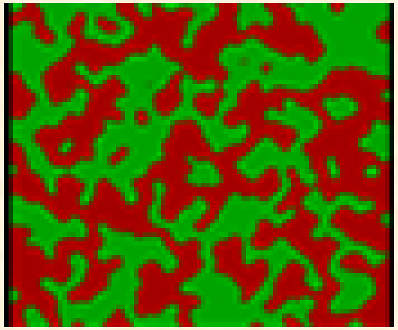
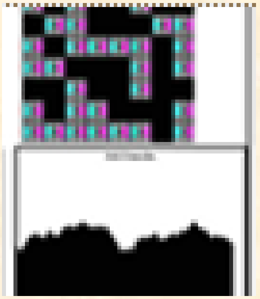
Nagypontosságú aritmetika I.

Nagypontosságú aritmetika

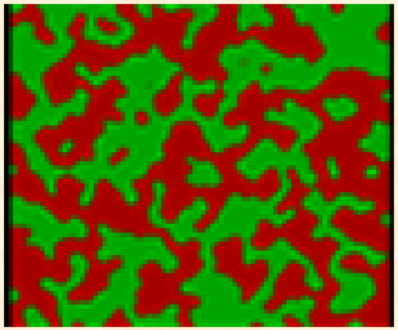


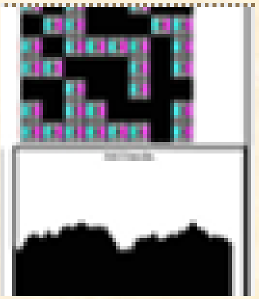


Problémák:

- sokjegyű (100 vagy 1000 vagy ...) egész számok kellenek több alkalmazásban;
- jó lenne, ha 1/3*3 értéke 1 lenne, azaz kellenének racionális számok, esetleg nagyon sok számjegyű számlálóval és nevezővel;
- ha beszélünk fixpontos számábrázolásról, jó lenne ha lennének fixpontos valós számok;
- nagy pontosságú lebegőpontos valós számok kellenek több alkalmazásban.

Nagypontosságú aritmetika: egész számok



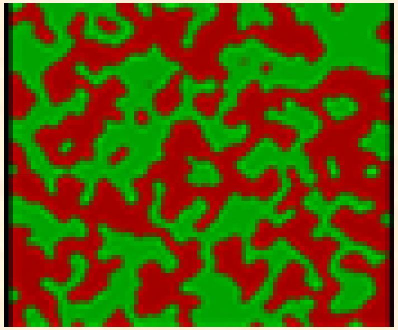


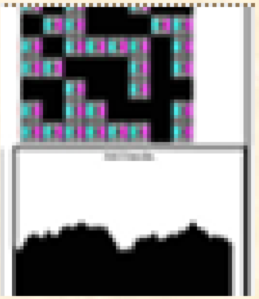
Számok ábrázolása:

- komplemens ábrázolás (negatív számok így nagyon sokjegyűek)
- előjel + számjegyek + hossz + számrendszer (tömb vagy szöveg):

$$x = \pm\, t_0 + t_1 S + t_2 S^2 + \ldots + t_n S^n$$

Nagypontosságú aritmetika: egész számok



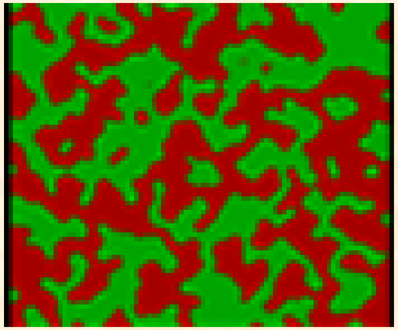


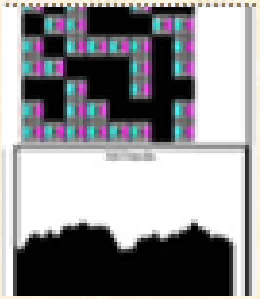
NagyEgész típus:

- előjel: {–,+}
- n: Egész
- S: alapszám
- T: tömb(0..Maxn:Egész)

$$x = \pm\, t_0 + t_1 S + t_2 S^2 + \ldots + t_n S^n$$

Nagypontosságú aritmetika: egész számok

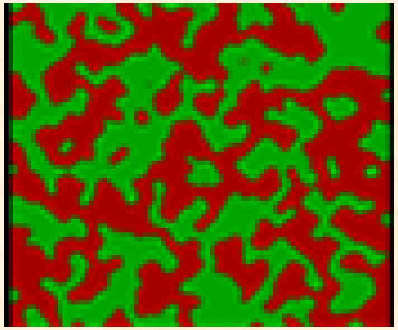


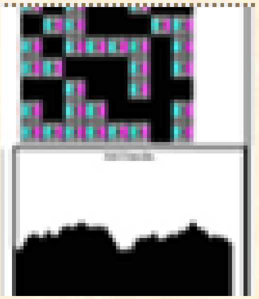


A műveleteknél az előjelet külön kezeljük, a műveleteket visszavezetjük pozitív számokkal végzett műveletekre.

```
Összeadás(x,y,z):
  z.előjel:=x.előjel; z.S:=x.S
  Ha x.előjel=y.előjel
              akkor Összead(x,y,z)
  különben ha x.előjel<y.előjel
              akkor Kivon(y,x,z)
              különben Kivon(x,y,z)
Eljárás vége.
```

Nagypontosságú aritmetika: egész számok



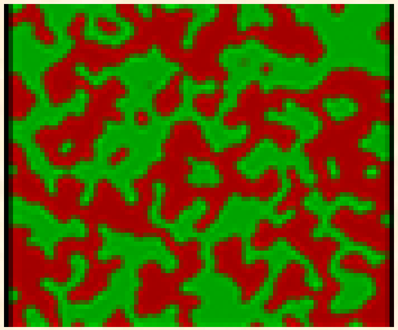


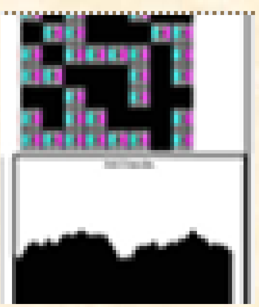
Még arra is figyelhetünk, hogy |x|≥|y| legyen!

```
Összeadás(x,y,z):
  z.előjel:=x.előjel; z.S:=x.S
  Ha x.előjel=y.előjel
    akkor Ha nagye(x,y) akkor Összead(x,y,z)
                     különben Összead(y,x,z)
  különben ha x.előjel='-'
     akkor Ha nagye(y,x) akkor Kivon(y,x,z)
                               z.előjel:='+'
                      különben Kivon(x,y,z)
  különben ha nagye(x,y) akkor Kivon(x,y,z)
                      különben Kivon(y,x,z)
                               z.előjel:='-'
Eljárás vége.
```

A belső eljárások pozitív számokat használnak!

Nagypontosságú aritmetika: egész számok



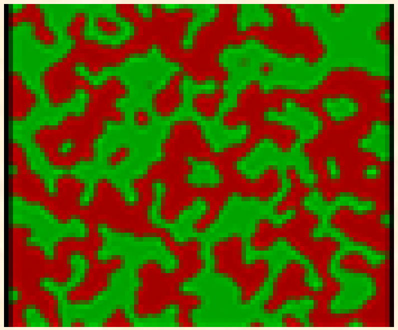


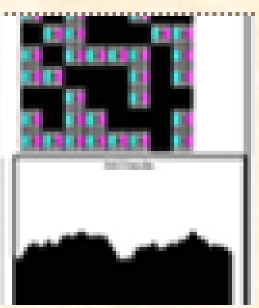
Összeadás, ahogyan az iskolában tanultuk!

```
   XXXXXXXXX
+       XXXX
 ___________
   XXXXXXXXX
```

```
Összead(x,y,z):
  ÁT:=0
  Ciklus i=0-tól y.N-ig
    a:=(x.t(i)+y.t(i)+ÁT)
    z.t(i):=a mod x.S; ÁT:=a div x.S
  Ciklus vége
  …
```

Nagypontosságú aritmetika: egész számok

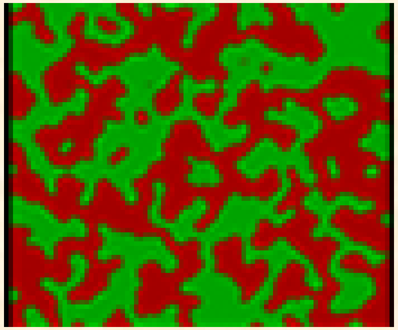


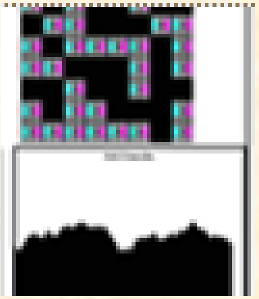


```
  …
  i:=y.N+1
  Ciklus amíg ÁT≠0
    a:=(x.t(i)+ÁT)
    z.t(i):=a mod x.S; ÁT:=a div x.S; i:=i+1
  Ciklus vége
  Ciklus amíg i≤x.N
    z.t(i):=x.t(i); i:=i+1
  Ciklus vége
  z.N:=i-1
Eljárás vége.
```

Probléma: a `div` és a `mod` művelet lassú!

Nagypontosságú aritmetika: egész számok





Felhasználhatjuk, hogy az átvitel maximum 1 lehet, így az osztást elkerülhetjük.

A megoldás:

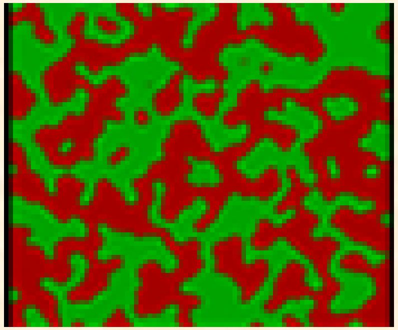
```
a:=(x.t(i)+y.t(i)+ÁT)
z.t(i):=a mod x.S; ÁT:=a div x.S
```

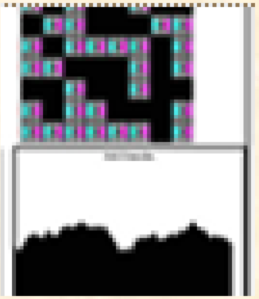
helyett

```
a:=(x.t(i)+y.t(i)+ÁT)
Ha a<x.S akkor z.t(i):=a; ÁT:=0
         különben z.t(i):=a-x.S; ÁT:=1
```

Ugyanezt a másik ciklusmaggal is megtehetjük.

Nagypontosságú aritmetika: egész számok

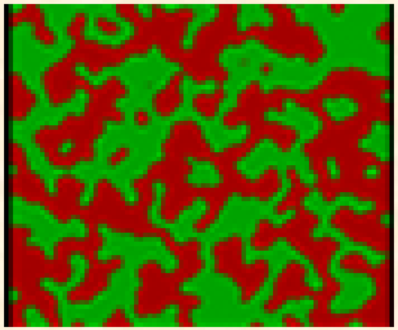


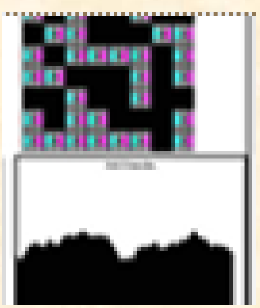


Megoldásváltozatok:

- kivonás – z:=x-y;
- eggyel növelés – z:=x+1;
- eggyel csökkentés – z:=x-1;
- növelés helyben – x:=x+y;
- csökkentés helyben – x:=x-y;
- eggyel növelés helyben – x:=x+1;
- eggyel csökkentés helyben – x:=x-1;
- …

Nagypontosságú aritmetika: egész számok





Szorzás, ahogyan az iskolában tanultuk!

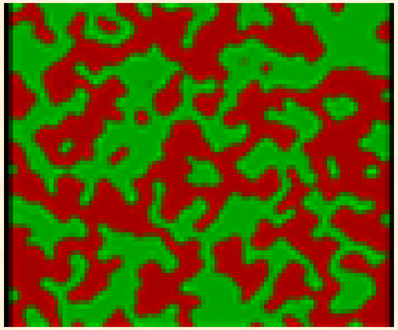
```
          XXXX*XXX
         XXXXX
        XXXXX
       XXXXX
      XXXXXXXX
```

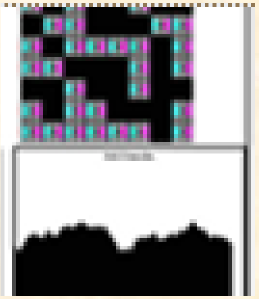
Probléma: a közbülső adatoknak mennyi helyet kell foglalnunk?

Megoldás: amint egy sort kiszámoltunk, azonnal hozzáadhatjuk az eredményhez.

Újabb probléma: az eredmény minden számjegyével sok műveletet végzünk.

Nagypontosságú aritmetika: egész számok





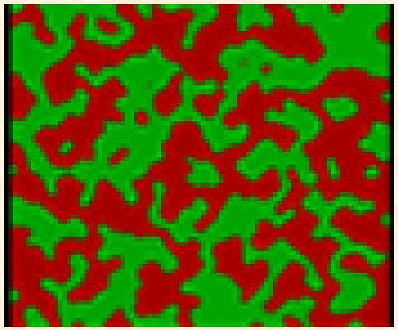
Megoldás egy különleges sorrendű kiszámolás!

$$
\begin{array}{r}
\underline{\mathtt{X\ X\ X\ X}}\mathtt{*XXX} \\
\mathtt{XXX_4X_2X_1} \\
\mathtt{XXXX_5X_3} \\
\underline{\mathtt{XXXXX_6}} \\
\mathtt{XXXXXX\ X\ X}
\end{array}
$$

Amikor egy oszlopot számolunk, azt azonnal adjuk hozzá a végeredményhez!

$$z.t(i) := \left(\sum_{j+k=i} x.t(j) * y.t(k) + ÁT \right) \quad \text{mod} \quad x.S$$

Nagypontosságú aritmetika: egész számok

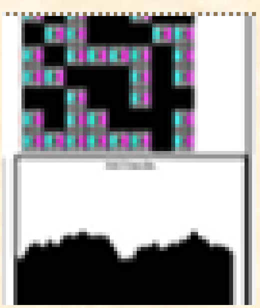


A képlet átfogalmazva:

$$z.t(i) := \left(\sum_{j=?}^{?} x.t(j) * y.t(i-j) + \acute{A}T \right) \quad \mathrm{mod} \quad x.S$$

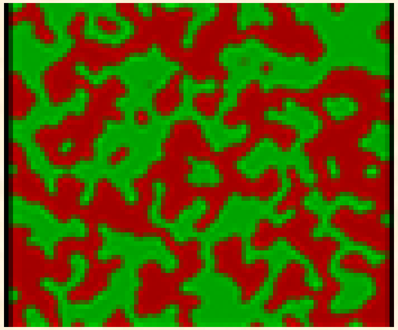
Az indexhatárok számítása:

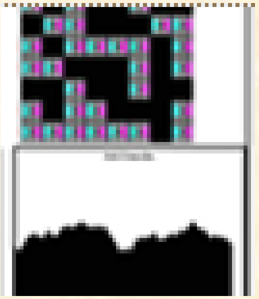
$$0 \le j \le x.N \text{ és } 0 \le i-j \le y.N$$



$$\rightarrow \quad z.t(i) := \left(\sum_{j=\max(0,i-y.N)}^{\min(i,x.N)} x.t(j) * y.t(i-j) + \acute{A}T \right) \quad \mathrm{mod} \quad x.S$$

Nagypontosságú aritmetika: egész számok

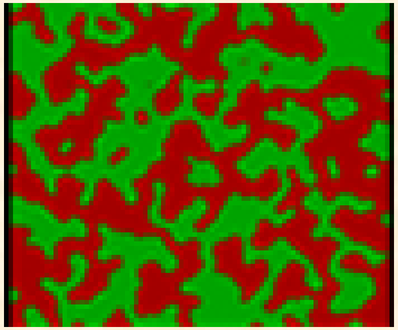




```
Szoroz(x,y,z):
  Ciklus i=0-tól x.N+y.N+1-ig
    a:=ÁT
    Ciklus j=max(0,i-y.N)-től min(i,x.N)-ig
      a:=a+x.t(j)*y.t(i-j)
    Ciklus vége
    z.t(i):=a mod x.S; ÁT:=a div x.S
  Ciklus vége
  Ha ÁT=0 akkor z.N:=x.N+y.N
       különben z.N:=x.N+y.N+1; z.t(z.N):=ÁT
Eljárás vége.
```

Megjegyzés: az alapszám hatványaival szorzást persze nem így oldjuk meg, hanem eltolással.

Nagypontosságú aritmetika: egész számok

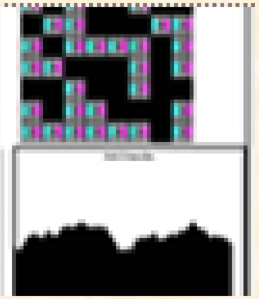


Felezéses szorzási algoritmus kettes számrendszerhez:

$$A*B=\begin{cases} A & \text{ha} \quad B=1 \\ (2*A)*(B/2) & \text{ha} \quad B \quad \text{páros} \\ A+A*(B-1) & \text{ha} \quad B \quad \text{páratlan} \end{cases}$$

Felezéses hatványozási algoritmus kettes számrendszerhez:

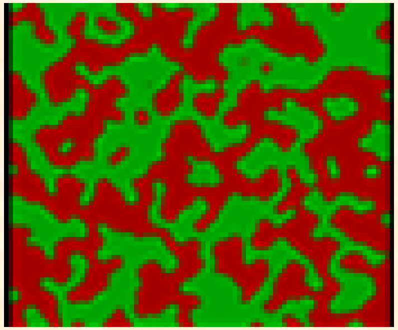
$$A\wedge B=\begin{cases} A & \text{ha} \quad B=1 \\ (A*A)\wedge(B/2) & \text{ha} \quad B \quad \text{páros} \\ A*A\wedge(B-1) & \text{ha} \quad B \quad \text{páratlan} \end{cases}$$

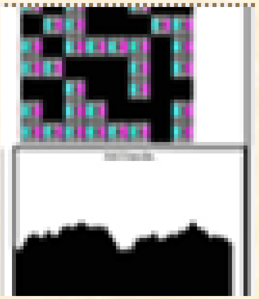


Kérdések:

- páros-páratlan?
- 2*A, B/2?
- B-1?

Nagypontosságú aritmetika: egész számok



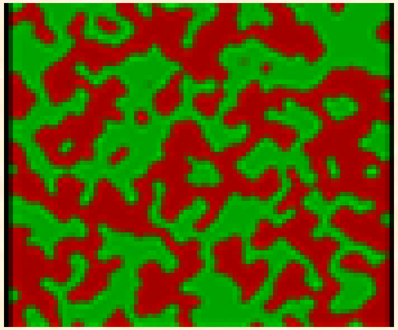


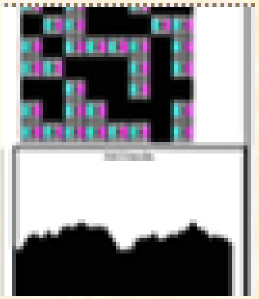
A megoldás:

```
Szoroz2(A,B,C):
  C:=0
  Ciklus amíg B≠0
    Ha páros(B) akkor Balraléptet(A)
                      Jobbraléptet(B)
             különben Össze(C,A)
                      Csökkent(B)
  Ciklus vége
Eljárás vége.
```

Hasonló a hatványozási algoritmus is!

Nagypontosságú aritmetika: egész számok





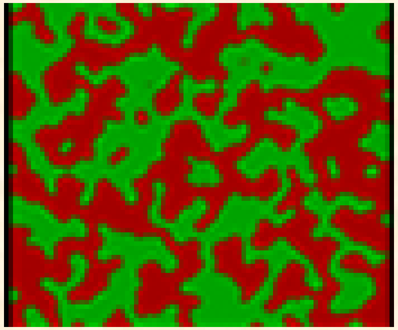
Osztás (C:=A div B, D:=A mod B)

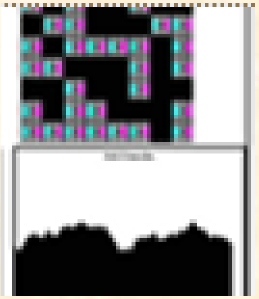
Naiv módszerrel:

```
Osztás(A,B,C,D):
  C:=0; D:=A
  Ciklus amíg D≥B
    C:=C+1; D:=D-B
  Ciklus vége
Eljárás vége.
```

Jól működne, de mennyi idő alatt?

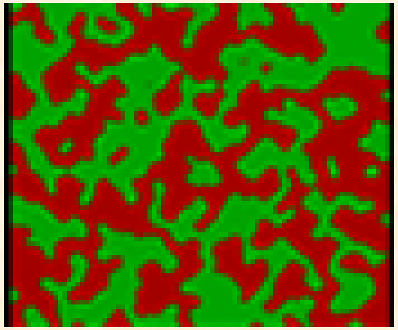
Nagypontosságú aritmetika: egész számok

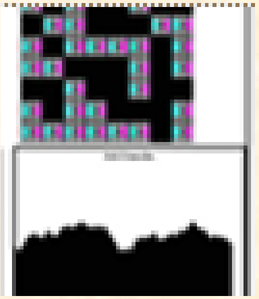




```
   XXXXXXXXX
-          XXXX
-          XXXX
   ...        ←
   XXXXXXXXX
-  XXXX
-  XXXX
   ...        →
   0XXXXXXXX
-   XXXX
   ...
```

Nagypontosságú aritmetika: egész számok



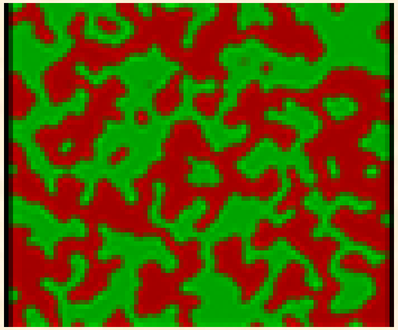


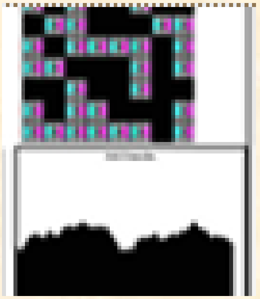
Osztás (C:=A div B, D:=A mod B)

Az osztandó balra tolása, kivonása, majd 1-1 lépéssel jobbra tolása a következő kivonások előtt:

```
Osztás(A,B,C,D):
  K:=A.n-B.n; D:=A; C:=0
  v:=Balraléptet(B,K);
  Ciklus i=1-től K-ig
    Balraléptet(C)
    Ciklus amíg D≤v
      C:=C+1; D:=D-v
    Ciklus vége
    Jobbraléptet(v)
  Ciklus vége
Eljárás vége.
```

Nagypontosságú aritmetika: egész számok





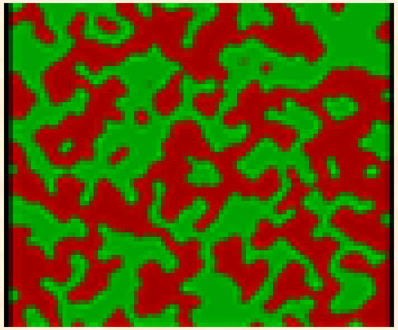
Hányados becslésével, visszavezetés "n+1-jegyű osztása n-jegyűvel" esetre: $(u_0,u_1,\ldots u_{n+1})/(v_0,\ldots,v_n)$. A H hányados Q becslése a következő ($v \geq S/2$ esetén): $Q := \dfrac{Su_{n+1}+u_n}{v_n}$, így Q=H vagy H+1 vagy H+2.

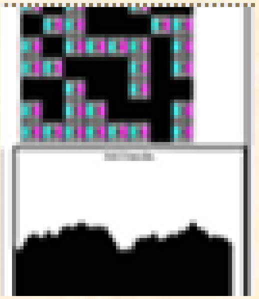
Egy gyors ellenőrzési lehetőség:

$$v_{n-1}*Q>(S*u_{n+1}+u_n-Q*v_n)*S+u_{n-1}$$

Ha igaz, akkor Q:=Q–1 (esetleg kétszer)

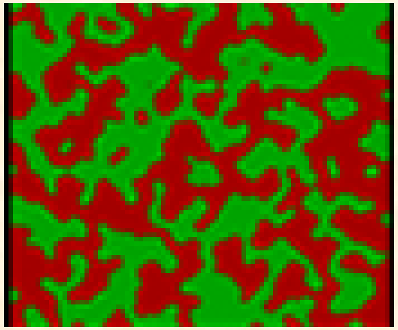
Nagypontosságú aritmetika: egész számok

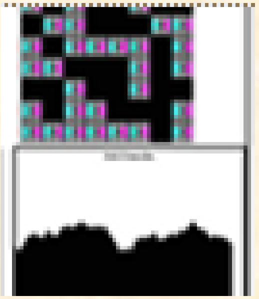




```
Osztás1(…):
  Ha U(N+1)=V(N) akkor  Q:=S-1
    különben Q:=(S*U(N+1)+U(N))/V(N)
  Ciklus amíg V(N-1)*Q>
              (S*U(N+1)+U(N)-Q*V(N))*S+U(N-1)
    Q:=Q-1
  Ciklus vége
  W():=V()*Q
  U():=U()-W()
  Ha U()<0 akkor U():=U()+V(): Q:=Q-1
  H:=Q
Eljárás vége.
```

Nagypontosságú aritmetika: egész számok





Relációk: kisebb (<) reláció

```
Kisebb(A,B):
  Ha A.előjel≠B.előjel
     akkor Kisebb:=A.előjel='-'
  különben
     Ha A.előjel='-' akkor csere(A,B)
     Ha A.n≠B.n akkor Kisebb:=A.n<B.n
     különben i:=A.n
              Ciklus amíg i≥0 és
                          A.t[i]=B.t[i]
                i:=i-1
              Ciklus vége
              Kisebb:=i≥0 és A.t[i]<B.t[i]
     Elágazás vége
  Elágazás vége
Eljárás vége.
```


Vége