```
program Crout;
const MaxN=1000;
type matrix=array[1..MaxN,1..MaxN] of double;
     vector=array[1..MaxN] of double;

procedure ReadData(var N:integer; var R:double; var A:matrix; var b:vector);
var i,j:integer;
begin
  R:=0.0;
  readln(N);
  for i:=1 to N do
  begin
    for j:=1 to N do
    begin
      read(A[i,j]);
      if abs(A[i,j])>R then R:=abs(A[i,j])
    end;
    readln(b[i]);
  end;
end;

procedure WriteResults(N:integer; W:extended; var A:matrix);// var b:vector);
var i,j:integer;
begin
  writeln('|A|=',W);
  for i:=1 to N do
  begin
    for j:=1 to N do write(' ',A[i,j]);
    writeln{(b[i]);}
  end;
end;
```

```
procedure Error(m:integer);
begin
  writeln('Diagonal element A[m,m]=0 m=',m)
end;

procedure LU_decomposition(N:integer; R:double; var W:extended; var A:matrix);
const eps=1e-20;
var i,j,k,m:integer;
    D,S:double;
begin
  W:=1.0;
  for m:=1 to N-1 do
  begin
    D:=A[m,m]; W:=W*D;
    if abs(D/R)<eps then begin error(m); exit end;
    for i:=m+1 to N do
    begin
      S:=A[i,m];
      for k:=1 to m-1 do S:=S-A[i,k]*A[k,m];
      A[i,m]:=S/D;
    end;
    for j:=m+1 to N do
    begin
      S:=A[m+1,j];
      for k:=1 to m do S:=S-A[m+1,k]*A[k,j];
      A[m+1,j]:=S;
    end;
  end;
  W:=W*A[N,N];
end;
```

```
var N:integer;
    A:matrix;
    b:vector;
    W:extended;
    R:double;
begin
  ReadData(N,R,A,b);
  LU_decomposition(N,R,W,A);
  WriteResults(N,W,A)
end.
```