

Apellido y Nombres	Legajo	Calificación

Todas las funciones pedidas deben ser colocadas en el archivo **primerParcialFunciones.c** y debe estar acompañado por su correspondiente **punto h**.

Además se debe escribir un programa de testeo para cada función requerida. El nombre de esos archivos es **primerParcialNumeroEjercicio.c**.

Implemente una función que realice la validación de una CBU (Clave Bancaria Uniforme). La CBU está formado de la siguiente manera.

0	1	4	0	1	2	5	6	5	5	6	5	4	1	8	5	4	7	6	5	4	3
E_0	E_1	E_2	S_0	S_1	S_2	S_3	D_0	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	D_1

Donde:

- **E0 a E2:** Es el número de la entidad bancaria.
- **S0 a S3:** Es el número de sucursal.
- **D0:** Es el dígito verificador de E y S
- **C0 a C12:** Es el número de cuenta.
- **D0:** Es el dígito verificador de C

Los dígitos verificadores de la clave bancaria única se calculan de la siguiente forma:

$$R_0 = E_0 * 9 + E_1 * 7 + E_2 * 1 + S_0 * 9 + S_1 * 7 + S_2 * 1 + S_3 * 3$$

$$D_0 = 10 - (R_0 \% 10)$$

$$R_1 = C_0 * 9 + C_1 * 7 + C_2 * 1 + C_3 * 3 + C_4 * 9 + C_5 * 7 + C_6 * 1 + C_7 * 3 + C_8 * 9 + C_9 * 7 + C_{10} * 1 + C_{11} * 3 + C_{12} * 9$$

$$D_1 = 10 - (R_1 \% 10)$$

La función tiene el siguiente prototipo:

int cbu_validar (char *dataPtr)

Donde **dataPtr** es el puntero la clave bancaria uniforme a validar terminada en '\0'

Devuelve:

- **-1** Si la CBU pasada no tiene 22 caracteres.
- **-2** Si alguno de los caracteres de la CBU no es un número.
- **-3** Si el dígito verificador D0 no corresponde.
- **-4** Si el dígito verificador D1 no corresponde.