

## FUNDAMENTOS DE COMPUTACIÓN - PRIMERA EVALUACIÓN - JULIO 3, 2012

### Una solución – Luis Rodríguez O.

**TEMA 1 (20 PUNTOS)** Para cierta aplicación informática se necesitan codificar números enteros.

Elabore un ALGORITMO que solicite al usuario un número entero positivo de tres dígitos, el cual necesita ser codificado. Considere que:

- Si el dígito de dicho número es 2, 5 o 7, se le debe sumar la unidad.
- Si el dígito es 1, 4, 8 o 9, se le resta la unidad.
- Los dígitos restantes no se alteran.

*Rúbrica: Validación (5 puntos), conversión por dígito (10 puntos), número codificado(5 puntos)*

Ejemplos:

Original		Codificado
472	←	383
503	←	603
615	←	606

(Una solución usando un vector. También puede resolverse sin usar el vector)

```
n=input('Ingrese un número de tres dígitos ');
d(3)=mod(n,10);           % Unidad
n=fix(n/10);
d(2)=mod(n,10);          % Decena
d(1)=fix(n/10);          % Centena
for i=1:3
    switch d(i)
        case {2, 5, 7}, d(i)=d(i)+1;
        case {1, 4, 8, 9}, d(i)=d(i)-1;
    end
end
x=d(1)*100+d(2)*10+d(3);
disp(x)
```

**TEMA 2 (30 PUNTOS)** "Carreras" es un juego de tablero para dos jugadores. En cada turno el jugador lanza dos dados y se usan los números obtenidos en las caras superiores. Para iniciar el juego, el jugador debe obtener las mismas caras de los dados en el lanzamiento. Para avanzar casillas, se usa la suma de las caras de los dados, con el objetivo de llegar a la casilla final del tablero numeradas desde 1 al 50. Existen casillas de premio (2, 17, 30, 42), en donde el jugador gana un lanzamiento adicional. Elabore un ALGORITMO que simule este juego y muestre cuál jugador ganó.



*Rúbrica: Control de jugadores y casillas (5 puntos). Aleatorios en reglas (10 puntos). Control de premios (10 puntos). Algoritmo estructurado (5 puntos)*

```

c1=0;                                % Conteo de casillas jugador 1
c2=0;                                % Conteo de casillas jugador 2
inicio1=0;                            % Señal para inicio jugador 1
inicio2=0;                            % Señal para inicio jugador 2
while c1<50 & c2<50
    if inicio1==0
        a=fix(6*rand)+1;
        b=fix(6*rand)+1;
        if a==b                        % Jugador 1 puede jugar
            inicio1=1;
        end
    end
    if inicio2==0
        a=fix(6*rand)+1;
        b=fix(6*rand)+1;
        if a==b                        % Jugador 2 puede jugar
            inicio2=1;
        end
    end
    if inicio1==1
        a=fix(6*rand)+1;
        b=fix(6*rand)+1;
        c1=c1+a+b;
        if c1==2|c1==17|c1==30|c1==42  % Jugada adicional para 1
            a=fix(6*rand)+1;
            b=fix(6*rand)+1;
            c1=c1+a+b;
        end
    end
    if inicio2==1
        a=fix(6*rand)+1;
        b=fix(6*rand)+1;
        c2=c2+a+b;
        if c2==2|c2==17|c2==30|c2==42  % Jugada adicional para 2
            a=fix(6*rand)+1;
            b=fix(6*rand)+1;
            c2=c2+a+b;
        end
    end
end
end
if c1>=50
    disp('Gana jugador 1');
else
    disp('Gana jugador 2');
end

```

**Tema 3** (20 puntos) En el control de INVENTARIO DE PRODUCTOS que se lleva en una bodega, se tiene un modelo donde se determina la cantidad máxima y mínima de stock por producto. Considerando el siguiente modelo:

Escriba un ALGORITMO que permita:

a) Registrar los datos de Consumo Máximo, Consumo Mínimo, Existencia actual y Tiempo de reposición de inventario para un listado de N productos.

b) Luego aplicando el modelo mostrado, determine la Cantidad de Pedido (CP) para cada producto.

c) Muestre aquellos productos donde la cantidad de pedido (CP) supere en un 70% la existencia actual.

*Rúbrica: Ingreso de datos en arreglos (5 puntos), calculo de pedidos (8 puntos), Salida (7 puntos)*

$$E_{max} = (C_{max} * Tr) + E_{min}$$

$$E_{min} = C_{min} * Tr \quad CP = E_{max} - E$$

Donde: Tr = Tiempo de reposición de inventario (en días)  
 Cmax = Consumo máximo (unidades diarias)  
 Cmin = Consumo mínimo (unidades diarias)  
 Emax = Existencia máxima  
 Emin = Existencia mínima (o de seguridad)  
 CP = Cantidad de pedido  
 E = Existencia actual



```

cmax=input('Vector de consumo máximo ');
cmin=input('Vector de consumo mínimo ');
e=input('Vector de existencia actual ');
tr=input('Vector de tiempos de reposición ');
n=length(cmax);
for i=1:n
    emin=cmin(i)*tr(i);
    emax=cmax(i)*tr(i)+emin;
    cp=emax-e(i);
    disp('Producto y cantidad de pedido');
    disp([i, cp]);
    if cp>0.7*e(i)
        disp('El producto tiene poca existencia actual');
    end
end
    
```

**Tema 4** (30 puntos). Una vez terminado el invierno, el subsidio de la tarifa eléctrica residencial cambia para los clientes residenciales en la costa. La tarifa se establece acorde a los consumos en pliego tarifario mostrado.

Realice un algoritmo que permita:

- Ingresar los consumos de un mes para n clientes residenciales,
- Calcular el valor a facturar y el incremento para cada cliente en verano,
- Mostrar el total facturado en el mes y
- ¿Cuál es el cliente que más se le ha facturado? (suponga que solo existe uno).

Fuente: "Terminado el invierno ajustan tarifas eléctricas". El Universo 09.06.2012.

Nota: Pliego tarifario resumido. Usar arreglos solo para tablas de clientes.

Rúbrica: literal b y d (10 puntos), literal a y c (5 puntos).

#### Tarifario Eléctrico

Consumo entre (KWh)	Invernal (\$)	Verano (\$)	cambio (\$)
<130	0.04	0.04	0.00
130 a 500	0.08	0.11	0.03
500 a 700	0.11	0.13	0.02
Superior a 700	0.16	0.26	0.10

#### Listado de consumo para Clientes Residenciales

Cliente	Consumo Verano (KWh)	Facturado Verano (\$)	Incremento (\$)
1	200	22.00	6.00
2	600	78.00	12.00
3	400	44.00	12.00
4	800	208.00	80.00
...			

Total facturado: \$ 360.00 Cliente mayor factura: 4

```

c=input('Ingrese el vector de consumos del mes ');
n=length(c);
fv=[]; % Vector de valores facturados a clientes
t=0; % Total facturado en el mes
for i=1:n % Rangos para elegir tarifas de verano
    if c(i)<130
        fc=c(i)*0.04;
        ic=c(i)*0;
    else
        if c(i)>=130 & c(i)<500
            fc=c(i)*0.11;
            ic=c(i)*0.03;
        else
            if c(i)>=500 & c(i)<700
                fc=c(i)*0.13;
                ic=c(i)*0.02;
            else
                fc=c(i)*0.26;
                ic=c(i)*0.10;
            end
        end
    end
    disp('Cliente Valor facturado Incremento');
    disp([i, fc, ic]);
    t=t+fc;
    fv=[fv, fc];
end
disp('Total facturado en el mes');
disp(t);
[m,p]=max(fv); % Cliente con mayor valor facturado
disp('Cliente con mayor valor facturado')
disp(p);

```