Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Examen de Teoría

10 de septiembre de 2020

Fundamentos de Informática

Grado en Tecnologías Industriales

Duración: 3 horas

Entrega:

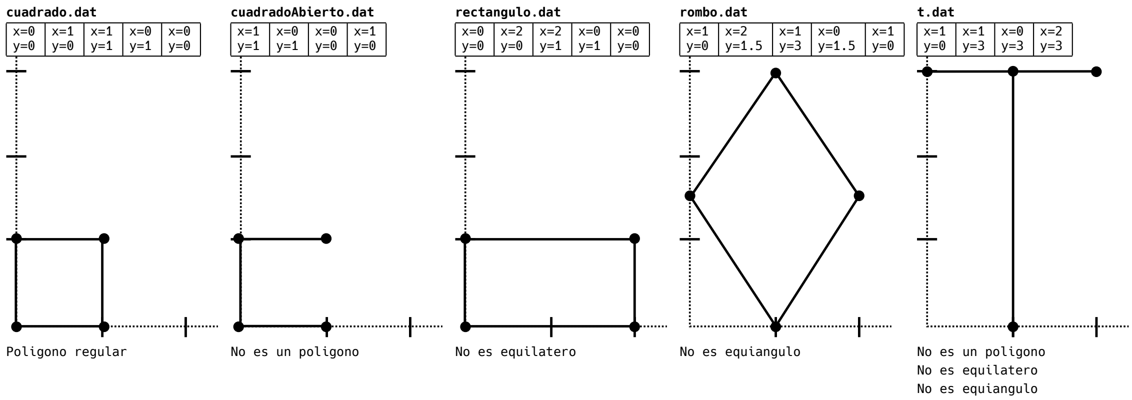
- Escribe tu nombre completo en **todas** las hojas que entregues como solución del examen.
- Responde a cada ejercicio comenzando **en una hoja diferente**, los ejercicios se entregan por separado.
- La resolución del examen no se puede escribir a lápiz ni con bolígrafo rojo. Lo que esté escrito a lápiz o en rojo será ignorado.

Ejercicio 1 **[3.5 puntos]**

Una herramienta de diseño CAD permite modelar polilíneas en base a **secuencias de vértices**. Cada vértice se representa mediante sus dos coordenadas reales, x e y. El programa CAD guarda cada polilínea en un fichero secuencial de vértices (registros). Como primera tarea, **define** la estructura de datos que represente una polilínea. Se pretende extender la herramienta CAD con nueva funcionalidad. En concreto **se pide** comprobar si una polilínea (obtenida a partir del nombre del fichero secuencial que la representa) es un **polígono regular** o no. Para ello hay que hacer las siguientes comprobaciones:

- Si la polilínea es un **polígono**, es decir, si el último vértice coincide con el primero.
- Si el polígono es **equilátero**, es decir, si todas sus aristas (rectas que unen vértices contiguos) tienen la misma longitud. La longitud se mide con la distancia euclídea, es decir, dados dos vértices con coordenadas $\mathbf{v_0} = (x_0, y_0)$ y $\mathbf{v_1} = (x_1, y_1)$ la arista $\mathbf{a_{01}} = \mathbf{v_1} - \mathbf{v_0} = (x_{01}, y_{01}) = (x_1 - x_0, y_1 - y_0)$ tendrá longitud $|\mathbf{a_{01}}| = \sqrt{x_{01}^2 + y_{01}^2}$.
- Si el polígono es **equiángulo**, es decir, si los ángulos que forman todos los pares contiguos de aristas son idénticos. Para calcular el ángulo hay que aplicar las dos definiciones de producto escalar $\mathbf{a_{01}} \cdot \mathbf{a_{12}} = x_{01}x_{12} + y_{01}y_{12} = |\mathbf{a_{01}}||\mathbf{a_{12}}| \cos \alpha$ donde α es el ángulo que forman las dos aristas (que debe ser despejado de la ecuación). Se debe tener en cuenta que la última arista y la primera arista también forman un ángulo que debe ser comprobado.

El programa deberá mostrar por pantalla si la polilínea es un polígono regular o no lo es. Si no es un polígono regular, se deberá mostrar por pantalla las razones por las que no es regular (no es polígono, no es equilátero y/o no es equiángulo). Puedes suponer que la polilínea tiene como mínimo tres vértices, pero existe límite máximo alguno de número de vértices. Se valorará negativamente que el fichero secuencial que define la polilínea se recorra más de una vez.



NOTA: Pascal dispone de la funcion `arccos` que permite calcular el arco cuyo coseno es un valor dado (la función inversa al coseno)

Ejercicio 2 **[3.5 puntos]**

En un futuro no distópico, una pandemia mundial se extiende por el planeta, limitando las posibilidades de interacción entre personas. Ante una falta de vacuna o tratamiento efectivo en todos los casos, una de las medidas más útiles para prevenir el contagio es el distanciamiento entre personas y el uso de mascarillas.

Una sala de conciertos quiere diseñar un programa para, en todo momento, comprobar que ninguno de los asistentes se salta las reglas de distanciamiento entre personas sentadas en su patio de butacas (no se permiten personas de pie).

El patio de butacas tiene las sillas dispuestas de forma regular, en una cuadrícula de N filas y M columnas. De acuerdo con las recomendaciones de las autoridades sanitarias, las reglas para el distanciamiento entre personas son las siguientes:

- R1. La distancia entre dos personas, si al menos una de las dos va sin mascarilla, debe ser de al menos dos butacas vacías en cualquiera de las direcciones (horizontal, vertical, o diagonal).
- R2. Para cada persona sentada, en una región de 5×5 butacas centrada en dicha persona (ver Fig. 1), un máximo de un tercio de butacas estarán ocupadas, independientemente de la mascarilla. En los bordes esta condición se aplica a los vecinos de una persona hasta una distancia de 2 butacas (como muestra la posición X2 de la Fig. 1). En caso de presencia de decimales en el cálculo del máximo número permitido de butacas ocupadas (que denominaremos max), se tomará la opción conservadora (el menor valor de los dos enteros más próximos). Así, $max = 8$ para X1 y $max = 4$ para X2.

Se pide, utilizando el lenguaje Pascal:

(a) Proponer una estructura de datos adecuada para representar la distribución de personas en el patio de butacas, `tpPatioButacas`.

(b) Desarrollar una función que, dada una distribución de personas en el patio de butacas, devuelva si dicha distribución es correcta o no, de acuerdo a si cumple las reglas R1 y R2 anteriores.

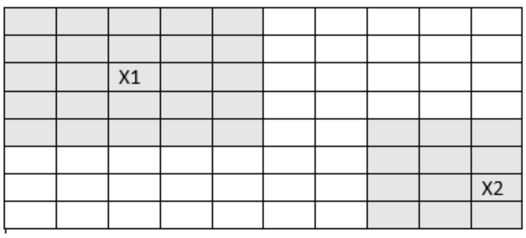


Figura 1:

NOTA (1): Se valorará la legibilidad de la solución.

NOTA (2): Los valores N y M vendrán dados por sendas constantes.

Ejercicio 3 **[3 puntos]**

Un polinomio de grado n es una función del tipo

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_i x^i + \ldots + a_n x^n$$

donde los valores a_i son coeficientes reales.

Define un tipo de datos **tpPoly** que permita almacenar un polinomio de coeficientes reales de cualquier grado hasta un máximo `MAX_GRADO`.

Define una función que evalúe un polinomio para un valor de x dado:

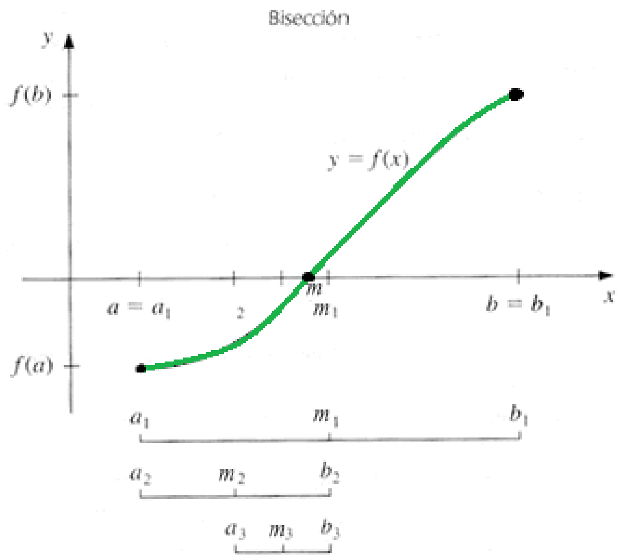
```
function poly_eval(p: tpPoly): real;
```

(puedes usar la función `power(base: real,exponente:real):real` de Pascal).

Una tarea común con polinomios es buscar ceros, es decir, valores de x que hacen nulo el valor del polinomio (raíces). Para ciertos polinomios (por ejemplo, de grado 2) sus raíces pueden expresarse mediante fórmulas, pero en el caso general es necesario usar métodos numéricos en un ordenador.

Uno muy simple y conocido es el **método de bisección**:

- empezamos con un par de valores de x entre los cuales estamos seguros de que existe un cero, llamémosles a y b. Definen el intevalo de valores $[a, b]$. Como entre ambos hay un cero, los signos de $p(a)$ y $p(b)$ son distintos.
- calculamos el centro (punto medio) del intervalo $m = (a + b)/2$
- si el valor del polinomio en m es del mismo signo que en a, nuestro intervalo pasa a ser $[m, b]$. En caso contrario, el nuevo intervalo es el $[a, m]$.



Ese proceso se repite hasta que el tamaño del intervalo (diferencia entre los dos valores límite) sea menor que una cota dada, definida mediante una constante real ERROR (por ejemplo, 0.0001). El valor de punto de cero es el promedio de los valores finales de a y b.

Define en Pascal una función que calcule el valor del cero de un polinomio contenido en un intervalo inicial $[a, b]$:

```
function poly_zero(p: ...; a,b: ...): real;
```