```
begin
   comment


   Time, N=21: 60.49s
   ;
   procedure INVERT2(n, a, eps, ERROR);
   value n, eps;
   integer n;
   real eps;
   array a;
   label ERROR;
   begin
      integer i, j, k;
      real pivot, z;
      integer array p, q[1:n];
      array b, c[1:n];
      for k := 1 step 1 until n do
      begin
         pivot := 0;
         for i := k step 1 until n do
         for j := k step 1 until n do
         if abs(a[i,j]) > abs(pivot) then
         begin
            pivot := a[i,j];
            p[k] := i;
            q[k] := j
         end;
         if abs(pivot) ≤ eps then go to ERROR;
         if p[k] ǂ k then
         for j := 1 step 1 until n do
         begin
            z := a[p[k], j];
            a[p[k], j] := a[k,j];
            a[k,j] := z
         end for j;
         if q[k] ǂ k then
         for i := 1 step 1 until n do
         begin
            z := a[i, q[k]];
            a[i, q[k]] := a[i,k];
            a[i,k] := z
         end for i;
         for j := 1 step 1 until n do
         begin
            if j = k then
            begin
               b[j] := 1/pivot;
               c[j] := 1
            end
            else
            begin
               b[j] := - a[k,j]/pivot;
               c[j] := a[j,k]
            end;
            a[k,j] := a[j,k] := 0
         end for j;
         for i := 1 step 1 until n do
         for j := 1 step 1 until n do
         a[i,j] := a[i,j] + c[i]×b[j]
      end for k;
      for k := n step -1 until 1 do
      begin
         if p[k] ǂ k then
      for i := 1 step 1 until n do
      begin
         z := a[i, p[k]];
         a[i, p[k]] := a[i,k];
         a[i,k] := z
      end;
      if q[k] ≠ k then
      for j := 1 step 1 until n do
      begin
         z := a[q[k], j];
         a[q[k], j] := a[k,j];
         a[k,j] := z
      end j
   end k
end INVERT2;
real procedure clock count;
begin
   real clock;
   boolean code;
   comment
   Pack the following instruction into code:

   zl,  hr s1
   62   17
   ;
   pack(code,
         0, 41,  0,
         0,  9,  0,
        10, 19,  1,
        20, 25, 62,
        30, 35, 17,
        39, 39,  1,
        40, 40,  1);
   clock count:=gier(code)
end;
integer Nmin,Nmax;
integer oldrand,N,mod,new;
Nmin := 19;
Nmax := 21;
mod := 2796203;
writecr;
writetext(≮<oldrand: ≯);
oldrand:=typein;
begin
   real time,maxerror,det;
   array xy[Nmin:Nmax,1:2];

   for N:=Nmin step 1 until Nmax do
   begin
      array A[1:N,1:N];
      integer i,j;
      real sum;
      writecr;
      write(≮dd≯,N);
      for i:=1 step 1 until N do
      begin
         sum:=0;
         for j:=1 step 1 until N do
         begin
            new := 125×oldrand;
            oldrand := new-new:mod×mod;
            A[i,j] := oldrand/mod-0.5;
         end for;
      end;
```

```
         clock count;
         INVERT2(N, A, 1₁₀-12, ERROR);
         goto OK;
ERROR:   writetext(‹<Error.›);
OK:      xy[N,2]:=clock count;
         xy[N,1]:=N;
         write(‹dddddd.dd›,xy[N,2]);
      end for N;
      begin
         procedure FIT1(n, meanerror, a, b, x, y);
         value n;
         integer n;
         real meanerror, a, b;
         array x, y;
         begin
            integer j;
            real SX, SX2, SY, SXY, SY2, DEN;
            SX := SX2 := SY := SXY := SY2 := 0;
            for j := 1 step 1 until n do
            begin
               SX := SX + x[j];
               SX2 := SX2 + x[j]↑2;
               SY := SY + y[j];
               SXY := SXY + x[j]×y[j];
               SY2 := SY2 + y[j]↑2
            end;
            DEN := n×SX2 - SX↑2;
            a := (SX2×SY-SX×SXY)/DEN;
            b := (n×SXY-SX×SY)/DEN;
            meanerror := sqrt((SY2+(2×SX×SY×SXY-n×SXY↑2-SX2×SY↑2)/DEN)/(n-1))
         end of FIT-1;
         array X,Y[1:Nmax-Nmin+1];
         real a,b,meanerror,x1,y1,e1,meanerror2;
         integer i;
         for i:=Nmax-Nmin+1 step -1 until 1 do
         begin
            X[i]:=ln(xy[i+Nmin-1,1]);
            Y[i]:=ln(xy[i+Nmin-1,2])
         end;
         FIT1(Nmax-Nmin+1, meanerror, a, b, X, Y);
         writecr;
         write(‹-nddddd.dddddd›,meanerror,a,b);
         writecr;
         writetext(‹<Time: ›);
         write(‹-n.ddd₁₀-d›,exp(a));
         writetext(‹<×n↑›);
         write(‹n.ddd›,b);
         if false then
         begin
            for i:=Nmin step 1 until Nmax do
            begin
               x1 := xy[i,1];
               y1 := exp(a)×x1↑b;
               e1 := y1-xy[i,2];
               writecr;
               write(‹ndd›,x1);
               write(‹-nddddd.ddd›,xy[i,2],y1,e1);
               meanerror2:=meanerror2+e1×e1
            end;
            writecr;
            write(‹-nddddd.ddd›,sqrt(meanerror2/(Nmax-Nmin)))
         end
      end fit
   end Nmin max
```

```
end;
```