



RECORDATORIO: El examen de Prácticas se realiza de forma individual.

ENTREGA: Se entregarán los archivos de código fuente `logExamen.pas` e `islasExamen.pas` que cumplan lo especificado en los enunciados de este examen empaquetados en un único archivo, llamado **examenPracticasFeb.zip**.

La entrega se hará utilizando el método habitual de entrega de prácticas, en la tarea de Moodle habilitada para ello.

NOTA: Recordad que se comprobará la existencia de plagios en el código.

Ejercicio 1

[5 puntos]

En el **Ejercicio 4** de la **Práctica 3** se pidió un programa `cosx.pas` que calculaba el desarrollo en serie de Taylor de la función coseno, de tal forma que el usuario introducía un valor para x y un valor para el número de términos de la serie a calcular, n .

Se pide escribir un nuevo programa en Pascal **logExamen.pas** (se puede tomar como partida el del coseno) que, en lugar de calcular la función coseno, calcule el valor aproximado de la función $\ln\left(\frac{1+x}{1-x}\right)$, mediante la siguiente serie:

$$\ln\left(\frac{1+x}{1-x}\right) = \sum_{n=1}^{\infty} \frac{2}{2n-1} x^{2n-1} = 2x + \frac{2}{3}x^3 + \frac{2}{5}x^5 + \dots$$

La interacción debe ser como la ilustrada en el ejemplo de ejecución que se muestra a continuación (en negrita lo que introduce el usuario), mostrando el resultado de la aproximación mediante el desarrollo en serie, y el resultado calculado mediante la función disponible en Pascal `ln(...)`:

```
1 Introduce x y numero de terminos a calcular: 0.1 -6 ↵
2 Introduce x y numero de terminos a calcular: 0.1 6 ↵
3 ln((1+0.100)/(1-0.100)) aproximado con 6 iteraciones = 0.200670695462136
4 ln((1+0.100)/(1-0.100)) calculado con funcion Pascal = 0.200670695462151
```

Puedes definir procedimientos o funciones adicionales si los consideras necesarios. Se valorará el correcto uso de subprogramas.

NOTA 1: Si el usuario no introduce un entero mayor que cero para el número de términos a calcular, habrá que seguir pidiéndoselo hasta que lo sea.

NOTA 2: El resultado se mostrará, como en el ejemplo, con 15 decimales.

NOTA 3: Para hacer las pruebas, toma valores de x en el rango $0 < x < 1$.

Entrega: `logExamen.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: datos de islas del tesoro (nombre, posición y valor del tesoro de hasta 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 modificar dicho programa para que, en vez de mostrar por pantalla la información de la isla más cercana al usuario, muestre la información (nombre, posición y valor del tesoro) de la isla con el mayor tesoro, así como la distancia de la misma al usuario.

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: 2 ↵
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 ↵
La isla con mayor tesoro es:
Nombre: Nohaynada
Posicion (longitud y latitud): 1.12 44.20
Tesoro: 6813 monedas
Distancia (en grados): 4.750
```

Observaciones/Recordatorios:

- Puedes definir procedimientos y/o funciones adicionales si lo consideras adecuado. Se valorará el correcto uso de subprogramas.
- Se recomienda usar los mismos tipos de dato que en `islas.pas`: **`tpIsla`** (que permite almacenar: el nombre de la isla, su posición, y el valor del tesoro escondido en ella) y **`tpGrupoIslas`** (que permite almacenar la información de un máximo de 100 islas, aunque puede haber menos de 100).
- 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) sean dos posiciones $p1$ y $p2$ dadas por 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.

Entrega: `islasExamen.pas` con la solución del ejercicio.