

Problema 1: Simulador de datos para accidentes

Se necesita llenar una base de datos para probar una interfaz de registro de accidentes laborales. Para esto se le solicita crear un simulador que devuelva por consola los datos de personas y accidentes (No es necesario que estén relacionados).

Las personas tienen que tener los siguientes datos al menos: Nombre, edad, sexo, rut(No necesariamente debe corresponder el digito verificador).

Los accidentes deben tener al menos los siguientes datos: Tipo (Se le otorgará una lista con los tipos), cantidad de días de licencia, procedimiento aplicado(indicador que informa si después de un accidente se actuó como corresponde o no)

Para probar este simulador haga que se pida una cierta cantidad de personas y las cree. Del mismo modo que se solicite una cierta cantidad de accidentes y los cree.

Se entregará una lista de nombres, apellidos y los tipo de accidentes que pueden haber para facilitar el trabajo.

Se prohíbe el uso de clases, así como de librerías no vistas en clase. Se permite el uso de funciones

hint: Se recomienda el uso del modulo random.

Problema 2: Cubic Tap Code

El Cubic Tap code es un sistema de cifrado de datos en donde se convierten letras en un sistema de puntos basados en su ubicación respecto a 3 cuadros.

1	1	2	3
1	A	B	C
2	D	E	F
3	G	H	I

Cuadro 1: Primera tabla

2	1	2	3
1	J	K	L
2	M	N	O
3	P	Q	R

Cuadro 2: Segunda tabla

3	1	2	3
1	S	T	U
2	V	W	X
3	Y	Z	-

Cuadro 3: Tercera tabla

El sistema de codificación se basa en 3 grupos de puntos, el primer grupo indica el número de tabla. el segundo grupo se refiere a la columna en que está la letra, y el tercer grupo está dado para la fila en donde se encuentra. Por ejemplo la letra B se codifica ". . ." (Primera tabla, segunda columna, primera fila). El espacio entre letras se representa con "... . . .".

Se solicita que se creen 2 funciones encode() y decode() donde en encode se deben entregar las string en mayúsculas y separada por espacio (Puede obviarse mayúsculas y caracteres especiales). Decode debe hacer lo inverso.

Ejemplos:

encode("N")=>"..."

```
encode("HELLO WORLD") => "...".
```