

## PROGRAMACIÓN

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

<b>Sesión</b>	9 (Auto-evaluación: examen parcial: Programación Estructurada + Diseño Modular)
<b>Temporización</b>	6 horas (no presenciales) → ver planificación temporal
<b>Objetivos formativos</b>	<p><u>Programación Estructurada:</u></p> <ul style="list-style-type: none"> <li>• Conocer la sintaxis básica del lenguaje de programación C: tipos de datos, constantes, variables, operadores, instrucciones de asignación y de control. Identificar algunas características genuinas de este lenguaje de programación.</li> <li>• Conocer las principales funciones de la biblioteca estándar de C para la entrada/salida de información por el terminal así como para cálculos matemáticos básicos.</li> <li>• Implementar algoritmos sencillos que hagan uso de las construcciones estructuradas de control (programación estructurada): secuencia, selección y repetición.</li> <li>• Identificar y corregir errores sintácticos del lenguaje de programación C que surgen durante la codificación.</li> <li>• Utilizar adecuadamente la función <b>scanf</b> para la entrada de datos al programa a través del teclado.</li> <li>• Presentar adecuadamente en pantalla los resultados de salida de un programa mediante la función <b>printf</b>.</li> <li>• Probar con datos operacionales la correctitud de los programas desarrollados e identificar y corregir los errores lógicos que surjan.</li> </ul> <p><u>Diseño Modular:</u></p> <ul style="list-style-type: none"> <li>• Conocer la estructura sintáctica de un programa modular en C: definiciones de funciones, prototipos de funciones, bibliotecas de funciones, paso de parámetros, tipo de dato puntero, funciones como parámetros, tipos de almacenamiento automático, externo, estático y registro.</li> <li>• Implementar programas modulares en lenguaje de programación C. Identificar y corregir errores sintácticos que surgen durante la codificación.</li> <li>• Construir módulos individuales (subprogramas), identificando su actividad funcional (entrada, salida, cálculo,...) y su interfaz para la comunicación de información con otros módulos (lista de parámetros formales + valor especial asociado al identificador del módulo). Distinguir entre parámetros de entrada y de salida. Documentar adecuadamente estos módulos mediante la notación sub-algorítmica.</li> <li>• Descomponer un problema en sub-problemas, identificando a partir del enunciado del mismo la información a procesar y la lista de tareas de manipulación de información; organizar esta lista de tareas de forma jerárquica, descomponiendo tareas complejas en tareas más simples. Representar mediante un diagrama en forma de árbol la estructura del programa.</li> <li>• Resolver problemas sencillos mediante recursividad como alternativa a</li> </ul>

	<p>la construcción de repetición: expresar el problema de forma recursiva distinguiendo entre el caso general y el caso base.</p> <ul style="list-style-type: none"> <li>• Utilizar módulos como parámetros con el objetivo de separar algunos algoritmos de cálculo de las funciones sobre las que actúan.</li> <li>• Probar con datos operacionales la correctitud de los módulos individuales (pruebas unitarias) y de los programas desarrollados e identificar y corregir los errores lógicos que surjan.</li> </ul>	
<b>Competencias a desarrollar</b>	<ul style="list-style-type: none"> <li>• RD1: Poseer y comprender conocimientos</li> <li>• RD2: Aplicación de conocimientos</li> <li>• UAL1: Conocimientos básicos de la profesión</li> <li>• UAL3: Capacidad para resolver problemas</li> <li>• UAL6: Trabajo en equipo</li> <li>• FB3: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en la ingeniería.</li> </ul>	<p>X</p> <p>X</p> <p>X</p> <p>X</p> <p></p> <p>X</p>
<b>Materiales</b>	No debería utilizar ningún material ni ayuda externa en la realización de esta actividad.	
<b>Tarea</b>	<p>Realizar los tres ejercicios del examen parcial propuesto como actividades de auto-evaluación de los temas 2 y 3, después de terminar las actividades de las sesiones 1 a 7 de Trabajo en Equipo Cooperativo, y de las sesiones 2 a 8 de Trabajo Individual.</p> <p>Autocalificarse de acuerdo con las indicaciones que se adjuntan. En caso de duda, consultar con el profesor.</p>	
<b>Fecha de entrega</b>	No hay (a ser posible antes de la sesión nº 8 de grupo de trabajo). Una vez completada la actividad, comunique al profesor los resultados de la misma, así como también los resultados de las actividades individuales realizadas previamente. Recuerde que es condición necesaria para poder aprobar la asignatura el haber realizado todas las actividades propuestas (tanto individuales como en equipo).	
<b>Criterios de éxito</b>	<ul style="list-style-type: none"> <li>• Terminar en el tiempo previsto la tarea.</li> <li>• Obtener una calificación superior a 5 según baremo que se adjunta.</li> </ul>	
<b>Plan de trabajo</b>	<b>Actividad</b>	<b>Temporización</b>
	Realizar individualmente (sin ayuda externa) los tres ejercicios de esta prueba de auto-evaluación (diseño + implementación + prueba), midiendo el tiempo empleado en su realización. Nota: para las pruebas utilizar los datos incluidos en los propios enunciados de los ejercicios.	350 mn
	Auto-calificarse de acuerdo con el baremo adjunto.	10 mn

<b>Asignatura</b>	Programación		
<b>Plan de Estudios</b>	Grados en Ingeniero Mecánico, Eléctrico, Electrónico Industrial y Químico Industrial		
<b>Actividad</b>	Trabajo individual	<b>Sesión</b>	9
<b>Tiempo empleado</b>			

<b>Apellidos, nombre</b>	<b>DNI</b>	<b>Firma</b>

**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 las que mejor se ajusten en su caso a cada uno de los ejercicios de la prueba, a continuación multiplique dicha categoría por el peso correspondiente del ejercicio y sume los números obtenidos. Finalmente reste 1 punto por cada hora adicional consumida que exceda de las horas previstas en esta prueba. Anote en el apartado de observaciones las principales dificultades que ha encontrado al realizar esta actividad (si las hubiera).

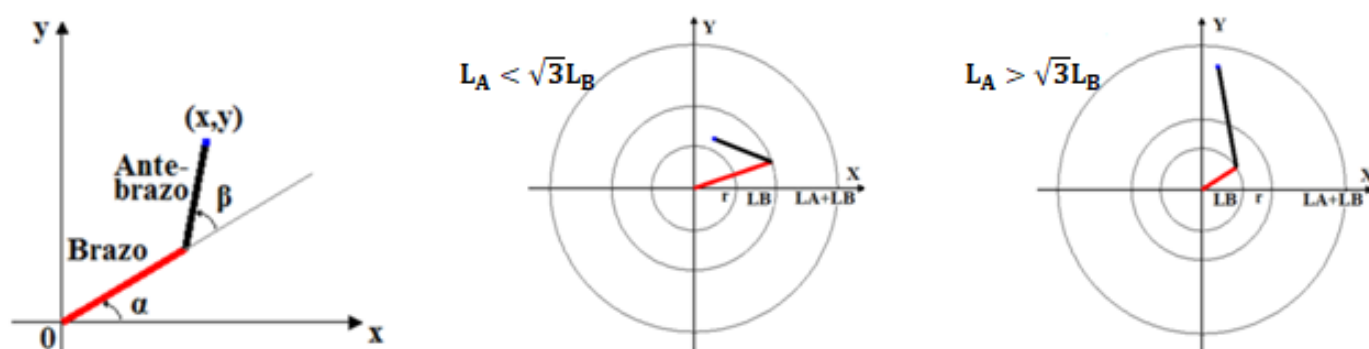
Calificación obtenida			categoría	Peso	Puntuación
		Ejercicio 1		*0.8	
		Ejercicio 2		*0.6	
		Ejercicio 3		*0.6	
		Penalización exceso de tiempo		-	
		Puntuación Total			
<b>Categorías de calificación</b>		<b>5:</b> Funcionamiento correcto del programa sin modificaciones del diseño inicial. <b>4:</b> Funcionamiento correcto tras modificación del diseño inicial. <b>3:</b> Funcionamiento incorrecto en algunos casos de prueba. <b>2:</b> Funcionamiento incorrecto en todos los casos de prueba. <b>1:</b> Se ha diseñado el algoritmo pero no se ha implementado el programa. <b>0:</b> No se ha diseñado el algoritmo			
<b>Observaciones</b>					

**Examen Parcial Fundamentos de Programación**  
**(Grados en Ingeniería Mecánica, Eléctrica, Electrónica Industrial y Química Industrial)**

**Ejercicio 1 (4.0 p):** construir un programa en C lo más modular posible (atendiendo a los criterios de modularidad). Documentar el diseño preliminar con el diagrama de módulos (estructura del programa) y los prototipos de las funciones en C, y el diseño detallado con las definiciones de las respectivas funciones en C.

**Ejercicios 2 y 3 (3.0 p/ejercicio):** implementar en C solamente la función indicada (no hay que construir el programa completo).

**Ejercicio 1:** Considerar un robot articulado bidimensional que consta de dos elementos rectangulares, un brazo de longitud  $L_B$  con un extremo fijo anclado al origen de coordenadas, que puede girar un ángulo  $\alpha$  cualquiera respecto del eje  $X^+$ , y un antebrazo de longitud  $L_A$  unido al otro extremo del brazo, que puede girar un ángulo  $\beta$  (comprendido entre  $0^\circ$  y  $150^\circ$ ) en sentido anti-horario con respecto al eje del brazo. El movimiento del sistema robotizado es controlado mediante los dos ángulos  $\alpha$  y  $\beta$ , pudiendo acceder el extremo libre del brazo solamente a las posiciones cartesianas comprendidas en la corona circular de radios  $r_{min} = \sqrt{L_A^2 + L_B^2} - \sqrt{3} * L_A * L_B$  y  $r_{max} = L_A + L_B$ :



Construir un programa en C para controlar el movimiento del robot a posiciones específicas del plano; el programa leerá por teclado una a una las posiciones finales  $(x,y)$  de posicionamiento del robot; cada punto introducido será validado para comprobar si es accesible, desplazándose a continuación el robot a la posición indicada (escribir en pantalla los desplazamientos angulares en grados correspondientes de los dos elementos del robot y la posición angular final). En caso de que dicha posición sea inalcanzable, se presentará un mensaje volviéndose a leer un nuevo punto. El programa finalizará cuando se introduzca como posición final el origen del sistema de coordenadas  $(0,0)$ , en cuyo caso los dos elementos del robot se mueven a la posición angular  $(\alpha, \beta) = (0,0)$ . Ejemplo:  $L_B=100.0$   $L_A=75.0$   $(\alpha_0, \beta_0) = (0,0)$

Posición final		Desplazamiento angular		Posición angular final	
x	y	$\Delta\alpha$ (°)	$\Delta\beta$ (°)	$\alpha_f$ (°)	$\beta_f$ (°)
100	25	-29.28	109.47	330.72	109.47
200	200	Posición inalcanzable		330.72	109.47
25	75	53.06	19.21	23.79	128.68
125	-25	-71.10	-41.07	312.69	87.61
0	0	47.31	-87.61	0.00	0.00

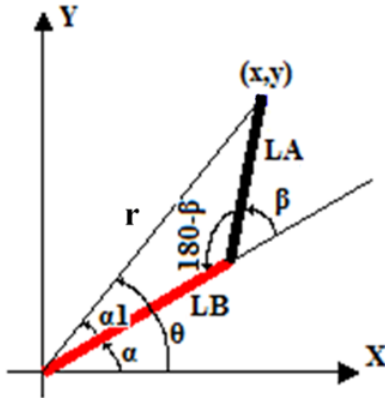
Nota:

$\alpha_f$ :  $[0,360]$  grados

$\Delta\alpha$ :  $[-180,180]$  grados

(menor desplazamiento angular del brazo posible)

Nota: en las siguientes figuras se ilustran respectivamente, la relación entre las coordenadas del punto final y el posicionamiento angular del robot, así como las relaciones dentro de un triángulo entre los lados y un ángulo del mismo.



$(x, y)$ : coordenadas cartesianas

$(r, \theta)$ : coordenadas polares

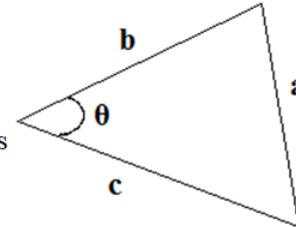
$$\begin{cases} r = \sqrt{x^2 + y^2} \\ \cos \theta = \frac{x}{r} \end{cases}$$

$$\alpha = \theta - \alpha_1$$

Nota: la función `acos` devuelve un valor comprendido en  $[0, \pi]$  radianes

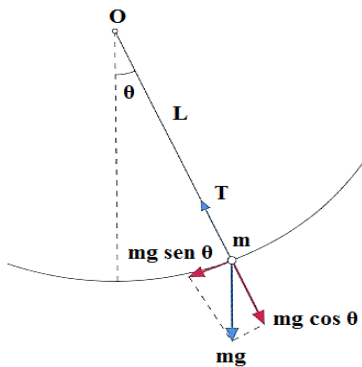
$$a^2 = b^2 + c^2 - 2 * b * c * \cos \theta$$

$$\cos \theta = \frac{b^2 + c^2 - a^2}{2 * b * c}$$



**Ejercicio 2:** Un péndulo simple consta de una pequeña partícula material de masa **m** suspendida de un hilo inextensible de masa despreciable y longitud **L**, que se separa un pequeño ángulo **θ** respecto de su posición vertical de reposo, empezando a oscilar la partícula con un movimiento armónico simple. La ecuación general del periodo de oscilación **T** viene dada por la serie:

$$T = 2\pi \sqrt{\frac{L}{g}} \left( 1 + \frac{1^2}{2^2} \sin^2 \left( \frac{\theta}{2} \right) + \frac{1^2 3^2}{2^2 4^2} \sin^4 \left( \frac{\theta}{2} \right) + \frac{1^2 3^2 5^2}{2^2 4^2 6^2} \sin^6 \left( \frac{\theta}{2} \right) + \dots \right)$$



El periodo de oscilación se puede determinar con suficiente precisión si se considera el suficiente número de términos de la serie anterior.

Para oscilaciones de pequeña amplitud **θ**, el periodo se puede calcular mediante la siguiente expresión:

$$T_2 = 2\pi \sqrt{\frac{L}{g}}$$

El error relativo cometido con esta aproximación viene dado por la expresión:

$$Error\ relativo = \frac{T - T_2}{T_2} = \frac{1^2}{2^2} \sin^2 \left( \frac{\theta}{2} \right) + \frac{1^2 3^2}{2^2 4^2} \sin^4 \left( \frac{\theta}{2} \right) + \frac{1^2 3^2 5^2}{2^2 4^2 6^2} \sin^6 \left( \frac{\theta}{2} \right) + \dots$$

Construir una función en C, lo más eficiente posible, que devuelva el valor en grados de la máxima elongación o amplitud posible **θ** con una precisión dada, de manera que el error relativo cometido en dicha aproximación del periodo de oscilación (calculada con 100 términos de la serie) sea inferior a un valor dado como argumento. Construir una segunda función que presente en pantalla una tabla con los valores de los errores relativos cometidos en las elongaciones que cumplan dicha condición calculados con diferente número de términos (1, 10, 100 y 1000) y con el formato indicado en el ejemplo. Ejemplo:

Error relativo máximo: 0.001 (0.1%)

Precisión: 1.0 (grados)

Elongación máxima: 7.000 grados

Tabla de errores relativos

		Numero de términos de la serie			
Elongacion(Grados)		1	10	100	1000
0.000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
1.000	0.000019038105451	0.000019038921006	0.000019038921006	0.000019038921006	0.000019038921006
2.000	0.000076146622613	0.000076159671566	0.000076159671566	0.000076159671566	0.000076159671566
3.000	0.000171308155678	0.000171374216705	0.000171374216705	0.000171374216705	0.000171374216705
4.000	0.000304493717522	0.000304702506088	0.000304702506088	0.000304702506088	0.000304702506088
5.000	0.000475662738532	0.000476172485987	0.000476172485987	0.000476172485987	0.000476172485987
6.000	0.000684763078966	0.000685820116037	0.000685820116037	0.000685820116037	0.000685820116037
7.000	0.000931731044835	0.000933689390824	0.000933689390824	0.000933689390824	0.000933689390824

**Ejercicio 3:** Construir una función en C que escriba en pantalla un patrón de rayas inclinadas ( $135^\circ$ ) formado por asteriscos (\*), dados como argumentos el número de filas, el n° de columnas y la distancia de separación (en líneas/caracteres) entre dos líneas del patrón (medida vertical/horizontalmente); si esta última distancia fuera negativa, no se dibujaría nada. El patrón comienza en la primera línea de la primera columna. Ejemplos:

	NF=12	NC=30	SEP=1	NF=12	NC=30	SEP=3	NF=12	NC=30	SEP=5
1	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
2	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
3	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
4	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
5	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
6	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
7	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
8	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
9	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
0	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
1	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
2	* * * * * * * * * * * *			* * * * * * * * * * *			* * * * * * * * * * *		
	123456789012345678901234567890			123456789012345678901234567890			123456789012345678901234567890		