

Prueba validación PAS: Python y Bash

Nombre:

1. *Prueba Python*: Desarrollar un script en python que a partir de una lista de enteros ya creada (tomad como ejemplo `example = [3, 4, 3, 5, 6, 7, 8, 8]`), genere otra lista de enteros a partir de la base. En esta segunda lista, los elementos serán los enteros de la primera lista concatenados n veces (para el caso tomad $n = 3$) y sin posibles repeticiones. Para el ejemplo considerado, la lista resultado sería `[333, 444, 555, 666, 777, 888]`. Aspectos a tener en cuenta:
 1. Los elementos de la lista resultado deben ser ENTEROS
 2. El resultado debe ser una LISTA
 3. Puede resolverse en una única línea de código
2. *Prueba Bash*: Desarrollar un script en bash que ejecute todos los ficheros `.c` que tenéis en la carpeta disponible en la url <http://www.uco.es/consigna/> (fichero `pruebaValidacion.zip` contraseña: jueves). Aspectos a tener en cuenta:
 1. Descargad la carpeta y situarla en el escritorio
 2. Utilizar para la práctica UNICAMENTE comandos y sentencias estudiadas en clase. No se considerarán otro tipo de sentencias (descargadas o copiadas de la Red).
 3. Puede resolverse en 4 líneas de código utilizando lo ya estudiado en clase.

Tiempo total para la prueba: 40 minutos

Prueba de Validación Mayo

(RECUPERACIÓN)

1. **Ejercicio Bash:** En primer lugar, descargar de la consigna (<http://www.uco.es/consigna/>) el fichero "PruebaValidacionMayo.zip", extraer su contenido documentos y situarlo en vuestra carpeta Escritorio. El objetivo de este primer problema será el de desarrollar un script en bash en el que se ejecuten de forma ordenada (atendiendo al número de líneas) los ficheros de código python.

Tened en cuenta que estamos listando por orden de número de líneas de código. De esta forma, el orden de ejecución será: primero Helloworld3.py (2 líneas), segundo Helloworld1.py (3 líneas) y finalmente Helloworld2.py (4 líneas).

La salida de ejecución será:

```
$bash ListaCodigo  
Hello, world! from 3  
Hello, world! from 3  
Hello, world! from 1  
Hello, world! from 1  
Hello, world! from 1  
Hello, world! from 2  
Hello, world! from 2  
Hello, world! from 2  
Hello, world! from 2
```

2. **Ejercicio Python:** Al igual que en el problema anterior, debéis descargar de nuevo de la consigna el fichero de texto foo.txt. El objetivo del script será el de crear una lista que contendrá tantos elementos como líneas tenga el fichero de texto foo.txt. Cada una de los elementos será de nuevo una lista de listas en la que cada objeto tendrá dos elementos: para las palabras que contengan el carácter "a", la palabra en minúscula, y la longitud de la palabra. Se valorará el uso de listas por comprensión.

Para el fichero de ejemplo:

```
Hello world auxiliar  
Hola mundo auxiliar  
Case 98
```

La salida sería:

```
[[[auxiliar,8]][[auxiliar,8],[hola,4]][[case,4]]]
```

La contraseña para los ficheros de la consigna es: PAS.

Tiempo estimado: 1 hora y 15 minutos.