

Ayudantía 3 IWI-131

Bryan Salas, Cristian Navarrete, Daniel Tapia

UTFSM

15/04/2019

Ejercicio 1

Las fuerzas armadas del país realizan una competencia interna para ver que rama tiene los hombres **más rudos**. Las ramas de las fuerzas armadas son: Ejército (**E**), Armada (**A**) y Aviación (**V**).

La cantidad de competencias y competidores varía año a año, sin embargo la forma de determinar quien gana no varía. La rama ganadora es la que mayor cantidad de participantes tenga entre los **primeros n participantes**, donde:

$$n = \text{cant_participantes_competencia} / 3.$$

Se les pide crear un programa que pregunte cuantas competencias se harán. Luego para cada competencia pregunte el resultado, el cual corresponde a un **texto** que identifica el orden en que llegaron los distintos participantes de las ramas. Finalmente, cuando ya se registraron los resultados de todas las competencias, el programa debe indicar la rama ganadora, es decir, aquella que más competidores tuvo a nivel general entre las primeros n posiciones, considerando todas las competencias. En caso de empate, mostrar cualquiera de las ramas ganadoras.

A continuación se muestra un ejemplo de cómo debiera verse la ejecución del programa

Ejercicio 1

Numero de competencias? **3**

Resultados Competencia 1: **AAAAEEEVVAAVAAE**

Resultados Competencia 2: **EEAVVVEAAAAAAAAA**

Resultados Competencia 3: **EEEVVVAADVVEAA**

Gano Ejercito

Numero de competencias? **4**

Resultados Competencia 1: **AAAAA**

Resultados Competencia 2: **VAAVEEEEA**

Resultados Competencia 3: **VVAEEEEE**

Resultados Competencia 4: **AEVAVV**

Gano Armada

Ejercicio 2

Dos grandes empresarios, al ver que sus amigos fueron descubiertos por colusión, decidieron crear y utilizar un medio de comunicación más privado/seguro. Crearon un sistema de comunicación encriptado (y no simples partes de matrimonio) para poder mandarse mensajes sobre este asunto tan delicado y así no ser descubiertos. Debido a que los mensajes son muchos, necesitan que los mensajes sean descryptados de manera automática. Entonces, se necesita:

Ejercicio 2.1

- a) Crear una función `codigo_palabra(codigo)` que reciba un `codigo` encriptado de sólo letras y entregue el mensaje desencriptado. La regla de desencriptación es la siguiente: la palabra desencriptada se obtiene recorriendo desde el final de la palabra hasta el comienzo, considerando solo las letras en ubicaciones impares. Empezando desde la última letra. La intención es obtener el lugar de la reunión.

```
>>> codigo_palabra('aczaarltp')  
'plaza'
```

```
>>> codigo_palabra('axruatgrrreov')  
'vergara'
```

Ejercicio 2.2

- b) Crear una función `codigo_hora(codigo)` que reciba un `codigo` encriptado de sólo números y el carácter `:` y entregue el mensaje descryptado en formato `hh:mm`. La regla de descryptación es la siguiente: sumar cada dígito anterior al carácter `:` y calcular el resto de la división entre esa suma y 24. Luego, lo mismo con los dígitos después del carácter `:`, pero la división es entre la suma de esos dígitos y 60.

La intención es obtener la hora de la reunión de los empresarios.

```
>>> codigo_hora('776199:68556')  
'15:30'
```

Ejercicio 2.3

- c) En un ataque de desconfianza, los empresarios incorporaron a su sistema infalible una palabra, también encriptada, que indica el fin del mensaje completo.

Usted debe crear un programa que reciba todos los códigos encriptados, desencriptarlos y mostrarlos por pantalla para crear el mensaje completo. La palabra de término desencriptada es: *'acun'*.

Ingrese codigo: **aczaarltp**
Ingrese codigo: **axruatgrrreov**
Ingrese codigo: **776199:68556**
Ingrese codigo: **npuecea**
El mensaje es: plaza vergara 15:30