

INFORMATIKA ALKALMAZÁS MÓDSZERTANA

I. beadandó

Készítette:
Kiss Endre Farkas

FEJLESZTŐI DOKUMENTÁCIÓ

Feladat:

Készítsünk programot, mely a nagypontosságú aritmetika eljárásai segítségével közelíti az alábbi függvényt a hatványsorával:

$$\ln(x) = \sum_{n=0}^{\infty} \frac{2}{2n+1} \left(\frac{x-1}{x+1} \right)^{2n+1}$$

Specifikáció:

Be: $x \in \mathbf{Z}$, tagszam $\in \mathbf{Z}$

Ef: -

Ki: aktuallag^{tagszam} : ValosSzam, vegeredmeny : ValosSzam;

Uf: $\text{aktuallag}_{\text{tagszam}} = \frac{2}{2 * \text{tagszam} + 1} \left(\frac{x-1}{x+1} \right)^{2 * \text{tagszam} + 1}$

$\text{vegeredmeny} = \sum_{n=0}^{\text{tagszam}-1} \text{aktuallag}_n$

Algoritmus:

Az algoritmus a nagypontosságú aritmetika („aritunit”, és „raciona5”) eljárásainak meghívására épül.

Az „aritunit” unitban használt típusok:

<i>Szamjegy</i>	Egész szám
<i>PozSzam</i>	Tömb [0.. <i>MaxPontossag</i> +1] <i>Szamjegy</i> típusúakból
<i>SzamrendszerTipus</i>	2..256;
<i>PontossagTipus</i>	0.. <i>MaxPontossag</i> +1;
<i>ElojelTipus</i>	(Poz,Neg);
<i>EgeszSzam</i>	Rekord <i>elojel</i> : <i>ElojelTipus</i> <i>hossz</i> : <i>PontossagTipus</i> <i>szamrendszer</i> : <i>SzamrendszerTipus</i> <i>jegy</i> : <i>PozSzam</i> Rekord vége
<i>Tort</i>	Rekord <i>Szamlalo, nevezo</i> : <i>ElojelTipus</i> Rekord vége

ValosSzam Rekord
 Egesz, tort: *egeszSzam*
 Rekord vége

Eljárások a „raciona5” unitban:

TortOsszead (*változó a, b, osszeg : Tort*);

Két TortSzam típusú szám összegét adja vissza TortSzam-ként

TortKivon (*változó a, b, kulonbseg : Tort*)

Két TortSzam típusú szám különbségét adja vissza TortSzam-ként

Egeszbol_tort (*változó szamlalo,nevezo: EgeszSzam; Var rac: Tort*)

Egy EgeszSzam típusú számlálóból és nevezőből egy TortSzam értéket ad vissza.

TortSzoroz: (*változó a, b, Szorzat : Tort*)

Két TortSzam típusú szám szorzatát adja vissza TortSzam-ként

TortOsz (*változó a, b, Hanyados : Tort*)

Két TortSzam típusú szám hányadosát adja vissza TortSzam-ként

Eljárások a „airunit” unitban:

Beolvasas (*változó a: EgeszSzam; kerdes: String*)

Felhasználótól vár egy számot, melyet EgeszSzam típusúként kezel a továbbiakban.

Kiiras (a: EgeszSzam; szoveg: *szöveg*)

Képernyőre ír egy EgeszSzam típusú értéket.

Kivon (*változó a,b,kulonbseg: EgeszSzam*)s

Két EgeszSzam típusú szám különbségét adja vissza EgeszSzam-ként

Osszead (*változó a,b,osszeg: EgeszSzam*)

Két EgeszSzam típusú szám összegét adja vissza EgeszSzam-ként

Szoroz (*változó a,b,szorzat: EgeszSzam*)

Két EgeszSzam típusú szám szorzatát adja vissza EgeszSzam-ként

Ezt egészíti egy, a programkódban a külön definiált **ertekreHoz** nevű eljárás, amely egy egész számot alakít át az aritmetikai unitokban definiált EgeszSzam típusúvá.

A megoldás során a következő változókat használtam:

szam, seged, egy, ketto, nevezó, szamlalo : EgeszSzam
tortEgy, tortKetto, tortSeged, masodikTagAlap,
masodikTag, tortSzamlalo, tortNevezó, elsotag,
aktualTag, vegeredmeny : tort
aktValos, valosVegeredmeny, masodikValos : valosSzam
tagszam, n, i : egész szám

Maga a megoldás a következő algoritmus szerint történik:

Beolvasas(szam, 'x? ')

Az 1-nek és 2-nek létrehozzuk a tört számú megfelelőjét

ertekreHoz(ketto, 2)
ertekreHoz(egy, 1)
egeszbol_tort (egy, egy, tortEgy)
egeszbol_tort (ketto, egy, tortKetto)
felhasználótól bekér (tagszam)
kivon (szam, egy, szamlalo)

A „szamlalo” ezen a ponton a képlet szerinti $x-1$

osszead (szam, egy, nevezó)

A „nevezó” ezen a ponton a képlet $x+1$

egeszbol_tort (nevezó, egy, tortNevezó)
egeszbol_tort (szamlalo, egy, tortSzamlalo)
tortOszT (tortSzamlalo, tortNevezó, masodikTagAlap)

A „masodikTagAlap” itt $(x-1)/(x+1)$ pontosan (tört)

masodikTag := masodikTagAlap

ha tagszam>0 akkor

 ciklus n:=0 to tagszam-1-ig

 masodikTag := masodikTagAlap

 ciklus i:=1-től (2^n) -ig

 tortSzoroz(masodikTag, masodikTagAlap, masodikTag)

 ciklus vége

A „masodikTag” ezen a ponton $(x-1)/(x+1)$ a $(2n+1)$ -ediken

ertekreHoz(seged,n)

A „segéd” innentől n

szoroz(ketto, seged, seged)

A „segéd” innentől 2n

osszead(seged, egy, seged)

A „segéd” innentől 2n+1

egeszbol_tort (ketto, seged, elsotag)

Az „elsotag” innentől $2/(2n+1)$

tortSzoroz (elsotag, masodikTag, aktualTag)

Konv_RacVal (aktValos, aktualTag)

képernyőre ír : n+1, '. tag értéke:'

valosKiir (aktValos)

Mivel törtszámként a nulla értelmezhetetlen

Ezért az első tagnál kezdőértéket adunk a végeredménynek

ha n=0 akkor

 vegeredmeny := aktualTag

különben

A többi tagnál pedig az aktuális tagot

az előzőekhez adjuk

 tortosszead (vegeredmeny, aktualTag, vegeredmeny)

elágazás vége

 ciklus vége

elágazás vége

Konv_RacVal (valosVegeredmeny, vegeredmeny)

képernyőre ír 'Az aktuális végeredmény:'

valosKiir (valosVegeredmeny)

PROGRAMKÓD:

```
{ Készítette: Kiss Endre Farkas }
program nagypo7nt;
uses AritUnit, crt, Raciona5;
var
  szam, seged,
  egy, ketto, nevezo,
  szamlalo          : EgeszSzam;

  tortEgy, tortKetto,
  tortSeged, masodikTagAlap, masodikTag,
  tortSzamlalo, tortNevezo,
  elsoTag, aktualTag, vegeredmeny : tort;

  aktValos, valosVegeeredmeny, masodikValos : ValosSzam;

  tagszam,n,i: integer;
  c, c2: char;

procedure ertekreHoz (var atalakitott: EgeszSzam; atadott: byte);
begin
  { A megkapottbyte típusú egyjegyű számot egész számként adjuk vissza. }
  atalakitott.elojel := Poz;
  atalakitott.hossz := 0;
  atalakitott.szamrendszer:=10;
  atalakitott.jegy[0]:=atadott;
end;

begin
  repeat
    clrscr;
    writeln ('At alkalmazás a következő függvényt közelíti a
hatványsorával:');
    writeln;
    writeln ('
    writeln ('ln'#$28'x'#$29' = szumma'#$28'  $2/2n+1$  * '#$28'x-1/x+1'#$29'
'#$29);
    writeln;
    writeln ('Ahol n = 0 .. végtelen');
    writeln;
    writeln('-----');
    writeln(' Adja meg az adatokat!');
    writeln;
    Beolvasas(szam, 'x? ');

    { Az 1-nek és 2-nek létrehozuk a tört számú megfelelőjét }
    ertekreHoz(ketto, 2);
    ertekreHoz(egy, 1);
    egeszbol_tort (egy, egy, tortEgy);
    egeszbol_tort (ketto, egy, tortKetto);

    writeln ('Hány tagig nézzem? ');
    readln (tagszam);

    kivon (szam, egy, szamlalo);
    { a szamlalo itt x-1 }
    osszead (szam, egy, nevezo);
    { a nevezo itt x+1 }
    egeszbol_tort (nevezo, egy, tortNevezo);
    egeszbol_tort (szamlalo, egy, tortSzamlalo);
    tortOsz (tortSzamlalo, tortNevezo, masodikTagAlap);
    { a masodikTagAlap itt (x-1)/(x+1) pontosan (tört) }
    masodikTag := masodikTagAlap;

    if tagszam>0 then
      begin
```

```

for n:=0 to tagszam-1 do begin
  masodikTag := masodikTagAlap;
  for i:=1 to ((2*n)) do
    begin
      tortSzoroz(masodikTag, masodikTagAlap, masodikTag);
    end;
  { a masodikTag ezen a ponton ((x-1)/(x+1)) a (2n+1)-ediken }

  ertekreHoz(seged,n); { A segéd inentől n }
  szoroz(ketto, seged, seged); { A segéd inentől 2n }
  osszead(seged, egy, seged); { A segéd inentől 2n+1 }
  egeszbol_tort (Ketto, seged, elsotag);
  { Az elsotag inentől 2/(2n+1) }
  tortSzoroz (elsőTag, masodikTag, aktualTag);
  Konv_RacVal (aktValos, aktualTag);
  writeln;
  writeln ( n+1, '. tag értéke:');
  valoskiir (aktValos);
  writeln;
  writeln ('Nyomjon egy gombot a folytatáshoz!');
  writeln;
  c2 := readkey;
  { Mivel törtszámként a nulla értelmezhetetlen ... }
  if n=0 then begin
    { Ezért az első tagnál kezdőértéket adunk a végeredménynek }
    vegeredmeny := aktualTag;
  end else begin
    { A többi tagnál pedig az aktuális tagot
      az előzőekhez adjuk }
    tortosszead (vegeredmeny, aktualTag, vegeredmeny);
  end;
end;
end;
writeln;

Konv_RacVal (valosVegeredmeny, vegeredmeny);
writeln ('Az aktális végeredmény:');
valoskiir (valosVegeredmeny);
writeln;
writeln ('Nyomjon Q-t az újratekéshez. ');
writeln ('Bármí mást új függvényadatokhoz. ');
c := readkey;

until c='q';
end.

```