

Algoritmos y Programación I (75.40)

Trabajo práctico n.º 1

Primer cuatrimestre 2015

Introducción

En la Universidad de Alfardería de la ciudad de Buenos Aires el querido docente Toban Lajp del curso de “Logaritmos I” ha desaparecido sin dejar rastros.

Aunque hay pruebas fehacientes de que se haya olvidado el cargador de su celular y se fuera de vacaciones sin avisar a una pomposa isla en el caribe, la idea que recorre los pasillos de Paseo Colón es más sombría. Los rumores cuentan que alguno de sus allegados se ha encargado de él para siempre. Una comitiva decidió que los sospechosos a investigar son: - Coronel D. Bárbara: Al haber desaparecido Toban Lajp, es quien queda a cargo de su curso de “Logaritmos I”. Es ciertamente el más beneficiado. - Christian Grace: el extravagante joven millonario que tiene locas a las alumnas del curso de “Logaritmos I” (del cual también es ayudante), pero que siempre se vio a la sombra del profesor Lajp. - Haskell Martinez: siendo que se encuentra en estado de querer ser docente del curso, la desaparición de uno de los docentes puede catapultarlo hasta la cima. - Ing. Alan Información: el nuevo ayudante extranjero que fue recibido con fiesta y panqueques, pero parecería querer escalar rápidamente hasta hacerse cargo del curso, con el fin de enseñarnos a usar “ficheros” y no “archivos”. - Yisus S. : El gurú del grupo, y quien lo provee de vino y asados. Nadie desconfiaría de él, pero nunca se sabe. . . - Sra Rosada:

Se decidió que se descubrirá el culpable a partir del juego Clue¹, pero nadie

¹ <http://en.wikipedia.org/wiki/Cluedo#Games>

cuenta con el famoso juego de mesa, ni tiene dinero para poder compararlo en el corto plazo. Por lo tanto, se le pidió a un grupo muy prestigioso de programadores que realicen el diseño y programación del juego. Dicho grupo terminó de realizar todo el diseño, y de implementar las interfaces y algunas partes del juego, pero al conocer a algunos personajes de la Universidad de Alfarería se desató una discusión que atentaba con una segunda desaparición misteriosa, por lo que será necesario pedirle a alumnos con menor experiencia que terminen de realizar el programa. Por suerte, el código que se pudo recuperar está completamente documentado, inclusive las partes faltantes, por lo que continuarlo no debiera ser una tarea tan laboriosa.

Consigna

Se pide implementar un juego interactivo de sopa de letras.

El juego se realiza en un tablero cuadrado de $N \times N$ y consta de **tres fases** que se detallan a continuación. Las fases I y III son interactivas (solicitan datos al usuario). La fase II, en cambio, es enteramente interna al programa.

Fase I: configuración del juego

A su comienzo, el programa imprime un mensaje de bienvenida y solicita al usuario los siguientes **parámetros**:

- tamaño (en filas) del tablero: un entero N ;
- número de palabras en la partida: un entero P ;
- palabras a colocar en el tablero: P cadenas de texto.

Se debe **validar** los valores recibidos de la siguiente manera:

- el tamaño mínimo del tablero es 10×10 , y el máximo 20×20 ;
- el número de palabras en la partida no puede exceder $N/2$;

- la longitud de cada palabra debe estar entre 3 y $N/2$, y solo pueden estar formadas por letras de la a a la z .
- no se admiten palabras repetidas.

Si no se cumple alguno de estos criterios, se solicitará de nuevo el valor al usuario.

Nota: se deberá solicitar P o N de nuevo si el usuario ingresó un valor no numérico. En cuanto a errores en una palabra p , se solicitará de nuevo esa palabra, pero no las anteriores.

Por último, se debe **transformar** cada palabra de la siguiente manera:

- cualquier carácter en mayúscula debe convertirse a minúscula.

Fase II: generación del tablero

Esta fase es interna al programa, y no se muestra nada por pantalla.

El programa coloca las P palabras de la partida en una **estructura de $N \times N$ celdas**, y rellena las celdas restantes con caracteres alfabéticos aleatorios.

La representación interna de esta estructura es a elección del alumno. Dos posibles ejemplos: una lista de N listas, una lista de N cadenas.

Las palabras pueden ser colocadas en dirección horizontal o vertical, según los siguientes **requerimientos**:

- para colocar correctamente una palabra, todas sus letras deben aparecer de manera contigua en la misma fila (orientación horizontal) o columna (orientación vertical);
- la primera palabra siempre debe ser colocada en vertical;
- la segunda palabra (si la hay) siempre debe ser colocada en orden inverso en horizontal (es decir, la última letra a la izquierda y la primera a la derecha);

- el resto de palabras pueden ser colocadas en horizontal o vertical, orden inverso o no, a elección del alumno.²

Consejo de implementación: escribir una programa inicial que simplemente coloque cada palabra al comienzo de una fila distinta, y comenzar a implementar la fase III inmediatamente después. Una vez la fase III sea funcional, volver a la lista de requerimientos de la fase II, y dedicar a su desarrollo el resto del tiempo hasta la fecha de entrega.

Para generar caracteres alfabéticos aleatorios (en minúscula), se puede utilizar la siguiente función:

```
import random
import string

def letra_aleatoria():
    """Devuelve al azar una letra minúscula.
    """
    return random.choice(string.ascii_lowercase)
```

Fase III: juego interactivo

Esta fase consiste en un ciclo en el que se imprime el estado actual del tablero y se solicita al usuario que encuentre la siguiente palabra; el ciclo se repite hasta que el usuario encuentre todas las palabras.

Para **mostrar el tablero**:

- las letras aparecen en una cuadrícula $N \times N$, separadas por espacios;
- las filas se etiquetan con números enteros de 1 a N , y las columnas con letras A, B, C...

² Es decir, a efectos de estos requerimientos, es válido colocar todas las palabras a partir de la 3.^a en horizontal de izquierda a derecha, o todas en vertical en orden inverso, o cualquier otra combinación intermedia.

- las palabras ya encontradas por el usuario aparecen en mayúscula; el resto de letras aparecen en minúscula.

Ejemplo:

	A	B	C	D	E	F
1	z	E	d	o	i	y
2	m	H	d	e	c	z
3	f	C	o	c	o	l
4	d	O	L	a	r	q
5	p	C	u	r	t	x
6	k	e	i	c	w	w

Tras mostrar el tablero, se mostrará al usuario cuántas palabras quedan, y se le solicitará **un par de celdas inicio-fin**. Por ejemplo, dado el tablero anterior:

Introduzca dos celdas (palabras restantes: 2): 4A 4E
Se encontró: "DOLAR"

	A	B	C	D	E	F
1	z	E	d	o	i	y
2	m	H	d	e	c	z
3	f	C	o	c	o	l
4	D	O	L	A	R	q
5	p	C	u	r	t	x
6	k	e	i	c	w	w

Introduzca dos celdas (palabras restantes: 1): 3F 3C
Se encontró: "LOCO"

Encontró todas las palabras: ¡enhorabuena!

En cada iteración del ciclo se debe manejar las siguientes **condiciones de error**:

- no se proporcionaron dos coordenadas, o exceden las dimensiones del tablero;

- las celdas ingresadas no están en la misma fila o columna;
- las celdas ingresadas no forman una palabra de las ingresada en la fase I. (Nota: en el caso de palabras en orden inverso, las celdas se han de ingresar en orden inverso también; véase el ejemplo anterior 3F/3C).
- la palabra ya había sido encontrada con anterioridad.

Criterios de aprobación

A continuación se describen los criterios y lineamientos que deben respetarse en el desarrollo del trabajo.

Informe

El informe debe consistir en una descripción del **diseño** del programa.

Debe recordarse que la etapa de diseño es *anterior a la implementación*, por lo tanto debe describirse, utilizando texto y/o diagramas, cómo se va a estructurar el código para cumplir con las especificaciones de la consigna.

Algunas preguntas que deberían responderse:

- A grandes rasgos, ¿cómo será el flujo del programa?
- ¿Cómo se va a guardar en memoria el estado del juego?
- ¿Qué operaciones se efectuarán durante el juego?

Código

Además de satisfacer las especificaciones de la consigna, el código entregado debe cumplir los siguientes requerimientos:

- El código debe ser claro y legible.
- El código debe estructurarse en funciones y, cuando corresponda, módulos. Las funciones deben definirse de la manera más genérica posible.
- Todas las funciones deben estar adecuadamente documentadas, y donde sea necesario el código debe estar acompañado de comentarios.

Entrega

La entrega del trabajo consiste en:

- El informe y código fuente impresos. Para el código fuente utilizar una tipografía `monoespacio`.
- El informe en formato *PDF*.
- Una versión digital de todos archivos `.py` de código, separados del informe. En el caso de ser más de un archivo, comprimidos en un `.zip`.

El informe impreso debe entregarse en clase. Los dos últimos (PDF y código fuente) deben enviarse a la dirección electrónica `tps.7540rw@gmail.com` con el asunto “*TP1 - PADRÓN*”.

Este trabajo práctico se desarrolla en forma **individual**. El plazo de entrega vence el **viernes 17 de abril de 2015**.