Grupo: 2BV3
Practica No: 6°
Fecha de realización: 10-Abril-07
Fecha de entrega: 17-Abril-07

ALGORITMO:

Algoritmo de factorizacion LDL^t

Entrada
La dimensión m; elementos a_{ij}, para 1≤i, j≤n de A

Salida
Los elementos l_{ij}, para 1≤j≤i y1≤i≤n de L y D_i, para 1≤i≤n de D

Paso 1 Para i=1,…,n haga los pasos 2-4

Paso 2 Para j=1,…i-1, tome $v_j=l_{ij}d_j$

Paso 3 Tome $d_i=a_{ii}-\sum_{j=1}^{i-1} l_{ij}v_j$

Paso 4 Para j=i+1,…n tome $l_{ij}=(a_{ji}-\sum_{k=1}^{i-1} l_{jk}v_k)/d_i$

Paso 5 SALIDA (l_{ij} para j=1,…,i-1 e i?1,…,n=;
SALIDA (d_i para i=1,…,n);
PARAR

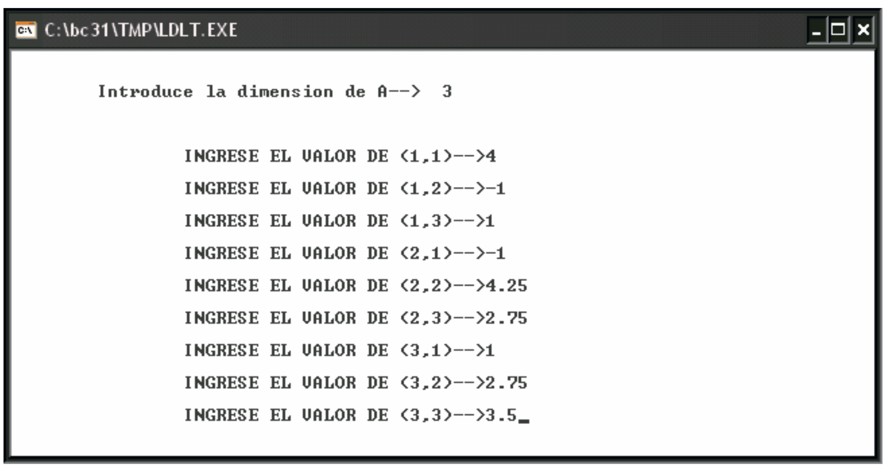
CÓDIGO FUENTE:

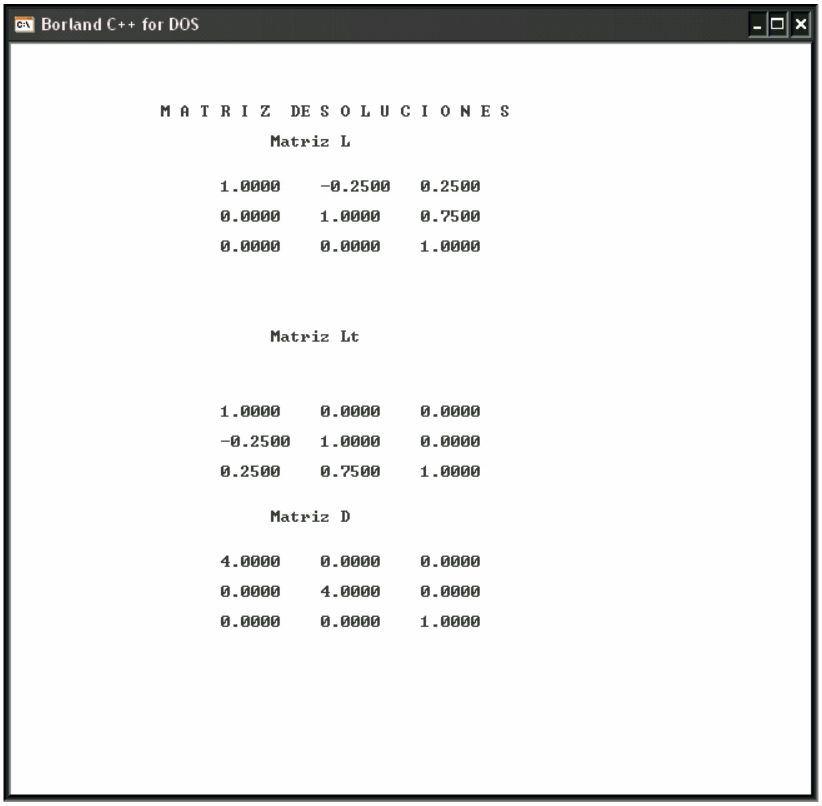
```
#include<stdlib.h>
#include<dos.h>
#include<stdio.h>
#include <conio.h>
#include<math.h>
void rellenar(int);
void soluciones (int);
void ldlt(int);
float a[10][10],l[10][10],d[10];

void main ()
{
int n;
clrscr();
printf ("\n\n\tIntroduce la dimension de A-->\t");
scanf ("%d",& n);
rellenar(n);
ldlt(n);
soluciones(n);
getch();
}
```

```
void ldlt(int n)
{
int i,j,k;
float sum,sum2,v[10];
for (i=1;i<=n;i++)
   {
   for (j=1;j<i;j++)
      {
      v[j]=l[i][j]*d[j];
      }
      sum=0;
   for (j=1;j<i;j++)
      {
      sum+=l[i][j]*v[j];
      }
      d[i]=a[i][i]-sum;
   for (j=i+1;j<=n;j++)
      {
      sum2=0;
      for (k=1;k<i;k++)
              {
              sum2+=l[j][k]*v[k];
              }
              l[j][i]=(a[j][i]-sum2)/d[i];
      }
   }
for (i=1;i<=n;i++)
   {
   l[i][i]=1;
   }
}

void soluciones (int n)
{
int y,r,i,j;
clrscr();
gotoxy(15,5);printf(" M A T R I Z  DE S O L U C I O N E S ");
gotoxy(27,7);printf ("Matriz L");
gotoxy(27,20);printf ("Matriz Lt");
gotoxy(27,32);printf ("Matriz D");
y=10;
for (i=1;i<=n;i++)
   {
   r=28-(2*n);
   for (j=1;j<=n;j++)
      {
      gotoxy(r,y);printf("%.4f",l[j][i]);
      gotoxy(r,y+15);printf ("%.4f",l[i][j]);
      if (j==i)
             {
             gotoxy (r,y+25); printf ("%.4f",d[i]);
             }
      else
              {
              gotoxy(r,y+25); printf ("0.0000");
              }
```

```
    r=r+10;
    }
  y=y+2;
  }
}
void rellenar(int n)
{
int i,j;
printf("\n\n");
for (i=1;i<=n;i++)
  {
  for (j=1;j<=n;j++)
    {
    printf("\n\t\tINGRESE EL VALOR DE (%d,%d)-->",i,j);
    scanf("%f",&a[i][j]);
    }
  }
}
```

CORRIDA





ALGORITMO:

Algoritmo de Choleski

Entrada
La dimensión m; elementos a_{ij}, para $1 \leq i$, $j \leq n$ de A

Salida
Los elementos l_{ij}, para $1 \leq j \leq i$ y para $1 \leq i \leq n$ de L. (Los elementos de $U = L^t$ son $u_{ij} = l_{ji}$, para $i \leq j \leq n$ y para $1 \leq i \leq n$.)

Paso 1 Tome $l_{11} = \sqrt{a_{11}}$
Paso 2 Para j=2,…,n Tome $l_{j1} = a_{j1}/l_{11}$
Paso 3 Para i=2,…,n-1 haga pasos 4 y 5

  Paso 4 Tome $l_{ii} = (a_{ii} - \sum_{k=1}^{i-1} l^2{}_{ik})^{1/2}$

  Paso 5 Para j=i+1,…,n

    Tome $l_{ji} = (A_{ji} - \sum_{k=1}^{n-1} l_{jk} l_{ik})/l_{ii}$

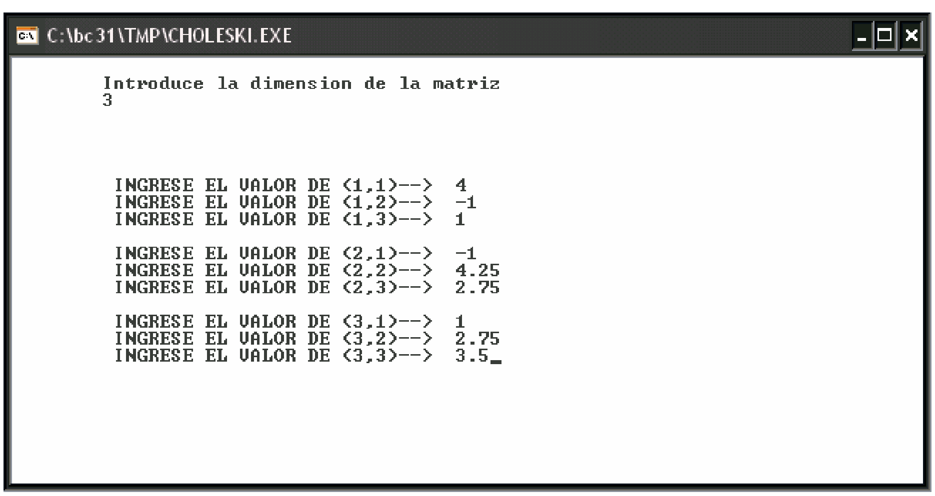
Paso 6 Tome $l_{nn} = (a_{nn} - \sum_{k=1}^{n-1} l^2)^{1/2}$
Paso 7 SALIDA (L_{ij} para j=1,…,i e i=1,…,n);
  PARAR

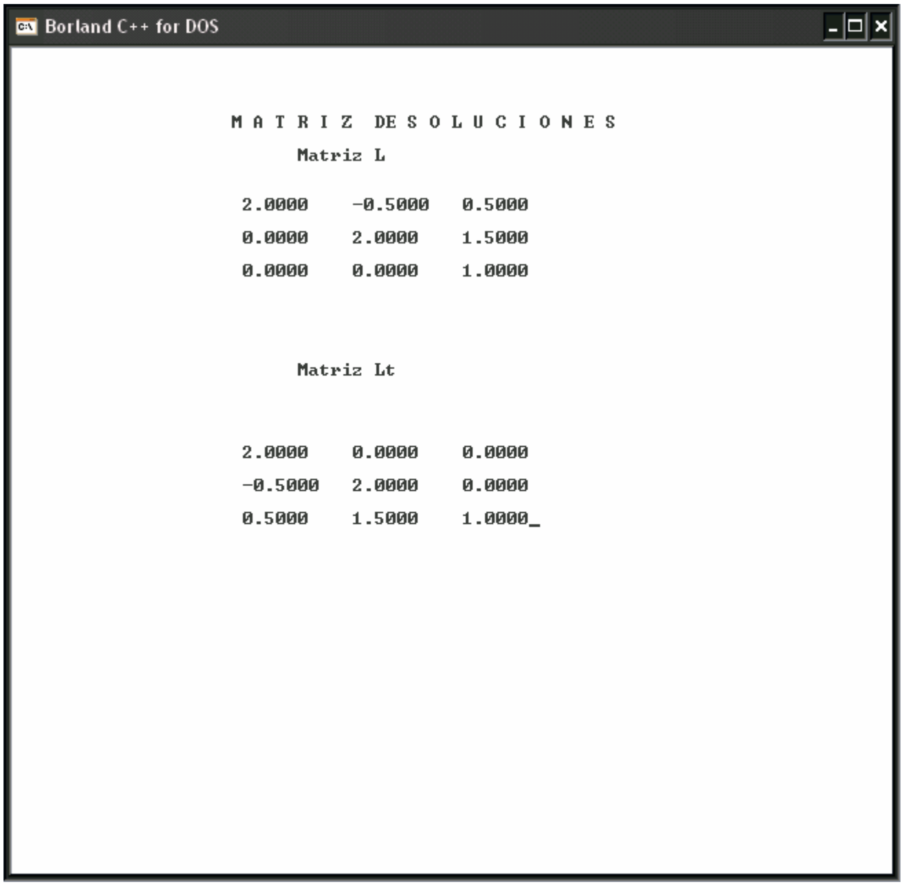
CÓDIGO FUENTE:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>

void rellenar(int);
void soluciones(int);
void transformacion(int);
float a[10][10],l[20][20];
void main ()
{
int n;
clrscr();
printf ("\n\tIntroduce la dimension de la matriz\n\t");
scanf ("%d",&n);
rellenar(n);
transformacion(n);
soluciones(n);
getch();
}

void transformacion (int n)
{
int i,j,k;
float sum,sum2,sum3;
l[1][1]=sqrt(a[1][1]);
for (j=2;j<=n;j++)
   {
   l[j][1]=a[j][1]/l[1][1];
   }
for (i=2;i<n;i++)
   {
   sum=0;
   for (k=1;k<i;k++)
      {
      sum=sum+(pow(l[i][k],2));
      }
   l[i][i]=sqrt(a[i][i]-sum);
   for (j=i+1;j<=n;j++)
      {
      sum2=0;
      for (k=1;k<i;k++)
            {
            sum2=sum2+(l[j][k]*l[i][k]);
            }
      l[j][i]=(a[j][i]-sum2)/l[i][i];
      }
   }
sum3=0;
for (k=1;k<n;k++)
   {
   sum3=sum+(pow(l[n][k],2));
   }
l[n][n]=sqrt(a[n][n]-sum3);
}
```

```
void rellenar(int n)
{
        int i,j,r=8;
        printf("\n\n");
        for (i=1;i<=n;i++)
        {
                for(j=1;j<=n;j++)
                {
                        gotoxy (10,r); cprintf("INGRESE EL VALOR DE (%d,%d)-->",i,j);
                        gotoxy (40,r); scanf("%f",&a[i][j]);
                        r++;
                }
                r++;
        }
}

void soluciones (int n)
{
int y,r,i,j;
clrscr();
gotoxy(20,5);printf(" M A T R I Z  DE S O L U C I O N E S ");
gotoxy(27,7);printf ("Matriz L");
gotoxy(27,20);printf ("Matriz Lt");
y=10;
for (i=1;i<=n;i++)
  {
  r=28-(2*n);
  for (j=1;j<=n;j++)
    {
    gotoxy(r,y);printf("%.4f",l[j][i]);
    gotoxy(r,y+15);printf ("%.4f",l[i][j]);
    r=r+10;
    }
  y=y+2;
  }
}
```

CORRIDA:




Borland C++ for DOS
M A T R I Z DE S O L U C I O N E S
Matriz L
2.0000   -0.5000   0.5000
0.0000   2.0000   1.5000
0.0000   0.0000   1.0000
Matriz Lt
2.0000   0.0000   0.0000
-0.5000   2.0000   0.0000
0.5000   1.5000   1.0000_