



# Examen de Prácticas

22 de junio de 2023

## Fundamentos de Informática

Grado en Tecnologías Industriales

*Duración: 50m (más 15m para preparación y entrega)*

**Nombre y apellidos:** \_\_\_\_\_ **NIA:** \_\_\_\_\_

**Recuerda:** El examen se realiza de forma individual.

**Entrega:**

- El examen se entrega en la tarea de Moodle preparada para ello.
- Se entrega un único archivo **examen\_XXXXXX.zip** (sustituyendo XXXXXX por tu NIA) que contenga **los ficheros del código fuente** (.pas) de los programas que implementes en este examen.
- Deberán compilar en las mismas condiciones que la práctica original.

## Ejercicio 1

**[ 5 puntos ]**

En el **Ejercicio 3** de la **Práctica 3** has implementado un programa 'afinFrase.pas' que cifra una frase, utilizando el cifrado afín. Dicho programa lee de teclado las constantes de decimación y desplazamiento, y una frase completa (acabada en **fin de línea**). A continuación, muestra por pantalla el resultado de aplicar el cifrado afín con esas constantes a toda la frase completa. Los caracteres que no sean letras mayúsculas o minúsculas del alfabeto anglosajón (por ejemplo, los dígitos, el espacio en blanco o los signos de puntuación) no son modificados.

Se pide escribir un nuevo programa en Pascal '**afinFichero.pas**' (tomando como partida el 'afinFrase.pas') que, en lugar de cifrar una frase, cifre un **fichero de texto** completo. Para ello, leerá la información de un fichero original con texto sin cifrar, y escribirá un nuevo fichero de texto con el texto cifrado (ver ejemplo al final de esta página). El nombre del fichero original será 'original.txt', y el del texto cifrado será 'cifrado.txt'.

La interacción debe ser como la ilustrada en el ejemplo de ejecución que se muestra a continuación (en negrita lo que introduce el usuario):

```
Introduce la constante de decimacion A: 11
Introduce la constante de desplazamiento B: 4
Cifrando fichero...
...hecho.
```

**original.txt:**

```
iHola!
Esto es un fichero de prueba
para comprobar si tu programa
puede aplicar el cifrado afin.
Dabale arroz a la zorra el abad.
¡Adios!
```

**cifrado.txt:**

```
iDcve!
Wufc wu qr hoadwjc lw njqwpe
neje acgnjcpej uo fq njcsjege
nqwlw envoaew wv aohjelc ehor.
Lepevw ejjct e ve tcjje wv epel.
iElocu!
```

Puedes definir procedimientos o funciones adicionales si los consideras necesarios.

Se valorará el correcto uso de subprogramas.

**Entrega:** afinFichero.pas con la solución del ejercicio.

*(continúa en la siguiente página)*

## Ejercicio 2

**[ 5 puntos ]**

En el **Ejercicio 3** de la **Práctica 4** se pidió un programa 'islas.pas' que permitía almacenar los datos de islas del tesoro (nombre, posición y valor del tesoro), hasta un máximo de 100 islas, y la ubicación del usuario. Tras leer dicha información de teclado y almacenarla, el programa ofrecía información sobre qué isla era la más cercana al usuario, y a qué distancia estaba.

Se pide escribir un nuevo programa 'islasTotal.pas' (partiendo de 'islas.pas') que, en vez de mostrar por pantalla la información de la isla más cercana al usuario, muestre la cantidad total de tesoros que se puede encontrar a menos de una distancia dada de la posición del usuario, es decir, la suma de los tesoros de todas las islas que estén dentro de un círculo de radio dado.

A continuación, se muestra un ejemplo de cómo debe ser la interacción entre el usuario y el programa (en negrita lo que introduce el usuario), así como la salida por pantalla a la que da lugar como resultado de la ejecución:

```
Introduce tu posicion (longitud y latitud): 0.31 48.88 ↵
Introduce el numero de islas a almacenar: 3 ↵
Datos de la isla 1:
Nombre de la isla: Bonita ↵
Posicion de la isla (longitud y latitud): 0.9 45.41 ↵
Valor del tesoro (monedas): 3250 ↵
Datos de la isla 2:
Nombre de la isla: Nohaynada ↵
Posicion de la isla (longitud y latitud): 1.12 44.20 ↵
Valor del tesoro (monedas): 6813 ↵
Datos de la isla 3:
Nombre de la isla: Calavera ↵
Posicion de la isla (longitud y latitud): 10.12 35.70 ↵
Valor del tesoro (monedas): 13215 ↵
Radio de busqueda: 10.0 ↵
Tesoro total: 10063 monedas
```

Observaciones/Recordatorios:

- Puedes definir procedimientos y/o funciones adicionales si lo consideras adecuado. Se valorará el correcto uso de subprogramas.
- Como en `islas.pas`: (i) las posiciones, tanto de las islas como del usuario, vienen dadas por sus coordenadas (longitud y latitud, en grados), y (ii) dadas dos posiciones  $p1$  y  $p2$ , definidas por su longitud y latitud en grados,  $p1 = (\theta_1, \phi_1)$  y  $p2 = (\theta_2, \phi_2)$ , puedes aproximar la distancia  $d$  entre ellas como  $d(p1, p2) = \sqrt{(\theta_1 - \theta_2)^2 + (\phi_1 - \phi_2)^2}$ . Se recomienda utilizar una función para calcular esta distancia a partir de la información de dos posiciones.
- En el ejemplo de ejecución consignado arriba, las islas Bonita y Nohaynada son las que están dentro de un círculo de radio 10.0 centrado en la posición actual del usuario (0.31, 48.88), de ahí que el tesoro total sea 10063 monedas (3250 de la isla Bonita más 6813 de la isla Nohaynada).

**Entrega:**    `islasTotal.pas` con la solución del ejercicio.