

PROGRAMACIÓN

(GRADOS EN INGENIERO MECÁNICO, ELÉCTRICO, ELECTRÓNICO INDUSTRIAL y QUÍMICO INDUSTRIAL)

Sesión	10 (Auto-evaluación: <i>arrays</i> unidimensionales)	
Temporización	1 hora (no presencial)	
Objetivos formativos	<ul style="list-style-type: none"> • Conocer la sintaxis de C para la definición de nuevas tipologías de datos y para la implementación de estructuras de datos estáticas de tipo <i>arrays</i>, tanto unidimensionales (vectores) como multidimensionales. • Diferenciar entre los conceptos de tipo de dato y de variable. • Representar conjuntos de datos estáticos y homogéneos mediante <i>arrays</i>. Construir los tipos de datos correspondientes, identificando los correspondientes elementos de la estructura de datos (tamaño, organización y tipo base). • Declarar variables de tipos de datos definidos por el programador. Acceder directamente a cada uno de los elementos individuales de una variable de tipo <i>array</i> mediante indexación. • Procesar todos los elementos de un <i>array</i>, de 1 en 1, empezando por el primero y siguiendo el orden físico de almacenamiento, mediante una instrucción de repetición controlada por un contador de iteraciones (recorrido secuencial). • Modificar el algoritmo básico del recorrido secuencial para adaptarlo a un problema particular: recorrido secuencial parcial, empezando por un elemento diferente del primero, retrocediendo el índice en lugar de avanzarlo, con varias condiciones de salida,... • Trasvasar elementos de información estructurados entre partes del programa a través de las interfaces de las funciones (lista de parámetros formales), tanto de <i>arrays</i> completos como de elementos individuales de un <i>array</i>. • Implementar programas modulares en lenguaje de programación C. Identificar y corregir errores sintácticos que surgen durante la codificación. • Resolver problemas sencillos con <i>arrays</i>, aplicando las operaciones básicas sobre los mismos (acceso directo a elementos individuales y acceso secuencial). • Probar con datos operacionales la correctitud de los módulos y programas desarrollados e identificar y corregir los errores lógicos que surjan. 	
Competencias a desarrollar	<ul style="list-style-type: none"> • RD1: Poseer y comprender conocimientos • RD2: Aplicación de conocimientos • UAL1: Conocimientos básicos de la profesión • UAL3: Capacidad para resolver problemas • UAL6: Trabajo en equipo • FB3: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en la ingeniería. 	X X X X X

Materiales	IDEs : Dev-C++/Code::Blocks (freeware)	
Tarea	Realizar la actividad de auto-evaluación propuesta después de terminar las actividades de la sesión 8 de Trabajo en Equipo Cooperativo. No debería utilizar ningún material ni ayuda externa en la realización de esta actividad.	
Fecha de entrega	No hay. Autocalifíquese de acuerdo con las indicaciones que se adjuntan. En caso de duda, consultar con el profesor.	
Criterios de éxito	<ul style="list-style-type: none">• Terminar en el tiempo previsto la tarea.• Obtener una calificación superior a 5 según baremo que se adjunta.	
Plan de trabajo	Actividad	Temporización
	Realizar individualmente (sin ayuda externa) el ejercicio de auto-evaluación propuesto (diseño + implementación + prueba), midiendo el tiempo empleado en su realización.	55 mn
	Auto-calificarse de acuerdo con el baremo adjunto.	5 mn

Ejercicio de auto-evaluación

Ejercicio	<p>Construir un programa para ajustar un conjunto de datos experimentales a una línea recta mediante el método de mínimos cuadrados:</p> $y = a + b * x \quad \left\{ \begin{array}{l} b = \frac{n * \sum(x_i * y_i) - \sum x_i * \sum y_i}{n * \sum(x_i^2) - (\sum x_i)^2} \quad \begin{array}{l} x_{med} = \sum x_i / n \\ y_{med} = \sum y_i / n \end{array} \\ a = y_{med} - b * x_{med} \\ r = \frac{n * \sum(x_i * y_i) - (\sum x_i) * (\sum y_i)}{[(n * \sum(x_i^2) - (\sum x_i)^2) * (n * \sum(y_i^2) - (\sum y_i)^2)]^{1/2}} \end{array} \right.$ <p>El programa presentará un menú en pantalla con las siguientes opciones:</p> <ul style="list-style-type: none"> • Insertar un nuevo punto experimental, introduciendo por teclado los valores de las variables x e y. • Imprimir en pantalla un listado numerado con todos los puntos experimentales disponibles. • Eliminar del conjunto de datos un punto, dado por teclado su número de orden dentro del conjunto. • Imprimir en pantalla la ecuación de la recta de regresión junto con el coeficiente de correlación. • Finalizar programa. 			
Datos de prueba		Opción menú	Datos de entrada	Resultados
		Recta regresión		Ningún punto introducido
		Insertar puntos	x = 100 y = 100	
		Recta regresión		Solo hay un punto experimental
		Insertar puntos	x = 300 y = 300	
			x = 200 y = 200	
		Imprimir puntos		Listado de puntos: 1.- (100.00,100.00) 2.- (300.00,300.00) 3.- (200.00,200.00)
		Recta regresión		y = 0.00 + 1.00 * x r = 1.00
		Eliminar puntos	nº punto: 1	
		Imprimir puntos		Listado de puntos: 1.- (300.00,300.00) 2.- (200.00,200.00)
		Eliminar puntos	nº punto: 1	
		Insertar puntos	x = 200 y = 400	
		Recta regresión		x = 200.00 r indeterminado
		Finalizar programa		

Asignatura	Programación		
Plan de Estudios	Grados en Ingeniero Mecánico, Eléctrico, Electrónico Industrial y Químico Industrial		
Actividad	Trabajo individual	Sesión	10
Tiempo empleado			

Apellidos, nombre	DNI	Firma

Resultados de la auto-evaluación: para obtener la calificación de esta prueba de auto-evaluación en una escala de 0 a 10, busque en primer lugar en las categorías de calificación la que mejor se ajuste en su caso al trabajo realizado, a continuación multiplique por dos el número de dicha categoría y finalmente reste 1 punto por cada hora adicional consumida (que exceda de las 2 horas previstas en esta prueba). Anote en el apartado de observaciones las principales dificultades que ha encontrado al realizar esta actividad (si las hubiera).

Calificación obtenida (categoría*2-penalización exceso tiempo):	
Categorías de calificación	5: Funcionamiento correcto del programa sin modificaciones del diseño inicial. 4: Funcionamiento correcto tras modificación del diseño inicial. 3: Funcionamiento incorrecto en algunos casos de prueba. 2: Funcionamiento incorrecto en todos los casos de prueba. 1: Se ha diseñado el algoritmo pero no se ha implementado el programa. 0: No se ha diseñado el algoritmo
Observaciones	