



Examen de Prácticas

12 de enero de 2024

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** (los .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 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 **arctanExamen.pas** (se puede tomar como partida el del coseno) que, en lugar de calcular la función coseno, calcule el valor aproximado de la función $\arctan(x)$ mediante la siguiente serie:

$$\arctan(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$$

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 `arctan(...)`:

```
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 arctan(0.100) aproximado con 6 iteraciones = 0.099668652491154
4 arctan(0.100) calculado con funcion Pascal = 0.099668652491162
```

Puedes definir procedimientos o funciones adicionales si los consideras necesarios.

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: `arctanExamen.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 un listado de todas las islas a una distancia menor que D, siendo este un valor introducido por el usuario. Se debe controlar que la distancia D sea siempre positiva, y volver a pedirla al usuario en caso contrario.

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 ↵
Mostrar islas a una distancia menor de: -20.5 ↵
La distancia debe ser positiva.
Mostrar islas a una distancia menor de: 10.0 ↵
Islas a una distancia menor de 10.0
- Bonita (tesoro 3250)
- Nohaynada (tesoro 6813)
```

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 la isla Calavera no aparezca en el listado.

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