

A feltételes fordítást a futás szemantikus helyességének finom ellenőrzésére használó próbaprogram:

```

Program Proba;
(*
  Az ellenőrzés indító kellékei:
  - fordítási kapcsolók (f/b)
  - feltételes fordítás kapcsolói ("változói", pl. ell, inv, megall1)
*)
{$f+ $b-}
{$DEFINE ell}           (*egyszerű ellenőrzés: hibakiírás*)
{$DEFINE inv}           (*invariánsok ellenőrzése: hibakiírás*)
{$DEFINE megall1}      (*hiba esetén hibakiírás + megállás*)
(*
  A lényegi rész deklarációi:
  (Figyelem a Crt unitot használni kell!)
*)
Uses Crt;
Const MaxN=10;
Var   N,i:Integer;
        x:Array[1..MaxN] of Integer;

(*
  Feltételes fordítás rutinjai: include FFinc.pas
*)
{$i FFinc.pas}
{
  Ellenorzo fv-ek: -----
}
Function Nkisebb10:Boolean;
Begin
  Nkisebb10:=N>=10;
End;

Function Paros(szam:Integer):Boolean;
Begin
  Paros:=(x[szam] Mod 2)=0;
End;
{
  Ellenorzo fv-ek vege -----
}

Begin
  ClrScr;
  N:=10;

  {$IFDEF ell}
    If not LogIgaz('{$DEFINE ell} után: ellenorzes1',Nkisebb10)
      then Megall('Hiba1');
  {$ENDIF};

  For i:=1 to N do x[i]:=Random(2);

  {$IFDEF inv}
    If not LogLetezik('{$DEFINE inv} után: ellenorzes2',Paros,1,N)
      then Megall('Hiba2');
  {$ENDIF};

  Writeln('Ez mindig végrehajtottik.');
```

```

  {$IFDEF inv}
    If not LogMind('{$DEFINE inv} után: ellenorzes3',Paros,1,N)
      then Megall('Hiba3');
  {$ENDIF};

  {$UNDEF ell}      nincs további ellenőrzés)

```

```

{$IFDEF ell}
  Writeln('ellenorzesN');
{$ELSE}
  Writeln({'$UNDEF ell} után: nincs ellenorzesN');
{$ENDIF};
  ReadKey;
End.

```

Az ellenőrzés szükséges kellekeit tartalmazó inklúd-állomány (FFinc.pas):

```

{FFinc
  Kellekek a felteteles forditashoz:
}
  Type lFv0=Function: Boolean;
        lFv1=Function(i: Integer): Boolean;

  {0-változós fv ellenőrzése: False=>Hiba}
  Function LogIgaz(uzenet:String; b:lFv0): Boolean;
  Begin
    If not b then
      Begin
        Writeln('? ',uzenet,':',b,' ?');
        ReadKey;
      End;
    LogIgaz:=b;
  End;

  {1-, Integer-változós fv ellenőrzése: False=>Hiba}
  Function LogMind(uzenet:String; inv:lFv1; tol,ig:Integer): Boolean;
  Var i: Integer; mind: Boolean;
  Const valasz:Array [Boolean] of String=('Nem mind','Mind');
  Begin
    i:=tol;
    While (i<=ig) and (inv(i)) do Inc(i);
    mind:=(i>ig);
    If not mind then
      Begin
        Writeln('? ',uzenet,'(Mind?):',mind,' ?');
        ReadKey;
      End;
    LogMind:=mind;
  End;

  {1-, Integer-változós fv ellenőrzése: False=>Hiba}
  Function LogLetezik(uzenet:String; inv:lFv1; tol,ig:Integer): Boolean;
  Var i: Integer; let: Boolean;
  Const valasz:Array [Boolean] of String=('Nem létezik','Létezik');
  Begin
    i:=tol;
    While (i<=ig) and not (inv(i)) do Inc(i);
    let:=i<=ig;
    If not let then
      Begin
        Writeln('? ',uzenet,'(Létezik?):',let,' ?');
        ReadKey;
      End;
    LogLetezik:=let;
  End;

  Procedure Megall(uzenet:String);
  Begin
    {$IFDEF megall1}
    Writeln('! ',uzenet,' MEGALL !');
    Halt;
    {$ENDIF megall1}
  End;

```