```
algol<
begin
   comment

   N=61, no index check:

   Time classic:        603.71
   Time turbo:          599.31 0.7pct

   N=61, index check:

   Time classic:        1407.11
   Time turbo:          1250.24 11.1pct

   No buffer, N=21, index check:

   Time classic         67.66
   Time turbo:          64.94 4.0pct

   No buffer, N=21, no index check:

   Time classic         36.59
   Time turbo:          36.05 1.5pct



   ;
   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
         begin
         if abs(a[i,j]) > abs(pivot) then
         begin
            pivot := a[i,j];
            p[k] := i;
            q[k] := j
         end;
         end for;
         if abs(pivot) ≤ eps then go to ERROR;
         if p[k] ≠ 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;
         if q[k] ≠ 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;
      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;
      for i := 1 step 1 until n do
      for j := 1 step 1 until n do
      begin
      a[i,j] := a[i,j] + c[i]×b[j]
      end for;
   end for k;
   for k := n step -1 until 1 do
   begin
      if p[k] ≠ 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 for;
      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 for;
   end k
end INVERT2;
real procedure clock count;
code clock count;
1, 37;
  zl        , grf p-1   ; RF:=clock count; stack[p-1]:=RF;
e;
integer Nmin,Nmax;
integer oldrand,N,mod,new;
Nmin := 59;
Nmax := 61;
mod := 2796203;
select(17);
writecr;
writetext(≠<oldrand: ≠);
oldrand:=read integer;
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(¢-dddddd.dddddd¢,meanerror,a,b);
          writecr;
          writetext(¢<Time: ¢);
          write(¢-d.ddd₁₀-d¢,exp(a));
          writetext(¢<×n↑¢);
          write(¢d.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(|ddd|,x1);
            write(|-dddddd.ddd|,xy[i,2],y1,e1);
            meanerror2:=meanerror2+e1×e1
          end;
          writecr;
          write(|-dddddd.ddd|,sqrt(meanerror2/(Nmax-Nmin)))
        end
      end fit
   end Nmin max
end;
t<
```