```
   *----*    MuPAD 2.0.0 -- The Open Computer Algebra System
  /|   /|
 *----* |    Copyright (c)  1997 - 2000  by SciFace Software
 | *--|-*                   All rights reserved.
 |/   |/
 *----*                     UNREGISTERED VERSION
              Please contact info@sciface.com to register.

>> /*++
&> A1_I_20 -- Az Analízis I. példatár I/20. feladatának megoldása
&>
&> A1_I_20 (a,m,n)
&>
&> a - bemenõ a
&> m - monotonitási irány (1: növekvõ, -1: csökkenõ)
&> n - a vizsgált n-ek maximális száma
&> ++*/
>>
>> A1_I_20 := proc (a,m,n)
&> local v,i,f,f_e;
&> begin
&>  if abs(m) <> 1 or n=1 then
&>   return(FAIL);
&>   end_if;
&>
&>  v:=TRUE;
&>  i:=1;
&>  while v and (i<=n) do
&>   f:=i*(a^(1/i)-1);
&>   print(float(f));
&>   if i>1 then
&>    /*
&>     * Ha itt nem konvertálunk float-ba, akkor hibaüzenetet
&>     * kaphatunk, ugyanis csak valós számokat
&>     * lehet összehasonlítani egymással:
&>     */
&>    if float(f-f_e)*m<0 then v:=FALSE;
&>     end_if;
&>    end_if;
&>   f_e:=f;
&>   i:=i+1;
&>   end_while;
&>
&>  i:=i-1;
&>  if not v then
&>   i:=i-1;
&>   end_if;
&>  if m=-1 then
&>    print(Unquoted,"Monoton csökkenés n=".expr2text(i)."-ig");
&>   else
&>    print(Unquoted,"Monoton növekedés n=".expr2text(i)."-ig");
&>   end_if;
&>  end_proc:
```

```
>> // Példa:
>> A1_I_20(3.5,-1,10);

                              2.5

                          1.741657387

                          1.554883458

                           1.4711296

                          1.423675786

                          1.393145575

                          1.371861719

                          1.356178365

                          1.344142968

                          1.334615817

                    Monoton csökkenés n=10-ig
>> quit
```

```
>> /*++
&> A1_I_30_121 -- Az Analízis I. példatár I/30/121. feladat megoldása
&>
&> A1_I_30_121 (x,n)
&>
&> x - bemenõ x
&> n - bemenõ n
&> ++*/
>>
>> A1_I_30_121 := proc (x,n)
&> begin
&>     float(cos(x/n)^n);
&>     end_proc:
>>
>> // Példa:
>> A1_I_30_121(0.2, 100);

                         0.9998000199
>>
>> ////////////////////////////////////////////////////////////////////////
```

```
>> /*++
&> A1_I_30_121a -- Az I/30./121. feladat közelítõ megoldása
&>
&> A1_I_30_121a (x,h,m)
&>
&> x - bemenõ x
&> h - Cauchy-féle hibaküszöb
&> m - maximális lépésszám
&> ++*/
>>
>> // Addig növeljük n-et, amíg a sorozat két szomszédos elemének
>> // különbsége nem csökken h alá.
>> // (Cauchy-féle konvergenciakritérium.)
>> A1_I_30_121a := proc (x,h,m)
&> local i,v,f,f_e,d;
&> begin
&>  if m<2 then
&>   return(FAIL);
&>   end_if;
&>  i:=1;
&>  v:=TRUE;
&>  while v and i<=m do
&>   f:=A1_I_30_121(x,i);
&>   if i>1 then
&>    d:=float(abs(f-f_e));
&>    if d < h then
&>     v:=FALSE;
&>      end_if; // float...
&>    end_if;  // i>1...
&>   i:=i+1;
&>   if v and i<=m then
&>    f_e:=f;
&>    end_if; // v and...
&>   end_while; // v and...
&>  i:=i-1;
&>  if not v then
&>   print(Unquoted,"Az n=".expr2text(i-1).
&>    " esetben |f(n+1)-f(n)|<h");
&>  else
&>   print(Unquoted,"Az n=".expr2text(i-1).
&>    " esetben |f(n+1)-f(n)|=".expr2text(d));
&>   end_if;
&>  print(Unquoted,"f(n)=".expr2text(f_e).", f(n+1)=".expr2text(f));
&>  if not v then
&>   return(f);
&>  else
&>   return(FAIL);
&>   end_if;
&>  end_proc:
```

```
>> // Példa:
>> A1_I_30_121a(0.5,0.0001,100);

                    Az n=35 esetben |f(n+1)-f(n)|<h

                 f(n)=0.9964348204, f(n+1)=0.9965336877

                              0.9965336877
>>
>> A1_I_30_121a(0.5,0.00001,100);

              Az n=99 esetben |f(n+1)-f(n)|=0.0000126105692

                 f(n)=0.9987381652, f(n+1)=0.9987507757

                                  FAIL
>>
>> ////////////////////////////////////////////////////////////////////
>>
>> /*++
&> A1_I_30_121b -- Az I/30./121. feladat közelítõ megoldása
&>
&> A1_I_30_121b (x,h,m)
&>
&> x - bemenõ x
&> h - Cauchy-féle hibaküszöb
&> m - maximális lépésszám
&> ++*/
>>
>> // Az "a" verzióhoz képest annyit módosítunk,
>> // hogy n helyébe 2^n-et írunk.
>> A1_I_30_121b := proc (x,h,m)
....................................................................
&>   f:=A1_I_30_121(x,2^i);
....................................................................
&>   print(Unquoted,"Az n=2^".expr2text(i-1).
&>     " esetben |f(2n)-f(n)|<h");
&>  else
&>   print(Unquoted,"Az n=2^".expr2text(i-1).
&>     " esetben |f(2n)-f(n)|=".expr2text(d));
&>   end_if;
&>  print(Unquoted,"f(n)=".expr2text(f_e).", f(2n)=".expr2text(f));
>>
>> // Példa:
>> A1_I_30_121b(0.5,0.000000000001,100);

                    Az n=2^30 esetben |f(2n)-f(n)|<h

                        f(n)=1.0, f(2n)=1.0

                                1.0
```

```
>> ////////////////////////////////////////////////////////////////////
>>
>> // Ezután számos határérték-számítási feladatot
>> // gyorsan meg tudunk oldani:
>> A1_I_39_7 := proc (n)
&> local szorzat,i;
&> begin
&> szorzat:=1;
&> for i from 1 to 2*n-1 step 2 do
&>  szorzat:=szorzat*i;
&>  end_for;
&> return(n/(szorzat^(1/n)));
&> end_proc:
>>
>> // Lusták trükkje:
>> A1_I_30_121 := proc (x,n)
&> begin
&>  float(A1_I_39_7(n));
&>  end_proc:
>>
>> // Példa (sajnos, ennél a sorozatnál problémás...):
>> A1_I_30_121b(0.5,0.00001,10);

                Az n=2^9 esetben |f(2n)-f(n)|=0.0004596069263

                    f(n)=1.358221437, f(2n))=1.358681044

                                    FAIL
>> quit
```

```
>> /*++
&> A1_I_40_1 -- Az I/40/1. megoldása
&>
&> A1_I_40_1 (x,n)
&>
&> x - bemenõ x (a feladatban 2)
&> n - bemenõ n
&> ++*/
>>
>> A1_I_40_1 := proc (x,n)
&> begin
&>  if n=1 then sqrt(x)
&>  else sqrt(x*A1_I_40_1(x,n-1));
&>   end_if;
&>  end_proc:
>>
>> // Példa:
>> A1_I_40_1(2, 10);

                                 1023/1024
                                2
```
