

## SIMULACRO DE EXAMEN DE PROGRAMACIÓN II

### Normas:

- El fichero entregado debe ser un fichero válido de Python (NO una sesión interactiva de Jupyter), es decir, debe tener la extensión .py.
- Debéis añadir todas las funciones de los ejercicios al fichero plantilla.py.
- Las funciones implementadas deben respetar la signatura definida en el ejercicio (nombre, parámetros y valores de retorno correctos).
- La entrega se encuentra abierta en <https://pracdlsi.dlsi.ua.es/>
- Recordad que la red puede ir lenta, y que el servidor se cierra exactamente a la hora de finalización del examen: reservad al menos 5 minutos al final del examen para la entrega.
- No olvidéis indicar al inicio del fichero vuestro nombre y turno (e.g. Viernes 13h-53h) mediante un comentario Python. **No uses ningún acento ni ñ.**
- En el cuerpo de las funciones **no debe haber ninguna sentencia de tipo input ni print.**
  - Si queréis que el ejercicio incluya vuestro código de prueba, lo debéis poner al final del archivo .py, dentro del bloque, incluido en la plantilla del examen:

```
if __name__ == "__main__":  
    pass  
    #aquí vuestro código de prueba
```

- Notad cómo tanto name como main van precedidos y sucedidos de DOS guiones bajos.
- Podéis entregar el fichero .py tantas veces como queráis hasta que se cierre el servidor de prácticas. **Se contará como válida exclusivamente la última entrega realizada.**
- No entreguéis código que contenga errores sintácticos: abortará el corrector y eso supondrá un 0 en el ejercicio.
- Podéis ir guardando vuestro código en la unidad D: del ordenador. De este modo, si se os reiniciase por cualquier motivo, no se perderá vuestro trabajo.

### Ejercicio 1 (10 puntos):

En matemáticas, la sucesión de Fibonacci es una sucesión infinita de números naturales como la siguiente:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34...

La sucesión comienza con dos números naturales cualesquiera y a partir de estos, «cada término es la suma de los dos anteriores», es la relación de recurrencia que la define.

Crea una clase **Fibonacci** que contenga en método estático llamado **termino** que calcule el término número n de la sucesión de Fibonacci. Si la entrada a la función no es un número positivo, la función estática devolverá **None**.

```
# Código de prueba
if __name__ == "__main__":
    print(Fibonacci.termino(1))          # 1
    print(Fibonacci.termino(10))         # 55
    print(Fibonacci.termino(-1))         # None
    print(Fibonacci.termino("hola"))     # None
```