```
{
      ***     \
  r **  *      |
  ** m  *      |
**------*--*    > x
  **..... *    |
  r **m'*. x'  |
      ***     /
}

Program PI_kozelites;
{ Kovács Zoltán, 2000/01/12 }

var r,x,m,x1,m1: extended;
    n,i: comp;
    c: char;

begin
    Write('r=');
    ReadLn(r);
    Write('x=');
    ReadLn(x);
    Write('n=');
    ReadLn(n);
    m:=sqrt(r*r-x*x/4);
    i:=1;
    Repeat
        WriteLn;
        WriteLn(i:0:0,'. közelítés, szabályos ',n:0:0,'-szög:');
        WriteLn('x=',x:0:19,', m=',m:0:19);
        WriteLn('Terület=',m*x*n/2:0:19);
        x1:=sqrt(x*x/4+(r-m)*(r-m));
        m1:=sqrt(r*r-x1*x1/4);
        x:=x1;
        m:=m1;
        n:=n*2;
        i:=i+1;
        Write('Tovább? (I/N) ');
        ReadLn(c);
        Until (c='N') or (c='n') or (n>1e18);
    WriteLn;
    WriteLn('Vége');
    WriteLn('A 19 jegyre pontos érték egyébként:');
    WriteLn('3.1415926535897932385 (a Turbo Pascal beépített értéke)');
    ReadLn;
    End.
```

```
{$N+}

{
      ***     \
  r **  *      |
  ** m  *      |
**------*--*    > x
  **..... *    |
  r **m'*. x'  |
      ***     /
}

Program PI_kozelites;
{ Kov cs Zolt n, 2000/01/12 }

var r,x,m,x1,m1: extended;
    n,i: comp;
    c: char;

begin
    Write('r=');
    ReadLn(r);
    Write('x=');
    ReadLn(x);
    Write('n=');
    ReadLn(n);
    m:=sqrt(r*r-x*x/4);
    i:=1;
    Repeat
        WriteLn;
        WriteLn(i:0:0,'. kM-^TzelĄtM-^Bs, szab lyos ',n:0:0,'-szM-^Tg:');
        WriteLn('x=',x:0:19,', m=',m:0:19);
        WriteLn('TerM-^Alet=',m*x*n/2:0:19);
        x1:=sqrt(x*x/4+(r-m)*(r-m));
        m1:=sqrt(r*r-x1*x1/4);
        x:=x1;
        m:=m1;
        n:=n*2;
        i:=i+1;
        Write('Tov bb? (I/N)');
        ReadLn(c);
        Until (c='N') or (c='n') or (n>1e18);
    WriteLn;
    WriteLn('VM-^Bge');
    WriteLn('A 19 jegyre pontos M-^BrtM-^Bk egyM-^BbkM-^Bnt:');
    WriteLn('3.1415926535897932385 (a Turbo Pascal beM-^BpĄtett M-^BrtM-^Bke)');
    ReadLn;
    End.
```

```
program RegulaFalsi;
{ Lásd Szendrei János: Algebra és számelmélet c. könyvét, 357. o. }

const
    debug=true;
    PONTOSSAG=15;

var a,b,e: real;

function f(x: real): real;
begin
    f:=x*x*x*x*x-x*x*x*x-3*x*x*x+2*x+5;
end;

function hurmodszer: real;
var x: real;
    v: boolean;
    j: integer;
begin
    v:=false;
    j:=1;
    repeat
        x:=(a*f(b)-b*f(a))/(f(b)-f(a));
        if debug then
            writeln('j=',j:2,' a=',a:0:PONTOSSAG,
                ' b=',b:0:PONTOSSAG,' x=',x:0:PONTOSSAG);
        if abs(f(x))<e then
            begin
                hurmodszer:=x;
                v:=true;
            end;
        if f(a)*f(x)<0 then b:=x else a:=x;
        j:=j+1;
    until v;
end;

begin
    a:=1; b:=7/4; e:=10e-15;
    writeln(hurmodszer:0:PONTOSSAG);
end.
```

```
{ Pontosság-tesztek }
{ Kovács Z., 2002/02/16 }

Type egesz = integer; { pl. integer, longint, int64 }
    valos = single; { pl. single, double, extended }

procedure Szohossz;
{ Az egész típusok pontosságait vizsgálja. Az 1-et addig duplázza,
  amíg negatív értéket nem kapunk. }
var e1, e2, h: egesz;
begin
    e2:=1;
    h:=1;
    repeat
        e1:=e2;
        e2:=e1*2;
        h:=h+1;
        until e2<e1;
    writeln('Szóhossz: e1=',e1,' e2=',e2,' h=',h);
    end;

procedure Legkisebb(o: egesz);
{ A valós típusok pontosságát vizsgálja. Az 1-et addig osztja a bemenettel,
  amíg 0-t nem kapunk. }
var v1, v2: valos;
    h: egesz;
begin
    v2:=1;
    h:=1;
    repeat
        v1:=v2;
        v2:=v1/o;
        h:=h+1;
        until v2=0;
    writeln('Legkisebb(',o,'): v1=',v1:0,' h=',h);
    end;

procedure Epszilon(o: egesz; r: boolean);
{ A valós típusok pontosságát vizsgálja. Az 1-et addig osztja a bemenettel,
  amíg 0-t nem kapunk, azután az utolsó nem 0 számmal végez alapműveleteket. }
var v1, v2: valos;
    i: integer;
begin
    v2:=1;
    repeat
        v1:=v2;
        v2:=v1/o;
        until v2=0;
    write('Epszilon(',o,'):');

    v2:=0;
    for i:=1 to 5 do begin
        v2:=v1+v2;
        write(' ',i,'*e=',v2);
        end;

    v2:=v1*v1;
    write(' e^2=',v2);

    { A reciprok Free Pascal 1.0.4-gyel 0-val osztást eredményez,
      Turbo Pascalban csak valos=double esetén: }
```



```
    if r then begin
        v2:=1/v1;
        write(' 1/e=',v2);
        end;

    writeln;
    end;

procedure Kilenc;
{ A valós típusok pontosságát vizsgálja. Az 1-et addig osztja 10-zel,
  amíg 0-t nem kapunk, s ezeket a hányadosokat szummázza. }
var v1, v2, s, s9: valos;
    h: egesz;
begin
    v2:=1;
    h:=1;
    s:=0;
    write('Kilenc:');
    repeat
        s:=s+v2;
        s9:=s*9;
        v1:=v2;
        v2:=v1/10;
        write(' s(',h,')=',s,' s9(',h,')=',s9);
        h:=h+1;
        until s9=10;
    writeln;
    end;

procedure Legnagyobb;
{ A valós típusok pontosságát vizsgálja. Az 1-et addig szorozza 2-vel,
  amíg nem kapunk hibajelzést. }
var v1, v2: valos;
    i: egesz;
begin
    v2:=1;
    i:=0;
    write('Legnagyobb:');
    repeat
        v1:=v2;
        v2:=v1*2;
        i:=i+1;
        write('.');
        if i mod 10 = 0 then write(i);
        until v2<v1;
    writeln;
    end;

begin
    Szohossz;
    Legkisebb(2);
    Legkisebb(10);
    Epszilon(2,false);
    Epszilon(10,false);
    Kilenc;
    Legnagyobb;
    end.
```