

## Ejercicios 4.1

### Ejercicio 4.1

Introducir una cadena por teclado e imprimirla sin vocales, sin consonantes y con las vocales en mayúsculas (las consonantes deben quedarse tal como estén). Para cada una de estas tareas, se debe crear una función diferente. Ejemplo:

```
Introduce texto: Programacion
Prgrmcn
oaaio
PrOgrAmAcIOn
```

### Ejercicio 4.2

Crear una función que genere una contraseña aleatoria. La contraseña tendrá una longitud aleatoria entre 7 y 10 caracteres. Cada carácter se seleccionará aleatoriamente de las posiciones 33 a 126 de la tabla ASCII. La función no recibirá parámetros y devolverá la contraseña aleatoria como resultado.

Nota: Es probable que sea de utilidad la función chr de python.

### Ejercicio 4.3

Crear un programa que evalúe una contraseña e indique si es correcta. Se define una contraseña correcta como la que tiene al menos 8 caracteres y contiene al menos una mayúscula, al menos una minúscula y al menos un número. La función devolverá True si la contraseña que se le pasa como parámetro es correcta o False si no.

En el programa principal (que no debe ejecutarse si se importa como un módulo) se debe solicitar una contraseña e indicar si es correcta o no.

### Ejercicio 4.4

Crear un programa que importe como módulos los dos ejercicios anteriores y genere una contraseña aleatoria que sea correcta según las condiciones del ejercicio anterior. Al mostrar la contraseña también debe indicar el número de intentos necesarios para conseguirla. (No es necesario hacer test unitarios)

### Ejercicio 4.5

Diseñar un programa que partiendo de esta lista:

```
["cerillas", "patrulla", "papel", "azor", "alerones", "conversar", "sollozo", "manzana"]
```

Permita jugar al ahorcado (no es necesario imprimir la horca) con 5 posibles fallos. Es necesario imprimir el texto que quede por adivinar usando guiones bajos. Ejemplo:

```
Quedan 5 fallos.
- - - -
Letra: a
Bien, hay una a
Quedan 5 fallos.
a _ _ _
Letra: b
Ops, fallo.
Quedan 4 fallos.
a _ _ _
Letra:
```