



## Examen Enero

Programación 1 (Universidad de Alicante)

# PROGRAMACIÓN 1

Ingeniería Informática

13 de Enero de 2023

## Instrucciones

- Todo el examen debe estar en un único fichero llamado `examen.c`
- Realiza cada ejercicio en un módulo o varios siguiendo las indicaciones del propio ejercicio.
- En el `main()` (ejercicio 5) realiza las llamadas a los módulos correspondientes a los ejercicios que has hecho.
- El examen tiene una duración de 2 horas y 30 minutos.
- El examen se entrega a través de Moodle, pestaña Evaluación.

## EJERCICIO 1. (2+0.5 puntos)

Este ejercicio consta de dos módulos

- a) Escribir un módulo llamado `ej1_proc`, que recibe una cadena de hasta 60 caracteres que contiene exactamente 6 palabras separadas entre sí por un espacio en blanco. El objetivo del módulo es averiguar en que posición de la cadena empieza y termina cada una de las palabras y el tamaño de cada palabra. Por tanto, se deberá construir una matriz con una fila por cada palabra en la que se incluya esta información.
- b) Escribir un módulo llamado `ej1_muestra` que recibe la matriz que construye el módulo anterior y la muestra por pantalla (cada fila en una fila de la pantalla).

Ejemplo:

Entrada: "Ana programa genial y Rudi también"

(que se correspondería con la siguiente cadena):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...	
A	n	a		p	r	o	g	r	a	m	a		g	e	n	i	a	l	y	...

Matriz de salida:

Inicio palabra		
Fin palabra		
Tamaño palabra		
0	2	3
4	11	8
13	18	6
20	20	1
22	25	4
27	34	8

**Importante:** Puedes usar la función `strlen()` para saber la longitud de la cadena. Recuerda incluir la librería `cstring` si usas esta función.

## EJERCICIO 2. (1.25 puntos)

Realizar un módulo llamado `ej2` que vaya pidiendo números al usuario hasta que el usuario introduce el número 0. Para cada número introducido muestra en pantalla el producto de los números que hay entre 1 y ese número. El módulo debe devolver al `main()` (no imprimir en pantalla) la cantidad de números introducidos (sin contar el 0) y la media de los mismos.

Ejemplo:

Introduce un número: 2  
num: 2 prod: 2  
Introduce un número: 5  
num: 5 prod: 120  
Introduce un número: 3  
num: 3 prod: 6  
Introduce un número: 0

**Importante:** En este ejercicio no se pueden usar arrays ni registros.

### EJERCICIO 3. (2 puntos)

El agente secreto Programín tiene que enviar un mensaje codificado y no sabe hacerlo. Escribe un módulo recursivo llamado *ej3* para ayudarlo. Este módulo debe recibir un número y lo devuelve codificado, este módulo no debe imprimir nada en pantalla. Para codificar el número el procedimiento es el siguiente, se descompone el número en cifras y si la cifra es un cero, se transforma en 1, si es múltiplo de 3 su valor se multiplica por 2 y en cualquier otro caso se deja como está. El número codificado es el resultado de multiplicar todas las cifras procesadas como se ha indicado.

Ejemplos:

Número: 2345789                      Codificado: 241920 (obtenido mediante  $2*3*2*4*5*7*8*9*2$ )  
Numero: 103468                      Codificado: 2304 (obtenido mediante  $1*1*3*2*4*6*2*8$ )

**Importante:** Sólo puntuará si se realiza de forma recursiva y no se pueden usar arrays ni registros.

### EJERCICIO 4. (1+1+1.25 puntos)

Se ha realizado una cata de turrónes en el pueblo en la que los vecinos han votado a cada turrón. Diseña las estructuras de datos adecuadas para almacenar los datos de 20 turrónes. De cada turrón se guardará la siguiente información: código (número correspondiente a la posición que ocupa en la estructura de datos) y votos de los distintos vecinos que han participado. El pueblo tiene un censo de 300 vecinos con lo que un turrón como máximo puede recibir 300 votos, pero no todos los vecinos han votado a todos los turrónes.

- Diseña las estructuras de datos necesarias para almacenar la información de los turrónes.
- Implementa un módulo llamado *ej4\_rell* que rellene los datos de los 20 turrónes. Para ello se genera un número aleatorio entre 2 y 300 que indica el número de votos que ha recibido un turrón y a continuación se generan los valores de los votos también de forma aleatoria, el valor de un voto oscila entre 0 y 10.
- Diseña un módulo llamado *ej4\_list* que muestre un listado de todos los turrónes indicando para cada turrón voto más bajo, voto más alto y media de votos que ha recibido. Al final del listado debe aparecer el turrón mejor valorado, es decir, aquel que ha recibido la media más alta de votos y el menos valorado (el que ha recibido la peor media). Ejemplos de dos turrónes:

	Votos	Voto más bajo	Voto más alto	Media
Turrón	6, 8, 10, 10, 5	5	10	7,8
Turrón	9, 0, 2	0	9	3,7

**Importante:** Para generar los números aleatorios puedes utilizar la función `rand()`. Recuerda inicializar la semilla del motor de generación de números aleatorios mediante `srand(time(NULL))` e incluir las librerías `stdlib.h` y `time.h`

### **EJERCICIO 5 (1 punto)**

Escribe el `main()` que llame a todos los ejercicios que hayas implementado. Declara todas las variables que consideres necesarias. Consideraciones:

- Para llamar a los módulos del ejercicio 1, hay que leer una cadena de caracteres desde teclado (usar `cin.getline(cadena, tamaño)` y utilizar dicha cadena.
- En el caso del ejercicio 2 imprimir por pantalla la media y la cantidad de números introducidos.
- Al módulo *ej3* se le llama dentro de una instrucción de escritura con el número 103468 como parámetro.
- Para el ejercicio 4 llamar primero al módulo *ej4rell* y luego al *ej4\_list*.