

## ¿Fácil De Decir?

Un password seguro es algo delicado. Los usuarios prefieren passwords que sean fáciles de recordar (como *amigo*), pero este password puede ser inseguro. Algunos lugares usan un generador aleatorio de passwords (como *xvtpzyo*), pero los usuarios toman demasiado tiempo recordándolos y algunas veces lo escriben en una nota pegada en su computador. Una solución potencial es generar password “pronunciables” que sean relativamente seguros pero fáciles de recordar.

Ganfer está desarrollando un generador de passwords. Su trabajo en el departamento de control de calidad es probar el generador y asegurarse de que los passwords sean aceptables. Para ser aceptable, el password debe satisfacer estas tres reglas:

- 1. Debe contener al menos una vocal.**
- 2. No debe tener tres vocales consecutivas o tres consonantes consecutivas.**
- 3. No debe tener dos ocurrencias consecutivas de la misma letra, excepto por ‘ee’ u ‘oo’.**

Note que Estas reglas no son perfectas; habrán muchas palabras comunes/pronunciables que no son aceptables.

La entrada consiste en una o más potenciales passwords, uno por línea, seguidas por una línea conteniendo una palabra 'end' que señala el fin de la entrada. Cada password tiene como mínimo una y como máximo veinte letras de largo y está formado por solo letras en minúscula. Por cada password, despliegue OK si es aceptable, de lo contrario NO, usando el formato mostrado en el ejemplo de salida.

Ejemplo de entrada (in.txt)	Ejemplo de salida (out.txt)
a	OK
tv	NO
ptoui	NO
bontres	NO
zoggax	NO
wiing	NO
eep	OK
houctuh	OK
end	

¿Qué

viene

después?

De acuerdo a Wikipedia, una progresión aritmética (PA) es una secuencia de números tales que la diferencia de cualquiera de los dos miembros sucesivos de la secuencia es una constante. Por ejemplo:

3, 5, 7, 9, 11, 13, . . . es una progresión aritmética con diferencia común 2.

Para este problema, nos limitaremos a la progresión aritmética cuya diferencia común es un número entero distinto de cero.

Por otro lado, una progresión geométrica (PG) es una secuencia de números, donde cada término después de la primera se encuentra multiplicando el anterior por un número distinto de cero y común. Por ejemplo:

2, 6, 18, 54, . . . es una progresión geométrica con razón común 3

Para este problema, nos limitaremos a la progresión geométrica cuya razón común es un número entero distinto de cero.

Dadas tres miembros sucesivos de una secuencia, es necesario determinar el tipo de la progresión y el siguiente miembro de la secuencia.

### **Formato de entrada.**

El programa se pondrá a prueba en uno o más casos de prueba. Cada caso se especifica en una sola línea con tres enteros ( $-10,000 < a_1, a_2, a_3 < 10,000$ ), donde  $a_1, a_2$  y  $a_3$  son distintos. El último caso es seguido por una línea con tres ceros. (in.txt)

### **Formato de salida**

Para cada caso de prueba, el programa debe imprimir una sola línea de la forma:

XX(espacio)V

Donde xx es cualquier PA o PG dependiendo si la progresión dada es una operación aritmética o geométrica y V es el siguiente miembro de la secuencia dada. Todos los casos de entrada se garantiza que sea ya sea una operación aritmética o progresiones geométricas. (out.txt)

Ejemplos:

next.in	OUTPUT
4 7 10	AP 13
2 6 18	GP 54
0 0 0	

## Importe exacto

Un administrador de sueldos se encarga de realizar los pagos diariamente a sus trabajadores con el importe exacto utilizando monedas de diferentes denominaciones, las denominaciones pueden variar dependiendo de la disponibilidad del banco.

Debido al incremento en el número de trabajadores, se requiere un programa que le diga al administrador de pagos la cantidad de monedas de cada denominación que debe solicitar al banco para pagar a cada uno de sus empleados con el importe exacto a cada uno.

La entrada consiste en un archivo de texto (in.txt) donde la primera línea corresponde a una secuencia de números correspondientes a las denominaciones disponibles en el banco, separadas cada una por un espacio en blanco.

A continuación se enlistan los salarios de cada uno de los trabajadores en líneas diferentes.

Se espera como salida un archivo de texto (out.txt) el cual ponga por línea las denominaciones en orden descendente seguidas de un espacio y la cantidad de monedas necesarias para realizar los pagos.

Ejemplos:

### Entrada (in.txt)

0.5 1 2 5 10 20 50

63.50

48.50

74

112.50

### Salida (out.txt)

50 4

20 3

10 2

5 1

2 5

1 2

0.5 3

## Lista de tareas

Se requiere un administrador de listas de tareas desarrollado con C# y con un frontend Web, la lista de tareas debe considerar la inserción de nuevas tareas y el marcado como completadas. Puede utilizar persistencia en memoria o en base de datos.

Las tecnologías aceptadas son:

- Microsoft .NET Framework 4.5, 4.0, 3.5
- C#
- Cualquier manejador de base de datos (si lo requiere)
- WCF, Web Api, MVC
- AngularJS 1.x
- Bootstrap
- HTML 5
- CSS

Se evaluará la estructura de la aplicación y el apego a buenas prácticas de programación.