

Tema 2 (30 puntos) Dice la historia que la criptografía fue utilizada por los gobiernos para comunicaciones secretas durante las campañas militares. En el siglo I A.C., Julio César usó un algoritmo que consistía en desplazar tres espacios hacia la derecha las letras del texto siguiendo el orden alfabético.

Realice una función **cifradocesar (mensaje, llave)** que usa una variante del método descrito para cifrar el mensaje, desplazando cada letra del mensaje en el alfabeto las posiciones que indique la "llave". Con el resultado, el mensaje original no es reconocido a menos que se use el valor negativo de la "llave".

Considerar que siguiente posición de la letra "Z" será la "A", y la anterior a la letra "A" será la "Z".

Ejemplo:

```
mensaje='HOLA' , llave=+3
```

```
>> cifradocesar(mensaje,3)      ans=KROD
```

```
mensaje='KROD' , llave =-3;
```

```
>> cifradocesar(mensaje,-3)     ans=HOLA
```

Letras=

A	...	H	I	J	K	L	M	N	O	P	Q	R	...	Z
---	-----	---	---	---	---	---	---	---	---	---	---	---	-----	---

Referencia: Cifrado César, http://es.wikipedia.org/wiki/Cifrado_C%C3%A9sar

Rubrica: definición de función (5 puntos), posición letra en alfabeto (7 puntos), desplazamiento (10 puntos), desplazamiento negativo (8 puntos)

```
function r=cifradocesar(x, k)
a='abcdefghijklmnopqrstuvwxyz';
n=length(x);
r="";
for i=1:n
[t,p]=ismember(x(i),a);
p=p+k;
if p>26
p=p-26;
end
if p<1
p=26+p;
end
r=[r,a(p)];
end
```

Tema 3 (20 puntos)

- a) Escriba una función **contando(vector,n)**, que reciba un **vector** de números enteros y cuente el número de veces que se encuentra cada número en el rango entre 1 y n.
Los números fuera de rango se descartan del conteo.

i	vector(i)
1	2
2	4
3	5
4	6
5	1
6	2
7	4
m=8	5

Ejemplo:
>>contando(vector,6)

k	cuenta(k)
1	1
2	2
3	0
4	2
5	2
n=6	1

- b) Realice una función **unicorango(vector,n)**, que muestre si los números en el **vector** no son repetidos y se encuentran en el rango entre 1 y n.

Ejemplos:

vector= 1 2 4 5
>>unicorango(vector,6) ans=1

vector= 1 4 4 7
>>unicorango(vector,6) ans=0

Rubrica: Definición de funciones (5 puntos), contadores (5 puntos), validación de únicos (5 puntos), validación de rango (5 puntos).

```
a) function c=contando(v,n)
c=zeros(1,n);
m=length(v);
for i=1:m
    k=v(i);
    if k>=1 & k<=n
        c(k)=c(k)+1;
    end
end
```

```
b) function r=unicorango(v,n)
m=length(v);
t=[]; %El vector t contendrá sólo elementos diferentes
for i=1:m
    if ismember(v(i),t)==0 & v(i)>=1 & v(i)<=n
        t=[t, v(i)];
    end
end
if length(t)==m
    r=1;
else
    r=0;
end
```

Handwritten notes:
 function r=unicorango(v,n)
 m=length(v);
 t=[];
 for i=1:m
 if ismember(v(i),t)==0 & v(i)>=1 & v(i)<=n
 t=[t, v(i)];
 end
 end
 if length(t)==m
 r=1;
 else
 r=0;
 end

Tema 4 (30 puntos) La Facultad de Ciencias Naturales y Matemáticas (FCNM) se conforma de tres departamentos:

Física, Química y Matemáticas. Para elegir el Consejo Directivo de la FCNM se requieren seleccionar **cuatro** profesores entre los candidatos por votación en base a las siguientes reglas:

- El candidato(a) que obtenga la **mayor cantidad de votos**.
- **Un candidato(a) de cada uno de los tres departamentos** que obtenga la mayor cantidad de votos.

Existen dos candidatos por cada departamento identificados por un número entero (ver tabla ejemplo b).

Escriba un **programa** para la selección del Consejo Directivo siguiendo los procesos de votación y selección.

a) Proceso de Votación: Para cada votante se debe realizar

a.1. Ingreso y validación de papeleta.- Se llena un vector *papeleta* con los números de **4** candidatos. Se valida que cada voto en la papeleta sea único y que correspondan a un candidato, sino se la descarta y se llena de nuevo.

Puede usar la función *unicorango()* del tema anterior.

a.2. Registro del voto.- la papeleta valida se añadirá al vector que contiene todos los votos del proceso.

Ejemplo a.1:
 >> unicorango(papeleta,6)
 ans=1

si la papeleta es:

i	papeleta[i]
1	1
2	2
3	4
4	5

a.2 Registra voto

i	voto[i]
1	2
2	4
3	5
4	6
5	1
6	2
7	4
8	5
...	...

b) Proceso de selección:

Consiste en realizar el conteo de votos por candidato, usando el vector *voto*.

Puede usar la función *contando()* del tema anterior.

Luego muestre los seleccionados conforme a las reglas especificadas.

Ejemplo b):

Departamento	Candidato	Conteo	Más votos	Selección/ Departam.
Física	1	25		1
	2	29	2	
Química	3	23		3
	4	2		
Matemáticas	5	24		5
	6	13		

>> candidatos seleccionados: 1, 2, 3, 5

c) Muestre los seleccionados.

Nota: El tema no considera la posibilidad de votos blancos o nulos. Para declarar un arreglo vacío: *voto=[]*;

Referencia: Convocatoria elecciones FCNM-ESPOL-2013.

www.espol.edu.ec/tribunal/

Rubrica: literal a.1 (5 puntos), literal a.2 (10 puntos), uso correcto de las funciones (5 puntos), literal c. (10 puntos)

```

n=input('Cantidad de votantes ');
v=[ ];
for i=1:n
    r=0;
    while r==0
        p=input('Ingrese papeleta con 4 votos ');
        r=unicorango(p,6);           %Verificar papeleta válida
    end
    v=[v, p];                       %Vector de votos
end
c=contando(v,6);                    %Conteo de votos
[x,mv]=max(c);                      %Candidato más votado
t=0;
for i=1:6
    if x==c(i)
        t=t+1;
    end
end
if t==1
    disp('Candidato más votado');
    disp(mv);
else
    disp('Empate en el más votado');
end
if c(1)>c(2)                          %Física
    disp('Física: Gana 1');
else
    if c(2)>c(1)
        disp('Física: Gana 2');
    else
        disp('Empate en Física');
    end
end
if c(3)>c(4)                          %Química
    disp('Química: Gana 3');
else
    if c(4)>c(3)
        disp('Química: Gana 4');
    else
        disp('Empate en Química');
    end
end
if c(5)>c(6)                          %Matemáticas
    disp('Matemáticas: Gana 5');
else
    if c(6)>c(5)
        disp('Matemáticas: Gana 6');
    else
        disp('Empate en Matemáticas');
    end
end
end

```



INSTITUTO DE CIENCIAS MATEMÁTICAS
 ICM00794 - FUNDAMENTOS DE COMPUTACIÓN
 2da Evaluación I Término 2012-2013
 Agosto 28, 2012



Paralelo: Matricula: Nombre:

Firma:

TODOS LOS TEMAS SE PRESENTAN EN MATLAB

Tema 1 (20 puntos) a) Escriba las funciones denominadas **band(A,B)**, que devuelve resultado "**A ∧ B**" entre 2 variables enteras, y otra función **bor(A,B)** que devuelve el "**A ∨ B**" entre 2 variables enteras.

Ambas funciones retornan 1 si es verdadero y 0 si es falso, deben validar los datos de A y B, de existir un error devuelven -1.

b) Escriba un programa que solicite valores para las variables A, B y C, usando las funciones anteriores evalúe las siguientes ecuaciones:

$X = (A \vee B) \wedge (A \wedge C)$
 $Y = (A \wedge B) \vee (B \wedge C)$
 $Z = A \vee (A \wedge B)$
 $W = C \wedge (A \vee B)$

Rúbrica: funciones band (5 puntos), bor (5 puntos), Programa estructurado y uso de funciones creadas en operaciones (10 puntos)

```
function r=band(a,b)
if (a==0|a==1)&(b==0|b==1)
    r=a&b;
else
    r=-1;
end
```

```
function r=bor(a,b)
if (a==0|a==1)&(b==0|b==1)
    r=a|b;
else
    r=-1;
end
```

```
a=input('a ');
b=input('b ');
c=input('c ');
x=band(bor(a,b),band(a,c));
y=bor(band(a,b),band(b,c));
z=bor(a,band(a,b));
w=band(c,bor(a,b));
disp([x,y,z,w]);
```

blog.espol.edu.ec/educasanas/

Tema 2. (25 puntos) El cifrado musical americano deriva de la notación griega que nombraba las notas desde la letra alfa hasta la gamma, siendo alfa la nota "la" actual y gamma la nota "sol" actual, tal como se muestra en el teclado de la figura.

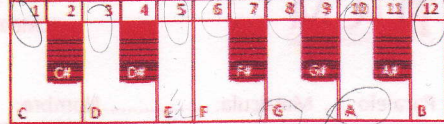
Realice una función `teclanum(canción)`, que permita recibir una **canción** en cifrado americano mediante una cadena de caracteres, y la transforme en un arreglo que represente el número de la tecla a ser usada en secuencia.

Nota: Considere un arreglo de letras y símbolos ordenados por posición, al implementar # observe que le añade una posición (+1) a la tecla anterior.

Referencia: http://es.wikipedia.org/wiki/Cifrado_ingl%C3%A9s, http://www.bgfl.org/custom/resourcesftp/client_files/2/musical_notation/index.htm

Rúbrica: Definición de función (5 puntos), selección de número de tecla (10 puntos), determinar tecla con dos caracteres (20 puntos)

Teclado



```
>>canción = 'EFGGFEDCCDEED'
>>teclanum(canción)
ans = 5 6 8 8 6 5 3 1 1 3 5 5 3
```

function t=teclanum(x)

```
j=0;
for i=1:length(x)
    j=j+1;
    switch x(i)
        case 'C', t(j)=1;
        case 'D', t(j)=3;
        case 'E', t(j)=5;
        case 'F', t(j)=6;
        case 'G', t(j)=8;
        case 'A', t(j)=10;
        case 'B', t(j)=12;
        otherwise
            if x(i)=='#'
                j=j-1;
                t(j)=t(j)+1;
            end
        end
    end
end
```

(opcional)

function r=teclanum(x)

```
r=[];
a=[1,3,5,6,8,10,12];
b=[c,d,e,f,g,a,b];
n=length(x);
```

```
for i=1:n
```

```
[j,k]=ismember(x(i),b)
if j==1
    r=[r,a(k)];
```

```
else
    if x(i)=='#'
        r(i)=r(i-1)+1;
    end
```



Tema 3. (25 puntos). En los contextos de fotografía digital, una imagen puede ser representada en una escala de grises, corresponden a un conjunto de colores en tonalidades entre el blanco y negro. Se emplean 8 bits para representar cada píxel lo que sólo permite una escala con 256 intensidades o escalas de gris [0,255].

a) Realice una función `totaltinta(matriz)` que para una imagen representada en una `matriz` de `nxm`, muestre el equivalente numérico de las unidades de tinta a consumir en la imagen.

Suponga que en una impresora de inyección de tinta, el consumo de tinta corresponde al número escrito en la casilla de la `matriz` escala de gris.

b) Realice una función `imnegativa(matriz)` que cambie la imagen a negativo, invirtiendo los valores en la escala [0,255] que contiene cada píxel (casilla).

Tal como se muestra en el ejemplo.

Referencia: es.wikipedia.org/wiki/Escala_de_grises

Rúbrica: definición de funciones (5 puntos), literal a (10 puntos), literal b (10 puntos)

matriz				
50	30	30	0	0
30	30	30	0	0
		30	30	0
			30	30
50	30	30	0	0
>>totaltinta(matriz) ans= 1550				
>>imnegativa(matriz)				
225	225	225	255	255
225	225	225	255	255
85	85	225	225	255
85	85	85	225	225
225	225	225	85	85

```
function t=totaltinta(e)
```

```
[n,m]=size(e);
```

```
t=0;
```

```
for i=1:n
```

```
    for j=1:m
```

```
        t=t+e(i,j);
```

```
    end
```

```
end
```

```
function f=imnegativa(e)
```

```
[n,m]=size(e);
```

```
for i=1:n
```

```
    for j=1:m
```

```
        f(i,j)=255-e(i,j);
```

```
    end
```

```
end
```

Tema 4. (30 puntos) El Consejo Nacional Electoral requiere de un programa que permita administrar los datos de registros de los afiliados a partidos y movimientos políticos, presentados al inicio para un proceso electoral.

Los datos disponibles se encuentran listados en dos tablas de datos como las siguientes:

Cédula	Nombre
0123456789	MARÍA RODRÍGUEZ
0234567891	JUAN PÉREZ
0345678923	JOSÉ PARRA
...	...

Cédula	Nombre	Partido	Estado
0234567891	JUAN PÉREZ	7	1
0123456789	MARÍA RODRÍGUEZ	9	3
0123456789	MARÍA RODRÍGUEZ	4	3
...

El programa debe permitir el ingreso de las tablas (1 y 2), luego validar y registrar el **Estado** final conforme a los criterios mostrados en la tabla 3. Para revisar cada registro por cada criterio, solo se validaran los registros con **Estado "1"**.

Si luego de revisar y validar, el **Estado** del registro se mantiene en "1", éste se considera válido.

Al final presente una tabla de **"Resultado"** que liste cuántos registros por estado tiene cada partido.

Criterios Tabla 3.

Estado	Descripción
1	Aceptado al inicio como válido, sin revisar "errores"
2	No empadronado.- La cédula de la ficha no se encuentra en el padrón
3	Afiliación duplicada.- El registro de cédula en la ficha se encuentra más de una vez
4	Nombres errados.- por no coincidir entre la ficha y el padrón

Nota: para comparar cadenas de caracteres, puede usar la función strcmp(cadenaA,cadenaB)

Referencia: "El sistema del CNE permitió la confusión de identidades", 6 -Agosto-2012, www.expreso.ec
Rúbrica. Ingreso con estructura de datos (5 puntos), validar empadronados (5 puntos), validar duplicados (5 puntos), validar nombres (5 puntos), Contador de registros por estado (5 puntos), Algoritmo estructurado (5 puntos)

Resultado	Estado			
	1	2	3	4

```

p=[];
n=input('Cantidad de registros en el Padrón Electoral ');
for i=1:n
    pr(i).ced=input('Cédula ');
    pr(i).nom=input('Nombre ','s');
end
f=[];
m=input('Cantidad de Fichas de Afiliación ');
for j=1:m
    fr(j).ced=input('Cédula ');
    fr(j).nom=input('Nombre ','s');
    fr(j).par=input('Partido ');
    fr(j).est=1;
end

for i=1:m
    t=0;
    for j=1:n
        if fr(i).ced==pr(j).ced
            t=t+1;
            if strcmp(fr(i).nom,pr(j).nom)==0
                fr(i).est=4;
            end
        end
    end
    if t==0
        fr(i).est=2;
    else
        if t>1
            fr(i).est=3;
        end
    end
end

```

%Lectura del Padrón electoral

%Lectura de fichas de afiliac.

%Registro inicialmente válido

%Revisión de registros

%Nombres no coinciden

%Registro no empadronado

%Registro duplicado

%Lectura de la lista de partidos y conteo de registros en una matriz

```
listapart=input('Ingrese vector con la lista de partidos ');
npart=length(listapart);
matrizcont=zeros(npart,4);
for i=1:m
    numpart=fr(i).par;
    estado=fr(i).est;
    for j=1:npart
        if numpart==listapart(j)
            matrizcont(j,estado)=matrizcont(j,estado)+1;
        end
    end
end
disp(matrizcont)
```

Matricula: _____ Nombre: _____ Paralelo: _____ Firma: _____

Tema 1 (15 puntos). Realice una función **enlista(número,vector)** que reciba un número entero y un vector que contiene una lista de números, revise si el número se encuentra en el vector. Las respuestas serán: si esta en lista (1), si no está en lista (0).

Ejemplo:

1	456
2	654
3	234
4	987
5	876

```
>> enlista(789,vector)
ans= 0
>> enlista(234,vector)
ans= 1
```

blog.espol.edu.ec/educasanas/

Rúbrica: Definición de función (5 puntos), revisión (5 puntos), respuestas (5 puntos).

```
function e=enlista(n,v)
k=length(v);
e=0;
for i=1:k
    if n==v(i)
        e=1;
    return;
end
end
```

```
function e=enlista(n,v)
n=length(v)
for i=1:n
    if ismember(n,v(i))==0
        e=0;
    else
        e=1;
    end
end
```

Tema 2 (35 puntos). Al finalizar la construcción del Cuarto Puente, el Ministerio de Transporte y Obras Públicas nombrará al complejo vial mediante votaciones electrónicas entre cuatro nombres. Realice un programa que permita administrar las votaciones mediante el siguiente menú:

- **Ingresar y validar voto.** - se registra cédula, nombre y voto. Se descarta el registro si el voto es repetido; puede validar usando la función del tema anterior.
- **Muestra registrados y descartados.** - Presenta los contadores de votos registrados y descartados por reintento del votante.
- **Resultado de votación.** - Muestra los resultados de la votación y el ganador.
- **Salir.**

Ejemplo de Nombres:

1. Unidad Nacional
2. Carlos Pérez Perazo
3. Rafael Mendoza Avilés
4. Otro por presentar.....

Sugerencia: Pedir los datos y registrarlos solamente si vota por primera vez. Copiar las cédulas válidas en un vector y usar la función `enlista()` del tema anterior para validar las cédulas repetidas. "Un concurso eliminará los nombres de tres viaductos" www.eluniverso.com – 30.Ago.2011

Rubrica: Menú (5 puntos). Definición y uso de estructura de datos (5 puntos). Validación de votos y uso de función (10 puntos).

Mostrar registrados y descartados (5 puntos). Resultados de votación (10 puntos)

```

p=[];
x=0;
nr=0;
nd=0;
while x~=4
    disp('1) Ingresar y validar voto');
    disp('2) Muestra registrados y descartados');
    disp('3) Resultado votacion');
    disp('4) Salir');
    x=input('Elija una opción ');
    switch x
        case 1
            r.ced=input('Cédula ');
            r.nom=input('Nombre ');
            r.voto=input('Voto ');
            e=validar(r.ced,p);
            if e==0
                p=[p,r];
                nr=nr+1;
            else
                disp('Voto repetido ')
                nd=nd+1;
            end
        case 2
            disp('Votos registrados');
            disp(nr);
    
```



```
disp('Votos descartados');  
disp(nd);  
case 3  
c=[0, 0, 0, 0, 0];  
for i=1:length(p)  
v=p(i).voto;  
if v>0 & v<5  
c(v)=c(v)+1;  
else  
c(5)=c(5)+1;  
end  
end  
[m,g]=max(c(1:4));  
disp('Ganador');  
disp(g);  
disp('Votos nulos')  
disp(c(5));  
end  
end
```

```
function e=validar(k,p)  
n=length(p);  
e=0;  
for i=1:n  
if k==p(i).ced  
e=1;  
return;  
end  
end
```

Tema 3 (25 puntos). Llamemos cuadrado "semi-mágico" a una matriz cuadrada conteniendo números de tal manera que cada suma parcial de la primera fila, última fila, primera columna, última columna y cada una de las dos diagonales, producen el mismo resultado.

Escriba un programa que solicite: el tamaño n del cuadrado y el máximo de intentos a realizar, para llenar aleatoriamente una matriz de $n \times n$ con enteros positivos de una cifra, hasta que la matriz sea un cuadrado "semi-mágico".

Ejemplo:

1	3	6	2
7	4	1	4
1	6	4	3
3	4	2	3

Muestre la matriz resultante y la cantidad de intentos realizados, si se logró el objetivo.

Rúbrica: generación de matriz (5 puntos), determinar si es semi-mágico (15 puntos), control de intentos y resultados (5 puntos)

```

n=input('Dimensión del cuadrado ');
m=input('Máximo de intentos ');
magico=0;
i=0;
while magico==0 & i<m
    i=i+1;
    a=fix(rand(n,n)*10+1);
    d1=0;
    d2=0;
    f1=0;
    fn=0;
    c1=0;
    cn=0;
    for i=1:n
        d1=d1+a(i,i);
        d2=d2+a(i,n-i+1);
        f1=f1+a(1,i);
        fn=fn+a(n,i);
        c1=c1+a(i,1);
        cn=cn+a(i,n);
    end
    if d1==d2 & d1==f1 & d1==fn & d1==c1 & d1==cn
        magico=1;
    end
end
if magico==1
    disp(a);
end
disp('Número de intentos ');
disp(i);
    
```



Tema 4 (25 puntos). Realice un programa que reciba una cadena de caracteres, que representa un número romano y la convierta a número en base decimal.

El equivalente de números romanos se muestra en la tabla, y para la conversión considere solo las siguientes reglas:

- Si a la derecha de una cifra romana se escribe otra igual o menor, el valor de ésta se suma a la anterior.

- Si entre dos cifras romanas cualesquiera existe otra menor, ésta restará su valor a la siguiente.

Casos para I, X y C

- En ningún número se puede poner una misma letra más de tres veces seguidas.

Suponga que la cadena de caracteres corresponde a un número romano válido.

Tabla de equivalentes

I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Ejemplos: Cadena= CLXIII C L X I I I

Decimal=163 +100 +50 +10 +1 +1 +1

Cadena=CXLIX C X L I X

Decimal=149 +100 -10 +50 -1 +10

Rúbrica: cálculo de equivalencias aditivas (10 puntos), equivalencias con signo menos (15 puntos)

```
function d=rom2dec(r)
```

```
%Convierte la cadena de números romanos a un vector  
equivalente decimal
```

```
d=[];
```

```
for i=1:length(r)
```

```
    switch r(i)
```

```
        case 'I', d=[d,1];
```

```
        case 'V', d=[d,5];
```

```
        case 'X', d=[d,10];
```

```
        case 'L', d=[d,50];
```

```
        case 'C', d=[d,100];
```

```
        case 'D', d=[d, 500];
```

```
        case 'M', d=[d,1000];
```

```
    end
```

```
end
```

```
r=input('ingrese número romano ');
```

```
d=rom2dec(r);
```

```
n=0;
```

```
while length(d)>1
```

```
    n=n+d(1);
```

```
    if d(1)>=d(2)
```

```
        if length(d)>2 & d(3)>d(2)
```