

FUNDAMENTOS DE COMPUTACIÓN Y PROGRAMACIÓN

Enunciado Control de Laboratorio N°3

I. Objetivos

Aplicar los conocimientos de Programación vistos en Cátedra y Laboratorio para generar un programa en Python que realice operaciones sobre estructuras de datos.

II. Problema

El Kino es un juego de azar chileno, el cuál consta de combinaciones de 14 números, escogiendo números desde el 1 al 25 para cada cartón. Cada semana se realiza un sorteo, en el cuál se reparte una cantidad de dinero determinada entre los cartones jugados para ese sorteo que hayan tenido 14, 13, 12, 11 y 10 aciertos, siendo mayor el monto del premio a un mayor número de aciertos.

Se requiere una aplicación en Python que permita simular un sorteo del Kino, a partir de dos archivos de entrada que se describen a continuación:

Datos del Sorteo: El archivo en cuestión maneja los datos relativos al sorteo, en este se tiene el pozo a repartir para cada número de aciertos, separando los campos: número de aciertos y monto del premio separados por un espacio. El archivo está nombrado con la palabra sorteo y el número de sorteo separados por un guión (Sorteo-7531.txt) y la extensión de este es un archivo de texto plano (.txt), cómo se presenta en el ejemplo:

Sorteo-2345.txt	
14	387021411
13	5729976
12	4570000
11	6940500
10	14331000

Cartones jugados: El segundo archivo que la simulación requiere es el conjunto de cartones jugados, el cual contiene el conjunto de cartones que se han jugado para un sorteo en cuestión. El archivo contiene en cada línea un cartón jugando, indicando el identificador del cartón y luego, separado por dos puntos y un espacio (:) los números jugados, separados por comas.

El archivo se nombra con la palabra Cartones y el número de sorteo separados por un guión y al igual que el archivo de sorteos, la extensión de este es un archivo de texto plano (.txt).

Cartones-2345.txt	
4030301959391:	1, 2, 3, 5, 6, 7, 8, 9, 13, 15, 16, 17, 21, 23
4441972959391:	2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 18, 19, 20, 21
4481251959391:	1, 2, 4, 6, 7, 9, 11, 12, 13, 15, 16, 17, 21, 25
...	

A partir de estos archivos de entrada, se requiere que el programa solicite por pantalla al usuario **el número del sorteo**, genere aleatoriamente los 14 números premiados (Usando el módulo random de Python o algún otro similar) y genere dos archivos de salida (considerando las operaciones que se requieren para entregar dicha salida) que se describen a continuación:

Resumen del Sorteo: El cuál presenta el resumen del sorteo, indicando el número del sorteo, la cantidad de cartones jugados para el sorteo en cuestión y los números sorteados. Además del resumen de la premiación por cada categoría, indicado para cada una:

- La categoría, el total del monto a repartir, la cantidad de ganadores por categoría, el monto que recibirá cada ganador por categoría.
 - Considere que los montos de dinero deben presentarse como se indica en el ejemplo (Es decir, con separación de miles y el símbolo \$ antecediéndolo), para ello puede usar las etiquetas de formato que se indican en la clase 12.
 - Considere que los números de jugadores premiados y totales también deben presentar separadores de miles a fin de que un usuario pueda comprender rápidamente el archivo de salida.

A continuación se muestra cómo debiera presentarse el archivo de salida, el cuál debiera crearse con el nombre “Resumen_del_Sorteo-<N° de Sorteo>.txt” a fin de impedir que cada sorteo borre el sorteo anterior.

Resumen_del_Sorteo-2345.txt				
=====				
SORTEO N°2345				
SE JUGARON 102.435 CARTONES				
=====				
RESULTADOS:				
1 2 4 5 6 8 12 14 17 20 21 22 23 24				
=====				
CATEGORÍA	TOTAL CATEGORÍA	N° DE GANADORES	PREMIO POR GANADOR	
14 Aciertos	\$387.021.411	1	\$387.021.411	
13 Aciertos	\$5.729.976	10	\$572.997	
12 Aciertos	\$4.570.000	450	\$10.155	
11 Aciertos	\$6.940.500	4.601	\$1.508	
10 Aciertos	\$14.331.000	25.305	\$566	

Cartones Premiados: Conjunto de archivos que presenta el detalle de los cartones premiados por sorteo indicando la categoría del premio y el monto del premio que dicho cartón recibirá. Además en cada fila el número del cartón y los números jugados por dicho cartón.

- Considere que los datos se presentan separados por categoría (De mayor cantidad de aciertos, a menor cantidad de aciertos) y luego, cada categoría presenta ordenadas sus líneas por número de cartón de menor a mayor.
- Considere que una persona con un premio de 14 aciertos, si bien tiene también correctamente 13, 12, 11 y 10 aciertos, sólo cobra en la categoría más alta, por lo que su número de cartón sólo debiese aparecer en la categoría de 14 aciertos. La misma situación aplica para todas las otras categorías, dónde sólo se paga el premio de la mayor cantidad de aciertos y no se considera para el resto de ellas.

A continuación se muestra un ejemplo del archivo solicitado, el cuál debiera crearse con el nombre "Cartones_Premiados-<N° de Sorteo>.txt" a fin de impedir coincidencias con archivos de sorteos futuros y pasados.

Cartones_Premiados-2345.txt

```
=====
14 ACIERTOS, PREMIO DE $387.021.411
=====
```

```
4030301959392: 1 2 4 5 6 8 12 14 17 20 21 22 23 24
=====
```

```
13 ACIERTOS, PREMIO DE $572.997
=====
```

```
4030301959211: 1 2 3 5 6 8 12 14 17 20 21 22 23 24
```

```
4030301959316: 1 2 4 5 6 8 11 14 17 20 21 22 23 24
```

```
4030301959989: 1 2 4 5 6 8 9 14 17 20 21 22 23 24
```

```
...
```

II.1. Requisitos previos

El trabajo solicitado debe desarrollarse en Python, en la versión utilizada en el curso, es decir Python 2.6.

II.2. Respecto a la representación en Python

Considere que, descontando las funciones que generan números aleatoriamente (Las cuales están en las experiencias de laboratorio de esta semana este trabajo fue conscientemente diseñado para ser resuelto utilizando únicamente los contenidos vistos hasta el momento, por lo que no es necesario investigar o añadir otras librerías además de las señaladas anteriormente para ejecutar el código.

Considere que se evaluará la estructura de programas en Python, por lo que a pesar de la envergadura del programa este **DEBE** estar subdividido en funciones claramente definidas, comentadas, indicando sus entradas, salidas y lo que éstas entregan, y de ser necesario ir comentando paso a paso las instrucciones.

Los bloques de programa deben ir claramente definidos, y como mínimo debiera diferenciar:

- Encabezado: Indicado la información relevante del programa.
- Definición de constantes: Indicando la importación y definición de valores constantes en el programa.
- Definición de funciones: Indicando las funciones necesarias para la ejecución del programa, además, se recomienda dividir dichas funciones de acuerdo a:
 - Funciones de lectura: Aquellas encargadas de leer los archivos de entrada.
 - Funciones de escritura: Aquellas encargadas de escribir en los archivos de salida y formatear datos y estructuras de datos para ser escritos en los archivos en cuestión.
 - Funciones de procesamiento: Aquellas que realizan el cálculo de ganadores, números de sorteo y otros cálculos orientados a las tareas que no son ni de lectura, ni de escritura.
- Bloque Principal
 - Entrada: Solicitando la entrada por teclado y realizando las operaciones necesarias de lectura de archivos.
 - Procesamiento: Realizando las funciones necesarias para obtener los datos necesarios para generar las salidas necesarias.
 - Salida: Escribiendo en los archivos de texto y notificando que éstos fueron creados con éxito.

Considere que identificando y dividiendo el programa en sub-procesos, el problema resultará más sencillo de abordar para el equipo de trabajo, en caso de alcanzar sólo parcialmente los objetivos del control, se sugiere colocar en los comentarios las abstracciones de datos y procesos implementados y no implementados.

III. Entrega

El plazo máximo para entregar el programa será el día sábado 14 de Junio a más tardar a las 13:00 horas.

Se requiere entregar un archivo .py con el programa desarrollado en Python, con los RUT de sus integrantes (sin puntos, ni guión) por nombre de archivo siguiendo el formato <RUT N°1>-<RUT N°2>.py por ejemplo:

173454322-18984321K.py

Además en el código, se debe añadir el siguiente encabezado del programa, con los siguientes datos para identificar su trabajo (Rellene el encabezado del programa con los datos en formato IDÉNTICO al indicado en el ejemplo a continuación):

```
# -*- coding: cp1252 -*-

# SECCIÓN DE LABORATORIO: L-1
# PROFESOR DE LABORATORIO: JUAN PADILLA

# INTEGRANTES
# 1.
# NOMBRE: Juan Carlos Perez Gonzalez
# RUT: 17.345.432-2
# CARRERA: Ingeniería Civil Mecánica
# PROFESOR DE CÁTEDRA: Andrés Rice

# 2.
# NOMBRE: Javier Rojas Madariaga
# RUT: 18.984.321-K
# CARRERA: Ingeniería de Ejecución en Computación e Informática
# PROFESOR DE CÁTEDRA: César Silva

# DESCRIPCIÓN DEL PROGRAMA ...<CONTINÚE CON EL PROGRAMA A PARTIR DE AQUÍ>
```

Se requiere que suba este archivo a la plataforma USACHVIRTUAL y, como respaldo, enviar una copia al mail de su profesor de laboratorio y de su ayudante de laboratorio **ANTES** del plazo establecido.

En caso de tener más archivos en su implementación, considere que el encabezado debe ser parte de cada uno de los archivos y se debe entregar el trabajo completo en un archivo comprimido de extensión “.zip” o “.rar” con el mismo nombre que se indicó anteriormente. Considere que archivos sin identificador de autores tanto en el nombre como dentro del encabezado del código, **no serán revisados, ni evaluados.**

Del mismo modo considere que archivos enviados únicamente al correo del profesor, sin copia en UsachVirtual, no serán considerados en la revisión.

IV. Evaluación

Respecto a la evaluación tenga en consideración las siguientes instrucciones (El no seguimiento de una o más de estas normas puede significarle ser evaluado con la nota mínima):

- Los grupos estarán compuestos por un máximo de **dos personas**, ambas pertenecientes a la **misma sección de laboratorio**.
- Se permite el trabajo individual, sin embargo, esto no significa una disminución en los criterios de evaluación.
- En caso de detectar **programas iguales, o con suficientes similitudes entre sí**, tanto en una misma sección como en secciones distintas, será considerado un acto de copia y calificado con nota mínima. Se compararán programas con un algoritmo de reconocimiento de texto y además se pasará por un proceso de validación visual por personal de la coordinación de Fundamentos de Computación y Programación para realizar dicha detección.
- Grupos que entreguen fuera del plazo establecido serán calificados con la nota mínima.
- Grupos que sean descubiertos en actos deshonestos serán sancionados con la nota mínima.
- Grupos que revelen aspectos claves de su implementación en UsachVirtual u otras plataformas (Incluyendo redes sociales) serán calificados con nota mínima.
- Considere que la implementación de su programa es de responsabilidad única de los integrantes del grupo, y que tanto en cátedra como en laboratorio se han entregado todas las herramientas y los contenidos para el desarrollo de ésta, por lo que **cualquier indicio de intervención de un tercero**, será sancionado con la nota mínima.
- La difusión de este enunciado, solicitando ayuda, ofreciendo dinero a cambio de la resolución e incluso la publicación de éste dentro del plazo de desarrollo de la tarea, se considera un acto deshonesto, que daña la imagen de la Facultad de Ingeniería y de la Universidad de Santiago y será calificado inapelablemente con nota mínima.
- Se sancionará con descuentos sobre la calificación final a los grupos que no cumplan con las instrucciones del programa y de entrega publicados en el enunciado.
- Se bonificará a los grupos que añadan funcionalidades adicionales, aspectos de presentación más elaborados y que en general no se limiten simplemente a resolver el

problema dado (en estos casos indicar en los comentarios o en un archivo .txt, requisitos, modo de uso, funcionalidades extras, etc).

- Considere que se evaluará el código mismo, las funcionalidades conseguidas, la estructura del programa, la ejecución del programa, la respuesta a eventos inesperados y los comentarios en el código fuente.
- Cualquier situación no contemplada en este documento será dirimida por la Facultad de Ingeniería.

La nota final de esta evaluación vale como una nota del total de notas de laboratorio, y esta se divide en dos partes, la implementación y el test de salida con las siguientes características:

- El test de salida, corresponde a un control que será tomado en la siguiente semana luego de la entrega del trabajo, en horario de laboratorio y en conjunto con la evaluación N°4 de Laboratorio, a fin de verificar que el grupo que presenta un trabajo que es de su autoría, en el cuál se le preguntará respecto a aspectos funcionales de **SU** implementación o la resolución sin el intérprete de Python de alguno de los sub-problemas del enunciado.
 - Si su nota del test es superior a 4,0, la ponderación es 10% test de salida y 90% programa.
 - Si su nota en el test de salida es inferior a 4,0, la ponderación del trabajo es 100% la nota más baja entre el control de salida y la nota del programa.