

---

```

%Leidy Vargas M
%15/05/2019
%Enunciado2
% El entrenador de Básquetbol de los Borregos del Tec. de Monterrey
% está interesado en preparar lo que ha bautizado como la ensalada
  vitamínica,
% la cual puede prepararse a partir de 2 verduras básicas disponibles
  y
% definidas como 1,2; se desea que la ensalada vitamínica contenga por
  lo menos
% 10 unidades de vitamina A y 25 unidades de vitamina C,
% la relación neta del contenido vitamínico y el costo de las verduras
  se proporcionan en las restricciones.
%X1= Cantidad de vitamina 1 a emplear en la E.V. X2= Cantidad de
  vitamina 2 a emplear en la E.V.

                                %Min z= 100X1+80X2
                                %s.a: 2X1+3X2>=10
                                %      1X1+2X2>=25
                                %
                                %      x1, x2 #0

%A=[100 80 95]; %Se define A como funcion objetivo

%B=[ 2 3 ; %%Se define B como restricciones
% 1 2
% 4 6];

%C=[ 10; %% vector de recursos de las desigualdades
% 25
% 24];

function Aplicacion2()
clc;
clear all;

A=[2 3 10
  1 2 25
  100 80 0]; %Se define A como funcion objetivo
TRANSPUESTA = A.'
A=TRANSPUESTA([3],[1,2])
B=TRANSPUESTA([1,2],[1,2])
C=TRANSPUESTA([1,2],[3])

% Ventana para definir la función objetivo
prompt = {'Función objetivo = ', 'max == 1 or min==2', 'Numero de
  restricciones = '};
lineno = 1;
title = 'Ingreso de Datos';
def = {'A', '1', '2'};

```

---

---

```

options.Resize = 'on';
a      = inputdlg(prompt,title,lineno,def,options);
a      = char(a);
[~,n]  = size(a);
cout   = eval(a(1,:)); %Se transforman los valores (cadena de
    caracteres) ingresados en la FO a un vector de enteros
type   = eval(a(2,1)); %Se transforman los valores (cadena de
    caracteres) ingresados en el tipo de FO a un entero
nbr    = eval(a(3,1)); %Se transforman los valores (cadena de
    caracteres) ingresados en el numero de restricciones a un entero
str1   = struct('vari',{},'Type',{});
str2   = struct('var_base',{},'valeur',{});
%=====
% Ventana para definir restricciones
for i=1:nbr %Se definen los tipos de restricciones en orden
    prompt = {strcat('Ingrese el tipo de restricción para la
        condición ',num2str(i),' (<=,>=,=):')};
    title  = 'Ingreso, de Datos';
    def    = {'<='};
    options.Resize = 'on';
    p = inputdlg(prompt,title,lineno,def,options);
    p = char(p);
    opert = p;
    str1(1,i).Type = opert;
end
%=====
% Ventana para definir coeficientes de las restricciones
prompt = {'Ingrese la matriz de restricciones'};
lineno = 1;
title  = 'Ingreso de Datos';
def    = {'[2 1 ;3 2 ]'};
options.Resize='on';
t      = inputdlg(prompt,title,lineno,def,options);
t      = char(t);
sc     = eval(t); %Se transforman los valores (cadena de caracteres)
    ingresados en el campo, a un matriz de enteros
%=====
% Ventana para definir coeficientes de 'b'
prompt = {'Ingrese el vector de valores independientes b'};
lineno = 1;
title  = 'Ingreso de Datos';
def    = {'[100,80]'};
options.Resize = 'on';
u      = inputdlg(prompt,title,lineno,def,options);
u      = char(u);
second = eval(u);
%=====

sc1    = []; %Matriz de Variables de holgura
v_a    = zeros(1,length(cout));
v_e    = [];
v_b    = [];
v_ari  = [];
j      = 1;

```

---

---

```

%Paso a forma estandar
for i=1:nbr
    n = str1(1,i).Type;
    if n(1)~= '<' && isempty(sc2)
        sc2=zeros(nbr,1);
    end
    switch str1(1,i).Type
        case '<='
            v_e=[v_e second(i)];
            sc1(j,length(v_e))=1;
            v_b=[v_b,second(i)];

    end
    j=j+1;
end
%=====
sc      =[sc,sc1]; %Nueva Matriz de restricciones con variables de
    holgura añadidas
vari     =[];
vari_a   =[];
vari_e   =[];

for i=1:size(sc,2)
    str1(1,i).vari=['y',num2str(i)];
    vari=[vari,str1(1,i).vari,' '];
    if i<length(v_a)
        vari_a=[vari_a,str1(1,i).vari,' '];
    elseif i<=length(v_a)+length(v_e)
        vari_e=[vari_e,str1(1,i).vari,' '];

    end
end
%Primera iteración
x=[v_a,v_e];
Cj=[cout,0.*v_e];
Vb=[];
Q=v_b;
Ci=[];
tabl=[];
for i=1:length(Q)
    tabl=[tabl; ' | '];
    str2(1,i).valeur=Q(i);
    ind=find(x==Q(i));
    str2(1,i).var_base=str1(1,ind).vari;
    Vb=[Vb,str2(1,i).var_base,' '];
    Ci=[Ci,Cj(ind)];
end
Z=sum(Ci.*Q);
for i=1:length(Cj)
    Zj(i)=sum(Ci'.*sc(:,i));
end
Cj_Zj=Cj-Zj;
l=[];

```

---

---

```

for i=1:nbr
    if length(str2(1,i).var_base)==2
        l=[1;str2(1,i).var_base, ' '];
    else
        l=[1;str2(1,i).var_base];
    end
end
end
fprintf('\n');
disp('===== Problema en forma estandar
=====');
disp(['Variables : ',vari]);
disp(['          -Variables No Básicas      : ',vari_a]);
disp(['          -Variables Básicas         : ',vari_e]);

disp('=====
disp(' ');
disp('===== Tabla 0
=====');
disp(['Inicialización de variables : ',vari]);
disp(['          -Variables No Básicas      :
',num2str(v_a)]);
disp(['          -Variables Básicas         :
',num2str(v_e)]);

disp('=====
disp(' ');
disp(['Cj          : ',num2str(Cj)]);
disp('_____');
disp([tabl,num2str(Ci'),tabl,1,tabl,num2str(Q'),tabl,num2str(sc),tabl]);
disp('_____');
disp(['Zj          : ',num2str(Zj)]);
disp(['Cj-Zj        : ',num2str(Cj-Zj)]);
disp(['Z          : ',num2str(Z)]);
disp('_____');
disp(' ');
%Iteraciones de Simplex
t      = 1;
arret  = 1;
while arret==1
    if type==1
        num=max(Cj_Zj);num=num(1);
        num1=find(Cj_Zj==num);num1=num1(1);
        V_ent=str1(1,num1).vari;
    else
        g=min(Cj_Zj);g=g(1);
        num1=find(Cj_Zj==g);num1=num1(1);
        V_ent=str1(1,num1).vari;
        ['x',num2str(num1)];
    end
    b=sc(:,num1);
    k=0;d=10000;
    for i=1:length(Q)
        if b(i)>0
            div=Q(i)/b(i);
            if d>div

```

---

---

```

        d=div;
        k=i;
    end
end
end
if k~=0
    num2=k;
else
    disp('No se puede encontrar solución : La solución es
infactible ');
    break;
end
V_sort=str2(1,num2).var_base;
str2(1,num2).var_base=str1(1,num1).vari;
pivot=sc(num2,num1);
Ci(num2)=Cj(num1);
sc(num2,:)=sc(num2,:)./pivot;
Q(num2)=Q(num2)/pivot;
h=size(sc,1);
for i=1:h
    if i~=num2
        Q(i)=Q(i)-sc(i,num1)*Q(num2);
        sc(i,:)=sc(i,:)-sc(i,num1).*sc(num2,:);

    end
end
Z=sum(Ci.*Q);
for i=1:size(sc,2)
    Zj(i)=sum(Ci'.*sc(:,i));
end
Cj_Zj=Cj-Zj;
l=[];V=[];
for i=1:nbr
    if length(str2(1,i).var_base)==2
        l=[l;str2(1,i).var_base, ' '];
        V=[V,l(i,:), ' '];
    else
        l=[l;str2(1,i).var_base];
        V=[V,l(i,:), ' '];
    end
end
Vb      = V;
disp(['===== Tabla ',num2str(t),'
=====']);
disp(['Variable de entrada : ',num2str(V_ent)]);
disp(['Variable de salida : ',num2str(V_sort)]);
disp(['Pivote : ',num2str(pivot)]);
disp(['Variables Básicas : ',num2str(Vb)]);

disp('=====');
disp(' ');
disp(['Cj : ',num2str(Cj)]);

disp('_____');

```

---

---

```

disp([tabl,num2str(Ci'),tabl,1,tabl,num2str(Q'),tabl,num2str(sc),tabl]);

disp('_____');
disp(['Zj                : ',num2str(Zj)]);
disp(['Cj-Zj            : ',num2str(Cj-Zj)]);
disp(['Z                : ',num2str(Z)]);

disp('_____');
disp(' ');
disp(' ');
t      = t+1;
if type==1
    a=max(Cj_Zj);a=a(1);
    if a<=0
        break;
    end
else
    a = min(Cj_Zj);a=a(1);
    if a>=0 break;
    end
end
end
disp(['Resultado F.O. OPTIMO : ',num2str(Z)]);
disp('=====');
disp('SOLUCIÓN')
disp(['Necesita emplear: ',num2str(Cj_Zj(3)*-1),' gr de vitamina tipo
1']);
disp(['Necesita emplear: ',num2str(Cj_Zj(4)*-1),' gr de vitamina tipo
2']);
disp(['El costo será de:',num2str(Z),' pesetas']);

%k = msgbox( p,'RESULTADO F.O. OPTIMO :')

```

TRANSPUESTA =

|    |    |     |
|----|----|-----|
| 2  | 1  | 100 |
| 3  | 2  | 80  |
| 10 | 25 | 0   |

A =

|    |    |
|----|----|
| 10 | 25 |
|----|----|

B =

|   |   |
|---|---|
| 2 | 1 |
| 3 | 2 |

C =

100  
80

===== Problema en forma estandar

=====

Variables : y1 y2 y3 y4

-Variables No Básicas : y1

-Variables Básicas : y2 y3 y4

=====

===== Tabla 0

=====

Inicialización de variables : y1 y2 y3 y4

-Variables No Básicas : 0 0

-Variables Básicas : 100 80

=====

Cj : 10 25 0 0

|  |   |  |    |  |     |  |   |   |   |   |  |
|--|---|--|----|--|-----|--|---|---|---|---|--|
|  | 0 |  | y3 |  | 100 |  | 2 | 1 | 1 | 0 |  |
|  | 0 |  | y4 |  | 80  |  | 3 | 2 | 0 | 1 |  |

Zj : 0 0 0 0

Cj-Zj : 10 25 0 0

Z : 0

=====

===== Tabla 1

=====

Variable de entrada : y2

Variable de salida : y4

Pivote : 2

Variables Básicas : y3 y2

=====

Cj : 10 25 0 0

|  |    |  |    |  |    |  |     |  |   |  |   |  |      |  |
|--|----|--|----|--|----|--|-----|--|---|--|---|--|------|--|
|  | 0  |  | y3 |  | 60 |  | 0.5 |  | 0 |  | 1 |  | -0.5 |  |
|  | 25 |  | y2 |  | 40 |  | 1.5 |  | 1 |  | 0 |  | 0.5  |  |

Zj : 37.5 25 0 12.5

Cj-Zj : -27.5 0 0 -12.5

Z : 1000

=====

Resultado F.O. OPTIMO : 1000

=====

SOLUCIÓN

Necesita emplear: 0 gr de vitamina tipo 1

Necesita emplear: 12.5 gr de vitamina tipo 2

El costo será de:1000 pesetas

---

*Published with MATLAB® R2019a*