercer lugar; los demás lugares no obtienen puntos. La cuarta y quinta vuelta no entrega puntos.

Existen 3 tipos de apuestas: al resultado de cada vuelta, al puntaje final de cada camello y al resultado final.

En cada vuelta, las apuestas se pagan típicamente: 1,2 veces lo apostado al primer lugar, 1,4 veces lo apostado al segundo lugar y 1,1 veces al tercero.

Al puntaje final, se paga 2 veces un acierto y el pago aumenta a más aciertos hasta 10 veces por acertar todos los puntajes (el incremento varía según el número de camellos en competición). Nótese que todas estas apuestas están limitadas a un máximo de 10.000 dólares por apostador.

Por último, el premio final, el centro del evento, consiste en un pozo a repartir entre los apostadores, consistente en un mínimo de 10 veces el valor apostado (garantizado), que puede acender hasta 15 veces el valor apostado (variable según el número de ganadores del premio máximo). Las apuestas al gran final tienen un máximo de 0,2 chilonos de dólares.

LA SOLUCIÓN “RICCA”

Como se mencionó anteriormente, el sistema automático para determinar el resultado falló. Sin embargo, nuestra querida Srta. Ricca descubrió que todos los camellos son monitorizados con bastante precisión, regulando sus signos vitales y reuniendo múltiples datos estadísticos – entre los que se encuentra cuantas cameyardas avanzó cada camello cada segundo. Con esta información, sumando los avances cada vez, se puede saber aproximadamente cuánto tiempo tomó cada vuelta y el total de la carrera.

El único problema con este sistema es que la medición será en segundos (lo que es muy impreciso en esta clase de eventos), por lo que habrá empates debido sólo a la falta de precisión. Además, debido a errores de medición en los avances, los resultados no serán del todo confiables ni comparables (comparados con los métodos convencionales de medición de tiempo).

De momento, la CAMEL (Camel Affiliation Measurer for Events and Leagues) ha anunciado que esta medición no será válida para efectos de records ni para fines estadísticos de cada camello. Por su parte, el Camellódromo de Chantiago, entendiendo su responsabilidad en el asunto ha aceptado esta medición para efectos de pago de apuestas, haciéndose responsable de los casos en que la precisión de la medición no es suficiente para determinar el resultado exacto. El acuerdo ha sido el siguiente:

(Vueltas 1 a 3)

Los primero, segundo y tercer lugar estará dado, como siempre, por el tiempo. Si dos camellos completan la vuelta en el mismo segundo que otro, compartirán el lugar (pero el siguiente lugar, obviamente, quedará vacante). Por ejemplo, si cinco camellos terminan la primera vuelta en 81, 81, 82, 83 y 84 segundos, los lugares serán 1, 1, 3, 4 y 5. Obviamente pueden haber resultados extraños que premien a más de 3 camellos cada vuelta, como que los cinco terminen primero, o que hayan dos primeros y tres terceros, un primero y tres segundos, etc.

Los premios al lugar parcial se pagarán como siempre, independiente de las repeticiones de los lugares.

Los puntajes aplicarán como siempre, para los lugares 1, 2 y 3 de cada vuelta.

El resultado final de la carrera, donde sólo se premia al primer lugar, se mantendrá como siempre. En caso de empate, el premio final “mínimo garantizado” se mantendrá, pero el máximo se repartirá según el número de ganadores.

PROGRAMACIÓN

Tu objetivo en esta tarea será leer un archivo con los datos de los camellos y entregar un resultado en pantalla con:

- Camello(s) en primer, segundo y tercer lugar de cada vuelta
- Puntaje obtenidos por todos los camellos
- Camello(s) ganador(es) de la carrera.

El archivo con los datos responderá al siguiente formato:

- Una línea indicando el número de camellos que compitió en la carrera
- Una línea indicando los nombres de cada camello y un código único de cada uno, bajo el siguiente formato:
 - Cada camello-código estará separado con un ";" (punto y coma) del siguiente.
 - En cada caso, primero vendrá el nombre del camello, seguido de una coma y luego el código del camello.
 - Entre los separadores podrían haber espacios, los que deben ser ignorados.

Por ejemplo, `<<Camello bkn1, BKN1 ; Otro camello, OC >>` posee los camellos "Camello bkn1" de código "BKN1" y "Otro camello" de código "OC" (nótese los espacios ignorados).

- Una o más líneas, hasta llegar al final del archivo, de códigos de camellos y avances en cada segundo. Por como se obtuvieron los datos, estos están un tanto revueltos, pero se garantiza que, para un camello dado, vienen sus datos del primer segundo, segundo segundo, etc. Los datos vienen en pares (código, [cameyardas](#)), separados por coma; las [distancias](#) usan un "." para separar los decimales. Por ejemplo, estas líneas serían válidas para 3 segundos de carrera de 5 camellos:

```
TF,8.23,TF,8.61,ARFC,8.64,CAM,8.6,FAST,8.17,  
TWC,8.09,TWC,7.71,TWC,7.74,ARFC,7.77,CAM,8.01,FAST,7.87,  
FAST,8,CAM,8.25,  
ARFC,8.3,TF,8.29
```

Con estos datos, el Camello TF avanzó: 8.23 cameyardas, luego 8.61 y luego 8.29, es decir, tras estos 3 segundos, avanzó 25,13 cameyardas. Por su parte, ARFC avanzó 8.64, 7.77 y 8.3, es decir, en 3 segundos avanzó 24,71 cameyardas. Si hubiera una meta en 25 cameyardas, TF la habría cruzado luego de 3 segundos, pero ARFC aún no.

RUBY

Esta tarea debes realizarla en Ruby versión 2.2 o superior.

Para esta tarea debes programar estrictamente orientado a objetos, aunque obviamente tu programa correrá a partir de un script que debe llamarse desde consola. Por ejemplo:

```
>ruby programa.rb datos.txt
```

PLAZO Y FORMA DE ENTREGA

El plazo máximo de entrega es el miércoles 23 de marzo, a las **23:50 hrs.** La forma de entrega consistirá en subir al sitio del curso un zip con la aplicación en un buzón especialmente creado y las instrucciones para correr la aplicación. Se recomienda incluir información del sistema operativo donde se probó la aplicación, así como la versión de Ruby utilizada y cómo se ejecuta.

AYUDANTÍA DE AYUDA

El día viernes 18 de marzo habrá una ayudantía para la resolución de dudas y apoyo de trabajo, en el módulo 5 (15:30 a 16:50) en sala por confirmar.

EVALUACIÓN

La evaluación de esta segunda entrega responderá a la siguiente rúbrica simple:

Pts	Dimensión	Logrado (pts x 1)	Parcialmente logrado (pts x 0.5)	No logrado (0 pts)
1	Complejidad de la solución	El programa lee el archivo, lo procesa y entrega un resultado. Falla si no se entrega el archivo o si el archivo posee otro formato. No importa si el resultado es correcto.	O bien el programa no lee el archivo, o no procesa los datos del archivo (y procesa otros datos) o no entrega un resultado (hace 2 de tres).	El programa no hace dos de los tres puntos anteriores.
3	Correctitud de la solución	El programa entrega un resultado correcto con los archivos de prueba publicados y con otros archivos de prueba (por ejemplo, de caso borde).	El programa sólo funciona bien con los archivos publicados, pero falla con otros archivos de prueba.	El programa no funciona bien con los archivos de prueba publicados.
2	Orientación a objetos*	El programa está compuesto por un modelo de múltiples clases, donde cada clase tiene una identidad propia clara o cumple un propósito claro.	Hay un modelo de clases, pero posee errores evidentes, como que hay hasta 3 clases que "confunden" sus roles, o sirven a varios propósitos (que debieron dividirse en otras clases).	O bien sólo hay unas pocas clases "que hacen un poco de todo" o hay más de 4 clases que poseen claros errores.

* Este punto en la rúbrica posee un criterio abierto a interpretación. Este curso o esta tarea no buscan dar una evaluación precisa sobre "calidad de programación orientada a objetos". En su lugar, se busca incentivar un diseño orientado a objetos; por esta razón, aunque la rúbrica en sí es poco objetiva, lo "normal" será considerar este punto como logrado, salvo casos evidentes de errores, que producirán un resultado de parcialmente logrado o, en casos extremos, un no logrado (por ejemplo, sólo un gran scripting sin clases, o un modelo de sólo dos clases: "TheClassThatDoesEverything" – la clase que lo hace todo - y "TheClassThatIsEverything" – la clase que es todo).