



Matricula: Nombre: Firma:

Tema 1. (20 puntos). Una persona que deposita C dólares en una cuenta de ahorro, el banco le paga una tasa de interés anual r , luego de n años tendrá un valor acumulado de A dólares. La siguiente fórmula relaciona estos valores:

$$A = C(1+r)^n$$

Juan y Pedro abren cuentas de Ahorro en diferentes Bancos.

En un Banco X, Juan deposita en una cuenta de ahorro $C=200$ que paga un interés anual de $r=0.08$.

En un Banco Y, Pedro deposita en otra cuenta de ahorro $C=300$ que paga un interés anual de $r=0.05$

Escriba un algoritmo que solicite los datos para las cuentas de Juan y Pedro, determine el año n cuando la cantidad acumulada A de Juan superará a la cantidad acumulada A de Pedro.

Nota: No se considerarán depósitos o retiros entre los años. Suponga que Juan deposita menos que Pedro.

Rubrica: Ingreso de datos (5 puntos), determinación de saldos individuales por año (5puntos), Respuesta solicitada (10 puntos)

```

cj=input('depósito de Juan ');
rj=input('tasa de interés de Juan ');
cp=input('depósito de Pedro ');
rp=input('tasa de interés de Pedro ');
n=0;
aj=0;
ap=0;
while aj<=ap
    n=n+1;
    aj=cj*(1+rj)^n;
    ap=cp*(1+rp)^n;
end
disp(n);
    %valor acumulado de Juan
    %valor acumulado de Pedro
    %en cada ciclo se incrementa un año a n
    %y se calculan los valores acumulados
    
```

Matricula: Nombre: Firma:

Tema 2. (30 puntos) SUBASTA INVERSA es un tipo de subasta en la que se invierte el papel de comprador y vendedor, con el objetivo principal de impulsar los precios de compra a la baja⁽¹⁾. Una vez que el comprador plantea el requerimiento, los vendedores registran el valor de su oferta y se selecciona la de menor precio; si más de un vendedor iguala el menor precio se selecciona aleatoriamente uno.

Elabore un algoritmo que, siguiendo las reglas descritas, permita:

- Ingresar las ofertas económicas para los n vendedores.
- Identificar el **monto** correspondiente a la **mejor oferta**.
- Determinar y mostrar cuántos vendedores cumplen con la mejor oferta y al vendedor seleccionado.

Ejemplo:
 ¿Cuántos vendedores?: 8
 valor oferta (1): 7
 valor oferta (2): 4
 valor oferta (3): 4
 valor oferta (4): 5
 valor oferta (5): 4
 valor oferta (6): 4
 valor oferta (7): 6
 valor oferta (8): 7
 El menor valor es: 4
 El vendedor ganador es: 3

(1) http://es.wikipedia.org/wiki/Subasta_inversa

Rubrica: Ingreso de datos en vector (5 puntos), menor valor (10 puntos), selección aleatoria (15 puntos)

```
v=input('ofertas de vendedores ');
n=length(v);
m=min(v);
vc=[];
for i=1:n
    if v(i)==m
        vc=[vc, i];
    end
end
nvc=length(vc);
k=fix(rand*nvc)+1;
disp(vc(k));
```

%la oferta de menor valor
 %vector de vendedores con la mejor oferta
 %agregar vendedor al vector
 %cantidad de vendedores con la mejor oferta
 %selección aleatoria del vendedor

Matricula: Nombre: Firma:

Tema 3. (25 puntos) Un almacén de ventas de productos por catálogos dispone de n vendedores asignados mensualmente de forma aleatoria a 4 regiones. El gerente de ventas mensualmente registra los **montos** de las ventas por cada vendedor para luego determinar el **total de ventas en dólares por región**.

Elabore un algoritmo, para un mes cualquiera, que permita ingresar los datos requeridos, asigne aleatoriamente el vendedor a una región, finalmente determine y muestre la información solicitada por el gerente de ventas.

Rubrica: Ingreso (5 puntos), Asignación de Regiones (10 puntos), Total de ventas/región (10 puntos). Vendedor del mes (10 puntos).

```

n=input('cuantos vendedores ');
for i=1:n
    r(i)=fix(rand*4)+1;           %asignación de región
end
for i=1:n                         %datos de ventas
    v(i)=input('total de ventas del vendedor ');
end
tr=[0, 0, 0, 0];                 %totales por región
for i=1:n
    nr=r(i);                       %número de región
    switch nr
        case 1, tr(1)=tr(1)+v(i);   %agregar venta a total por región
        case 2, tr(2)=tr(2)+v(i);
        case 3, tr(3)=tr(3)+v(i);
        case 4, tr(4)=tr(4)+v(i);
    end
end
disp(tr);
  
```

Matricula: Nombre: Firma:

Tema 4. (25 puntos) Se dispone de dos vectores que identifican el monto y región de las ventas de cada vendedor descritos en el problema anterior. Elabore un algoritmo para mostrar los *mejores vendedores por región*.

Ejemplo: n=7

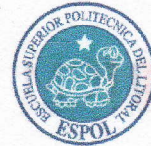
vendedor	monto	región
1	50	3
2	70	2
3	90	1
4	20	1
5	10	2
6	80	3
7	40	3

Resultado del Ejemplo: Región 1 = 3
 Región 2 = 2
 Región 3 = 6
 Región 4 = 0

Rubrica: inicialización de mayores (5 puntos), selección de mayor por región (15 puntos).
 Nota: Esta solución requiere el uso de arreglos.

```

n=input('cuantos vendedores ');
for i=1:n
    r(i)=fix(rand*4)+1;           %asignación de región
end
for i=1:n                         %datos de ventas
    v(i)=input('total de ventas del vendedor ');
end
for nr=1:4                         %ciclo para cada región
    vr=[ ];                       %inicio vector con vendedores de la región
    tvr=[ ];                       %inicio vector con las ventas de la región
    for i=1:n
        if r(i)==nr               %buscar los vendedores de la región
            vr=[vr,i];            %agregar vendedor y venta a los vectores
            tvr=[tvr, v(i)];
        end
    end
    mr=max(tvr);                   %la mejor venta de la región
    for i=1:length(vr)             %vendedores con la mejor venta en la región
        if tvr(i)==mr
            disp([nr, vr(i)]);
        end
    end
end
end
    
```



Nombre: Luis Rodríguez Ojeda Paralelo: 3

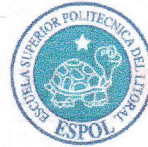
Tema 1 (20 puntos) Una serie alternada $1 - 1/2 + 1/3 - 1/4 + 1/5 - \dots$ es una serie en la cual los términos alternan el signo.

Realice un algoritmo para encontrar el resultado de la suma de la serie indicada hasta incluir al término $1/n$, siendo n un dato dado al inicio.

Rubrica: Alternar signos (5 puntos), resultado de la serie (15 puntos)

```
n=input('¿Cuántos términos? ');
s=0;
for i=1:n
    s = s + (-1)^(i+1)*1/i;
end
disp('La suma es');
disp(s);
```

blog.espol.edu.ec/educasanas/



Tema 2: (25 puntos) Existen tres salas contiguas A, B, C. Para entrar a la sala A hay dos puertas. Para pasar de la sala A a la sala B hay tres puertas y para pasar de la sala B a la sala C hay 4 puertas. Para entrar a cada sala se requiere escoger una puerta, la cual puede estar en tres estados aleatoriamente:

"Caminata aleatoria"			
	1	2	3
Inicio	2	3	4
Sala A	1	2	3
Sala B			
Sala C			

(1) pase a la siguiente sala, (2) puerta bloqueada, o (3) retroceda a la sala anterior. En cada intento, debe elegirse aleatoriamente una puerta y realizar la acción respectiva.

Escriba un algoritmo que simule la caminata desde el inicio hasta llegar a la sala C y determine la cantidad de intentos que se requirieron.

Rubrica: Generar estados diferentes de cada puerta por intento (5 puntos), validar el paso a otra sala (5 puntos), controlar éxito o fracaso (5 puntos) presentar resultados de los intentos (10 puntos).

```
sala=0;
i=0;
while sala < 3
    i=i+1;
    p=fix(rand*(sala+2))+1;
    e=fix(rand*3)+1;
    if e==1
        sala=sala+1;
    end
    if e==3 & sala > 0
        sala=sala-1;
    end
    disp('puerta');
    disp(p);
    disp('sala');
    disp(sala);
end
disp('Cantidad de intentos');
disp(i);
```

%Intentos
%Número de la puerta
%Estado de la puerta
%Pasa a la siguiente sala
%Retrocede a la sala anterior



Tema 3 (30 puntos) Para usar un "*Plan Acumulativo Mensual de Ahorro Programado*" se han establecido las reglas que se muestran a continuación:

- El depósito de ahorro mensual **A** es un valor fijo dado al inicio.
- El valor acumulado hasta donde se quiere llegar es un valor **P** dado inicialmente como un dato.
- Los **intereses ganados** serán registrados en forma mensual multiplicando el valor acumulado por 1% y agregándolo al valor acumulado actual.
- Cada tres meses pueden hacerse retiros parciales con un valor **C** de hasta 25% del valor total acumulado.
- El **valor total acumulado** podrá retirarse sólo cuando se haya alcanzado el valor **P**.

Realice un algoritmo que lea el valor del depósito mensual **A** y el valor al cual se desea llegar **P**. Cada mes debe agregarse un valor **A** al monto acumulado **P**. Cada tres meses debe realizarse aleatoriamente un retiro con un valor aleatorio de hasta 25% del valor acumulado. Determine: *cuántos meses* fueron necesarios hasta alcanzar la meta **P**, y *cuántos retiros* se realizaron.

Rubrica: manejo ahorro e intereses separados (5 puntos), control de retiros (5 puntos), conteo de transacciones (5 puntos), solución estructurada (10 puntos).

```
a=input('Depósito mensual fijo ');
p=input('Valor meta ');
s=0; %Valor acumulado actual
nr=0; %Número de retiros
nm=0; %Número de meses
while s<p
    s=s+a; %Agregar depósito mensual
    nm=nm+1;
    s=s+0.01*s; %Agregar interés
    if mod(nm,3)==0 %Retiro cada 3 meses
        r=fix(rand*2);
        if r==1 %Realizar retiro
            nr=nr+1;
            pr=fix(rand*25)+1; %Porcentaje de retiro
            c=pr/100*s; %Valor del retiro
            s=s - c; %Se reduce valor acumulado
        end
    end
end
disp('Cantidad de meses');
disp(nm);
disp('Cantidad de retiros');
disp(nr);
```



Tema 4 (25 puntos) Para el Homenaje a Michael Jackson, más de 1,6 millones de seguidores se registraron en Internet para participar en el sorteo de entradas para asistir a la ceremonia en estadio "Staples Center", y solo 8.750 participantes serían seleccionados para asistir. (www.eluniverso.com 07.07.2009).

Realice un algoritmo para sortear m boletos entre los n participantes registrados y presentar el listado de los números seleccionados.

Nota: Se supone que las personas se registran una sola vez, Suponga que n es mayor que m .

Rubrica: selección de m personas (10 puntos), selección de n personas no repetidas en el grupo anterior (5 puntos), muestra listado de seleccionados (5 puntos).

```
n=input('Número de participantes ');
m=input('Número de boletos ');
a = [];
while length(a)<m
    x = fix(rand*n)+1;
    if ismember(x, a)==0
        a = [a, x];
    end
end
disp('Lista de participantes afortunados');
disp(a);
```


Nombre: Luis Rodríguez Ojeda

Tema 1 (20 puntos)

El número 5 tiene la propiedad que al sumar el número 2 a su cuadrado, se obtiene un número que tiene raíz cúbica exacta: $5^2 + 2 = 27$

Realice un algoritmo para encontrar entre los números del 1 al 1000, si existe alguno que al sumar 2 a su cuadrado, el resultado tiene raíz cúbica exacta

```
%Búsqueda de enteros que cumplen una propiedad matemática
for n=1:1000
    r=n^2+2;
    if fix(r^(1/3))^3==r
        disp(n);
    end
end
```

blog.espol.edu.ec/educasomas/

Nombre: Luis Rodríguez Ojeda

Tema 2 (20 puntos)

El número de serie de un producto tiene 12 dígitos y está compuesto por:

Año: dos dígitos

Mes: dos dígitos

Día: dos dígitos

Inspector de calidad: dos dígitos

Orden de producción: cuatro dígitos

Realice un algoritmo que reciba un número de serie. Constate que tiene doce dígitos. Extraiga la información del producto y muestre el dato del inspector responsable y la fecha del reporte de fabricación

```
%Número de serie de 12 cifras
n=input('Ingrese un número de 12 cifras ');
if n>999999999999 | n<100000000000
    disp('Error en el código');
else
    an=fix(n/10000000000);
    n=mod(n,10000000000);
    mes=fix(n/100000000);
    n=mod(n,100000000);
    dia=fix(n/1000000);
    n=mod(n,1000000);
    insp=fix(n/10000);
    n=mod(n,10000);
    disp('Inspector');
    disp(insp);
    disp('Fecha del reporte');
    disp([an, mes, dia]);
end
```

```
%Solución usando una cadena de caracteres
x=input('Ingrese un número de 12 cifras ','s');
if length(x)~=12
    disp('error');
else
    a=x(1:2);
    m=x(3:4);
    d=x(5:6);
    insp=x(7:8);
    n=x(9:12);
    disp([insp,' ',a,' ',m,' ',d])
end
```

Nombre: Luis Rodríguez Ojeda

Tema 3 (30 puntos)

En el juego de Parchís intervienen dos jugadores con las siguientes reglas

- La meta es llegar primero a la casilla número 50
- Cada jugador realiza alternadamente una jugada lanzando un dado regular
- Al inicio, cada jugador está en la casilla de salida. Cada uno debe obtener el 5 en el dado para avanzar a la casilla número 1. Sinó, pierde el turno
- En cada turno el jugador avanza tantas casillas acorde al resultado obtenido en el dado.
- Si un jugador llega exactamente a la casilla ocupada por el rival, el rival debe volver a la casilla de salida.

Realice un algoritmo que permita simular el juego de Parchís indicando al final cual es el jugador triunfador y cuantos turnos se jugaron.

```
%Juego de Parchis entre dos jugadores (A y B)
a=0;
b=0;
turno=0;
while a<50 & b<50
    turno=turno+1;
    if a==0 %Jugador A
        x=fix(rand*6)+1;
        if x==5
            a=1;
        end
    else
        x=fix(rand*6)+1;
        a=a+x;
        if a==b
            b=0;
        end
    end
end
if a>=50
    break;
end
if b==0 %Jugador B
    x=fix(rand*6)+1;
    if x==5
        b=1;
    end
else
    x=fix(rand*6)+1;
    b=b+x;
end
end
```

Nombre: Luis Rodríguez Ojeda

```
if a==b
    a=0;
end
end
end
if a>b
    disp('Gana A');
else
    disp('Gana B');
end
disp('Turnos');
disp(turno);
```

Nombre: Luis Rodríguez Ojeda

Tema 4 (30 puntos)

Una empresa de correos recibe solicitudes de compras de sus clientes. Cada una es un envío que incluye los siguientes datos: Valor y Peso. Cada envío debe pagar el impuesto del IVA si el valor es mayor a \$400 y el peso mayor a 8 libras.

Realice un algoritmo para procesar un embarque de n envíos. Muestre el impuesto por cada envío y el total de impuestos por el embarque. Luego seleccione aleatoriamente una muestra de m envíos para que sean examinados por Aduana. Estos envíos seleccionados no deben generar impuestos y no deben estar repetidos

```
%Registro de envíos con valores y pesos
n=input('Cuántos envíos ');
for i=1:n
    v(i)=input('Ingrese valor de envío ');
    p(i)=input('Ingrese peso del envío ');
end
tot=0;
for i=1:n
    imp=0;
    if v(i)>400 & p(i)>8
        imp=1.12*v(i);
    end
    disp('Impuesto por envío ');
    disp([i, imp]);
    tot=tot+imp;
end
disp('Total de impuestos ');
disp(tot);
rev=[];
m=input('Cantidad de envíos para revisión ');
while length(rev)<m
    i=fix(rand*n)+1;
    if (v(i)<=400 | p(i)<=8) & ismember(i,rev)==0
        rev=[rev, i];
    end
end
disp('Envíos para revisión de Aduana');
disp(rev);
```

blog.espol.edu.ec/educasomas/

%m envíos elegidos para Aduana



COMPROMISO DE HONOR

Yo, al firmar este compromiso, reconozco que el presente examen está diseñado para ser resuelto de manera individual, que puedo usar un lápiz o esferográfico; que sólo puedo comunicarme con la persona responsable de la recepción del examen; y, cualquier instrumento de comunicación que hubiere traído, debo apagarlo y depositarlo en la parte anterior del aula, junto con algún otro material que se encuentre acompañándolo. Además no debo usar calculadora alguna, consultar libros, notas, ni apuntes adicionales a los que se entreguen en esta evaluación. Los temas debo desarrollarlos de manera ordenada.
Firmo al pie del presente compromiso, como constancia de haber leído y aceptado la declaración anterior.

Firma

NÚMERO DE MATRÍCULA: _____ PARALELO: _____

Tema No. 1 (25 PUNTOS)

Dos números son *amigables* si el primero es la suma de los divisores del segundo y viceversa.

Ejemplo: Los números 220 y 284 son *amigables*.

La suma de los divisores de 220 {1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110} es 284.

La suma de los divisores de 284 {1, 2, 4, 71, 142} es 220.

Escriba un programa que durante 100 oportunidades genere aleatoriamente parejas de números enteros de tres cifras y determine cuántos y cuáles resultaron *amigables* entre sí.

%Números amigables de tres cifras

```
for i=1:100
    a=fix(900*rand)+100;           %genera un número entre 100 y 999
    s=0;
    for d=1:a-1
        if mod(a,d)==0
            s=s+d;
        end
    end
    b=fix(900*rand)+100;           %genera otro número entre 100 y 999
    t=0;
    for d=1:b-1
        if mod(b,d)==0
            t=t+d;
        end
    end
    if s==b & t==a
        disp([a,b]);
    end
end
```

Tema No. 2 (25 PUNTOS)

Escriba un programa para el pago a los n vendedores por comisión de cierta empresa. Para cada vendedor se deben leer los siguientes datos: nivel al que pertenece (entero que puede ser 1, 2, 3), y el monto en dólares vendido en el mes. El pago del sueldo básico depende de su nivel:

Nivel 1:	\$400
Nivel 2:	\$500
Nivel 3:	\$600

A este sueldo básico hay que agregar el 5% de comisión sobre el monto de las ventas realizadas y adicionalmente hay un bono de \$100 al vendedor con el mayor monto de ventas (suponga que es uno solo). Lea los datos de cada vendedor y muestre:

- a) El valor que hay que pagar a cada vendedor
- b) El monto total que se requiere para pagar a todos los vendedores

```
%Pago de vendedores
n=input('Cantidad de vendedores ');
t=0;
v=0;
m=0;
for i=1:n
    nivel=input('Nivel del vendedor ');
    monto=input('Monto de ventas ');
    switch nivel
        case 1, p=400;
        case 2, p=500;
        case 3, p=600;
    end
    p=p+0.05*monto;
    t=t+p;
    disp('Vendedor');
    disp(i);
    disp('Valor a pagar');
    disp(p);
    if monto>m
        m=monto;
        v=i;
    end
end
disp('El vendedor');
disp(v);
disp('Recibe un bono de $100');
t=t+100;
disp('Valor total a pagar');
disp(t);
```

%Opcionalmente se puede usar vectores

blog.espol.edu.ec / educasanas!

Tema No. 3 (25 PUNTOS)

Una persona tiene una lista con los precios de n artículos y dispone de una cierta cantidad de dinero. Los artículos son identificados con la numeración natural. Escriba un programa para leer estos datos y obtener los siguientes resultados:

- Muestre la identificación de los artículos que puede comprar
- Para cada artículo cuyo precio es menor que la cantidad de dinero disponible, determine la cantidad que puede comprar

```
%Lista de precios
n=input('Número de artículos ');
c=input('Dinero disponible ');
for i=1:n
    disp('Artículo');
    disp(i);
    p=input('Ingrese el precio '); %Opcionalmente puede usar un vector
    if p<=c
        disp('Puede comprarlo en la cantidad')
        k=fix(c/p);
        disp(k);
    end
end
```


Tema No. 4 (25 PUNTOS)

Un polinomio de grado 5 puede ser representado con un vector de 6 posiciones:

$$p(x) = 4x^5 - x^4 + 2x^3 + x^2 - 3x + 5 \Rightarrow \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 4 & -1 & 2 & 1 & -3 & 5 \\ \hline x^5 & x^4 & x^3 & x^2 & x^1 & x^0 \end{array}$$

Escriba un programa para generar aleatoriamente coeficientes enteros de una sola cifra para los términos del polinomio y evaluarlo en un valor real $x = a$ con dos decimales de precisión (también generado aleatoriamente) comprendido en el intervalo [0.00, 10.00]. Muestre el par ordenado resultante $(a, p(a))$.

```
%Polinomio
for i=1:6
    c(i)=fix(10*rand);           %Coeficientes generados aleatoriamente
end
x=fix(1000*rand)/100;         %Valor aleatorio con dos decimales
p=0;
for i=1:6
    p=p+c(i)*x^(6-i);          %Evaluar el polinomio
end
disp([x,p]);
```

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.

Ejemplos:

Original	Codificado
472	← 383
503	← 603
615	← 606

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

blog.espol.edu.ec/educasanas/

(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);
n=fix(n/10);
d(2)=mod(n,10);
d(1)=fix(n/10);
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)
```

% Unidad

% Decena

% Centena

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
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), cálculo 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
    
```