unit NODE : class;
(* NODE OF BINARY TREE *)
var LEFT,RIGHT : NODE, VAL : INTEGER; (*SEARCHING KEY *)

unit INS : procedure (VALUE : INTEGER);
begin
if VAL > VALUE
  then
    if LEFT = NONE
      then
        LEFT := NEW NODE;
        LEFT.VAL := VALUE
      else
        CALL LEFT.INS(VALUE)
    fi
  else
    (* ELEMENTS NOT LESS THAN VAL ARE LOCATED IN THE RIGHT SUBTREE*)
    if RIGHT = NONE
      then
        RIGHT := NEW NODE;
        RIGHT.VAL := VALUE
    else
      CALL RIGHT.INS(VALUE)
    fi;
fi
end INS;

unit TRAVERS : COROUTINE (X :NODE);
(* CONSECUTIVE ELEMENTS OF TREE NODE ARE LOCATED IN THE GROWING ORDER to *)
(* THE "MAIL BOX" VAL AND SENT to THE ATTACHING unit *)
var VAL : INTEGER;

unit T : procedure (Y : NODE);
(* RECURSIVE procedure for INFIX TRAVERSION RESULTING TREE ELEMENTS *)
(* IN NOT DECREASING ORDER *)
begin
if Y /= NONE
  then
    CALL T(Y.LEFT);
    VAL := Y.VAL;
    DETACH;
    (* CONSECUTIVE ELEMENTS OF TREE Y ARE SENT for FURTHER *)
    (* PROCESSING to THE MASTER PROGRAM *)
    CALL T(Y.RIGHT);
  fi
end T;

begin
RETURN;
CALL T(X);
VAL := M;
(* VAL IS MAXIMAL VALUE TREATED AS A SENTINEL while ENTIRE TREE IS *)
(* TRAVESED *)
end TRAVERS;

var N,I,J,MIN,M,K : INTEGER,
(* N - THE NUMBER OF TREES
M - MAXIMAL KEY VALUE + 1
MIN - MINIMAL VALUE PRODUCED AT A GIVEN MOMENT BY SYSTEM OF COROUTINES*)

D : array of NODE,
TR : array of TRAVERS;

begin
WRITELN(" PROGRAM USES COROUTINES AND MERGES A GIVEN NUMBER OF BINARY",
" SEARCHING TREES");
do WRITELN(" GIVE THE NUMBER OF TREES:" );
READ(N);
WRITELN(N);
if N>0 then EXIT else WRITELN(" THE NUMBER MUST BE > 0") fi
od;
WRITELN(" ELEMENTS OF THE TREES ARE INTEGERS");
WRITELN(" to TERMiate INSERTING TREE TYPE -1.");
WRITELN(" THIS NUMBER IS NOT INSERTED AS AN ELEMENT");

array D DIM(1:N);
for I := 1 to N do
WRITELN(" GIVE THE ELEMENT SEQUENCE for THE TREE NO.",I:4);
READ(J); WRITE(J); if J>M then M := J fi;
D(I) := NEW NODE;
D(I).VAL := J;
do
READ(J);
if J = -1 then WRITELN; EXIT fi;
WRITE(J);
if J > M then M := J fi;
CALL D(I).INS(J)
od;
M := M+1;
end

WRITELN(" THE MERGED SEQUENCE IS:" );

array TR DIM(1:N);
MIN := 0;
(* GENERATE THE TRAVERSERS SYSTEM *)
for I := 1 to N do
TR(I) := NEW TRAVERS (D(I));
ATTACH(TR(I));
od;
K := 0;
do
if MIN = M then EXIT fi;
MIN := TR(1).VAL;
J := 1;
for I := 2 to N do
if MIN > TR(I).VAL then MIN := TR(I).VAL; J := I fi;
od;
if MIN < M then WRITE(' ',MIN); ATTACH(TR(J));
K := K + 1; if K = 10 then WRITELN fi
od; WRITELN
end