

# ICOM2015 – 1er Parcial

17 de setiembre de 2015

## Notas:

1. Al finalizar, enviar por e-mail los archivos fuente de cada ejercicio con nombre APELLIDO\_NOMBRE\_Ejer\_N.c a [icom@ib.cnea.gov.ar](mailto:icom@ib.cnea.gov.ar)
2. Uso de prácticos: se pueden utilizar los trabajos prácticos propios realizados.
3. Uso de Internet: solo para la consulta de referencias de funciones de C.

## Problema 1. Validación de CBU.

La Clave Bancaria Uniforme (CBU) es un código utilizado en Argentina por los bancos para la identificación de las cuentas de sus clientes. Mientras que cada banco identifica las cuentas con números internos, la CBU identifica una cuenta de manera única en todo el sistema financiero argentino. Está compuesta por 22 dígitos, separados en dos bloques (de 8 y 14 dígitos respectivamente). Por ejemplo:

**28505909 40090418135201**

En el primer bloque, los primeros 3 dígitos corresponden al Banco. Luego le sigue el primer "dígito verificador". Los siguientes 3 dígitos identifican la sucursal y por último el segundo dígito verificador.

El segundo bloque se compone de 13 dígitos para el número de cuenta y el tercer y último dígito verificador.

En el ejemplo dado sería:

Banco: <b>285</b> ( $B_0=2, B_1=8, B_2=5$ )	$DV_0 = 0$
Sucursal: <b>590</b> ( $S_0=5, S_1=9, S_2=0$ )	$DV_1 = 9$
Cuenta: <b>4009041813520</b> ( $C_0=4, C_1=0, C_2=0, \dots$ )	$DV_2 = 1$

El procedimiento para validar una clave CBU es el siguiente:

- Validación del primer bloque: Se deben sumar los dígitos correspondientes al Banco,  $DV_0$ , y Sucursal, con los siguientes pesos: (7, 1, 3, 9, 7, 1, 3). En el ejemplo sería  $B_0*7 + B_1*1 + B_2*3 + DV_0*9 + S_0*7 + S_1*1 + S_2*3=81$ . El último dígito de esta suma se resta de 10 (En el ejemplo:  $DIF_1=10-1=9$ ) y este valor debe coincidir con el  $DV_1$  (En el ej.  $DV_1=9=DIF_1$ ).
- Validación del segundo bloque: Sumar los dígitos del número de cuenta, con los siguientes pesos: (3, 9, 7, 1, 3, 9, 7, 1, 3, 9, 7, 1, 3). En el ejemplo sería  $(C_0*3 + C_1*9 + C_2*7 + C_3*1 + C_4*3 + C_5*9 + C_6*7 + C_7*1 + C_8*3 + C_9*9 + C_{10}*7 + C_{11}*1 + C_{12}*3=139)$ . El último dígito de esta suma se resta de 10 (En el ejemplo:  $DIF_2=10-9=1$ ) y este valor debe coincidir con el  $DV_2$  (En el ej.  $DV_2=1=DIF_2$ ).

Implemente una función que reciba una CBU en forma de string, y lo verifique. Debe retornar verdadero o falso. Respete el siguiente prototipo:

```
int cbu_check(char cbu[]);
```

Implemente un programa que solicite al usuario un número de CBU, lo verifique e imprima por pantalla si el número es válido o no. Implemente la función de forma que resulte sencillo cambiar los pesos que se le da a cada dígito.

## Problema 2. Resonancia Magnética.

Es una técnica de diagnóstico por imagen en la que se obtiene información de las características de diferentes puntos del cuerpo a partir de su respuesta a un campo magnético variable.

Se puede suponer que el cuerpo a escanear se puede representar como una matriz tridimensional de puntos que representa una imagen adquirida por este método.

Cada punto tiene un valor almacenado en 1 byte sin signo (valores entre 0 y 255), correspondiente a un distinto nivel de gris. Suponga que imagen es un cubo de  $100 * 100 * 100$  puntos (**IMG\_SZ = 100**).

Se desea escribir un programa que permita detectar aspectos en la imagen que pueden ser indicadores de una enfermedad.

Se solicita implementar las siguientes funciones:

- a. **puntoSospechoso**: Función que dado el cuerpo (imagen) y un punto a través de sus coordenadas x, y, z detecte si es sospechoso, considerando sospechoso aquellos puntos para los que el punto y todos los puntos adyacentes directos (8 puntos vecinos del propio plano, los 9 del plano superior y los 9 del plano inferior) tienen valores en el intervalo [25, 50]. La función debe retornar 1 si el punto es sospechoso, o 0 en caso contrario. Suponga que los puntos de la superficie externa del cuerpo no son analizados. El prototipo (respetar) deberá ser:

```
int puntoSospechoso(unsigned char img[IMG_SZ][IMG_SZ][IMG_SZ], int x, int y, int z);
```

- b. **planoSospechoso**: Función que dado el cuerpo y la coordenada z de un plano (paralelo al plano x,y), detecte si el plano presenta un porcentaje mayor al 20% de puntos sospechosos. Prototipo:

```
int planoSospechoso(unsigned char img[IMG_SZ][IMG_SZ][IMG_SZ], int z);
```

- c. Implemente un programa que lea una imagen desde un archivo y la procese imprimiendo los planos sospechosos de la misma. El archivo tiene una línea por punto de la imagen, teniendo cada línea las coordenadas (x y z) y el valor del punto. Ejemplo:

```
0 0 0 34
0 0 1 28
0 0 2 130
...
0 0 99 211
0 1 0 21
0 1 1 219
...
99 99 99 132
```

### ***Problema 3. Paradoja de Parrondo.***

Consideremos un par de juegos de azar, en los que en cada juego un jugador puede ganar o perder un peso con cierta probabilidad.

En el primero de los juegos, llamémoslo A, el jugador gana un peso con probabilidad 49,5% y pierde con probabilidad 50,5%. Este juego, debido al pequeño sesgo que separa las probabilidades del 50%, es un juego perdedor. El jugador, en promedio, pierde de forma sistemática. Un juego ganador es aquel en el que la ganancia media a lo largo de varias jugadas es positiva.

En el segundo juego, B, las probabilidades dependen de lo que el jugador haya acumulado hasta el momento (llamaremos a esa cantidad el capital). Si el capital (ya sea positivo o negativo) es múltiplo de 3, entonces gana con probabilidad 9,5% y pierde con probabilidad 90,5%. Si el capital no es múltiplo de 3, la probabilidad de ganar es 74,5% y la de perder 25,5%.

Como se ve, el juego B es en ocasiones bastante favorable, pero en otras (cuando el capital es múltiplo de 3) es muy desfavorable. En promedio el juego es perdedor. Ambos juegos son perdedores.

Sin embargo, si se alternan los juegos de manera aleatoria el jugador en promedio gana. Y no es ésta la única secuencia que produce ganancias. Jugando A dos turnos, seguidos de dos turnos de B, o tres turnos de A seguidos de dos turnos de B la ganancia es aún mayor. Analizando la ganancia media de 500 jugadores a lo largo de 100 turnos, mostrar que:

- a. A es un juego perdedor
- b. B es un juego perdedor
- c. La alternancia aleatoria entre A y B da un juego ganador