

```

//Szlávi Péter
//SZPKAFT.ELTE
//szlavip@elte.hu
//
//Feladat: N szám magyarul, betűkkel kiírása...
//
//Ezt a programot a Közép-európai (IBM852) kódolással kell beolvasni
//a benne található ékezetes betűk miatt.

Program SzamokAtalakitasa;
Uses
  Crt;
Type
  TSzamjegy=Array [0..9] of String;
Const
  Egyes:TSzamjegy=('','egy','kettő','három','négy',
                  'öt','hat','hét','nyolc','kilenc');
  Tizes:TSzamjegy=('','Tizen','Huszon','Harminc','Negyven',
                  'Ötven','Hatvan','Hetven','Nyolcvan','Kilencven');
  MaxN=20;
Type
  TSzamok=Array [1..MaxN] of Integer;
  TKiirtSzamok=Array [1..MaxN] of String;
Var
  N:Integer;
  szamok:TSzamok;
  kiirtSzamok:TKiirtSzamok;

//az UpCase kiokosítása:
Function UpCase(Const c:Char):Char;
  Var
    cc:Char;
  Begin
    Case c of
      'é' : Begin cc:='É'; End;
      'í' : Begin cc:='Í'; End;
      'ó' : Begin cc:='Ó'; End;
      'ö' : Begin cc:='Ö'; End;
      'ő' : Begin cc:='Ő'; End;
      'ú' : Begin cc:='Ú'; End;
      'ü' : Begin cc:='Ü'; End;
      'ű' : Begin cc:='Ű'; End;
    else Begin cc:=System.UpCase(c); {a többire ugyanaz} End;
  End;
  UpCase:=cc;
End;

Procedure AdatBeolvasas;
  Var
    i:Integer;
  Begin
    Repeat
      Write('A számok száma:'); {$i-}Readln(N);{$i+}
    Until (IOResult=0) {szintaktikusan helyes-e}
      and (0<N) and (N<=MaxN) {szemantikusan helyes-e};
    Writeln('Kérem a számokat!');
    For i:=1 to N do
      Begin
        Repeat
          Write(i,':'); {$i-}Readln(szamok[i]);{$i+}
        Until (IOResult=0) {szintaktikusan helyes-e}
      End;
    End;
  End;

```

```

        and (0<szamok[i]) and (szamok[i]<100) {szemantikusan helyes-e};
    End;
End;

Function SzambolSzoveg(Const szam:Integer):String;
    Var
        t,e:Integer;
        s:String;
    Begin
        Case szam of
            10: Begin
                s:='Tiz';
            End;
            20: Begin
                s:='Husz';
            End;
            else Begin
                t:=szam Div 10; e:=szam Mod 10;
                s:=Tizes[t]+Egyes[e];
                s[1]:=UpCase(s[1]);
            End;
        End;
        SzambolSzoveg:=s;
    End;

Procedure Atalakitas;
    Var
        i:Integer;
    Begin
        For i:=1 to N do
            Begin
                kiirtSzamok[i]:=SzambolSzoveg(szamok[i]);
            End;
        End;

    Procedure EredmenyMegjelenites;
        Var
            i:Integer;
        Begin
            For i:=1 to N do
                Begin
                    Writeln(kiirtSzamok[i]);
                End;
            End;

    Begin
        ClrScr;
        AdatBeolvasas;
        Atalakitas;
        EredmenyMegjelenites;
        ReadKey;
    End.

```