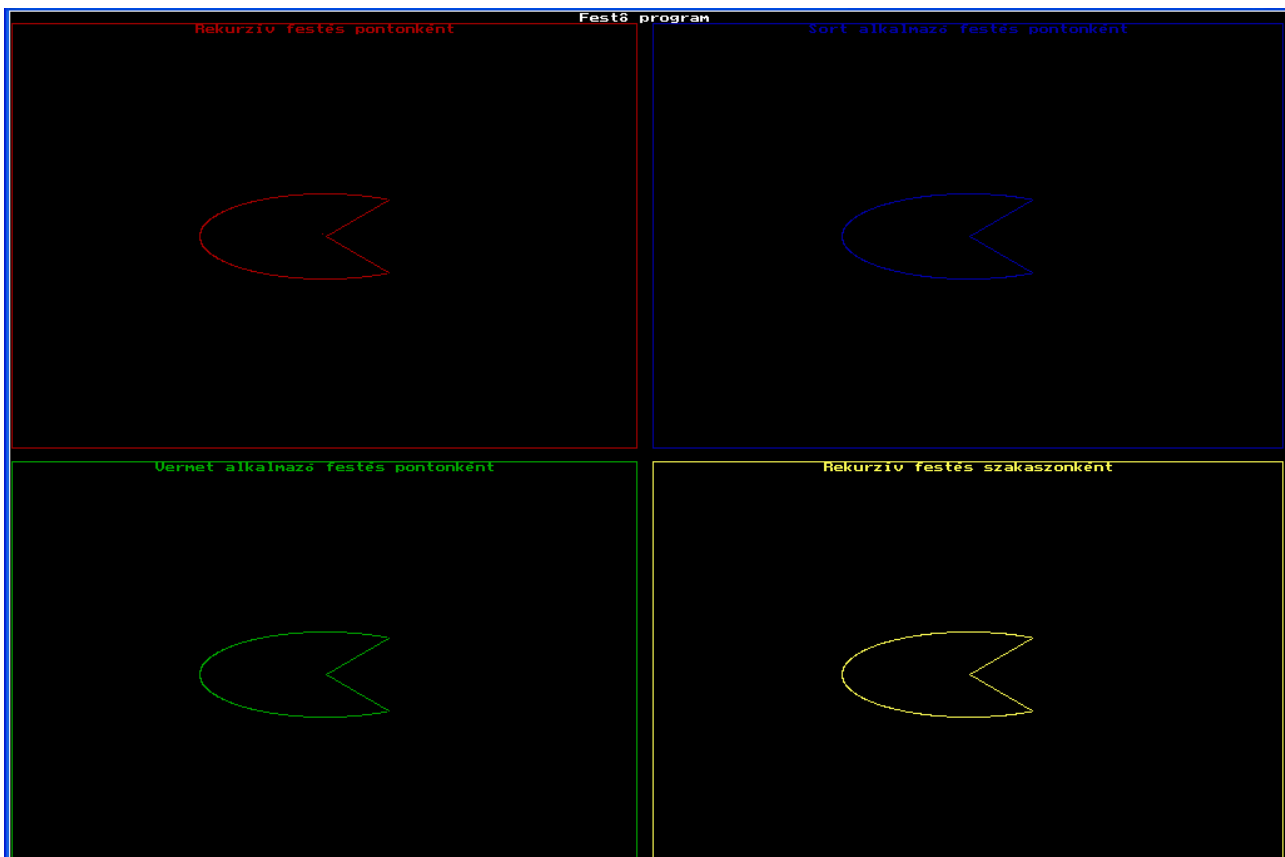


# TARTOMÁNYFESTÉS

## TARTALOM

A feladat.....	2
Rekurzív festés „pontonként” .....	3
Sort alkalmazó festés „pontonként” .....	3
Vermet alkalmazó festés „pontonként” .....	4
Rekurzív festés „szakaszonként” .....	5
„Technikai” tanácsok.....	6

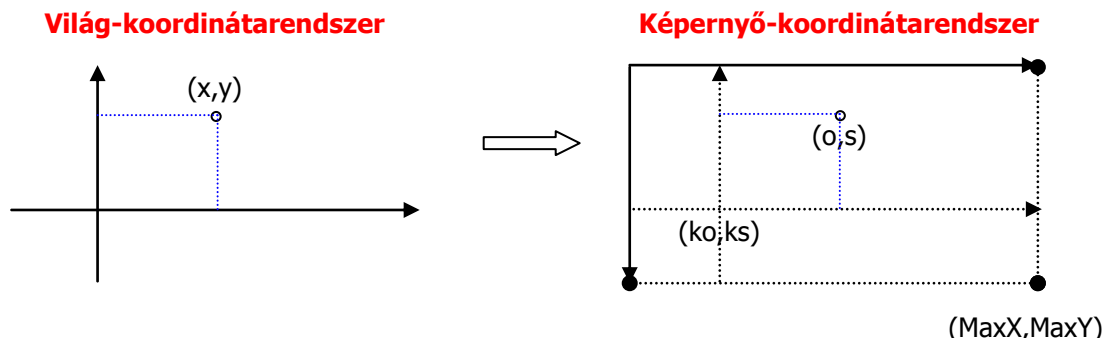


A program nyitóképernyője

## A FELADAT

Adott színű (AktSzín), tetszőleges görbével határolt, zárt tartomány kiszínezése, ismert belső pontból kiindulva.

A koordináta-rendszer megint a szokásos:



Megjegyzem: e feladatkörben nincs sok szerepe a világ-koordináta-rendszernek, hiszen tisztán a képernyőn található zárt tartománnyal kell valamit tenni, ezért nem is foglalkozunk az egyébként szokásos koordináta-transzformációval (sem a tükrözéssel, sem az eltolással vagy a léptékezéssel). Éppen e „nemtörődömség” miatt a programunkban a *főnt és lent megfordul* a szemléletünkhöz képest.

A tényleges rajzolást a PontKi eljárás végzi, amely felhasználja a képernyőre pontot rajzoló Pont utasítást (Turbo grafikában: PutPixel<sup>1</sup>):

**Eljárás** PontKi (**Konstans** x, y: Egész) :  
**Ha**  $y \in [0, \text{MaxY}]$  **és**  $x \in [0, \text{MaxX}]$  **akkor** Pont(x, y)<sup>2</sup>  
**Eljárás vége.**

Fontos szerepet kap egy olyan függvény is, amely egy pontról eldönti, hogy belső pont-e vagy határpont-e. Üres (ez felhasználja a képernyő adott pontjának színét kiolvasó PontSzín függvényt, Turbo grafikában: GetPixel):

**Függvény** Üres (**Konstans** x, y: Egész) : Logikai  
 Üres := PontSzín(x, y) ≠ AktSzín  
**Függvény vége.**

Talán hasznos, ha a feladat megoldása előtt rápillant egy lehetséges megoldást jelentő [program futására](#).

<sup>1</sup> Ha ki akarjuk használni a Turbo grafika különféle rajzoló módjait (WriteMode), akkor a pontrajzolásra is a Line művelet alkalmazandó!

<sup>2</sup> Itt lenne szerepe a koordináta-transzformációnak, amit most elmismásolunk.

## REKURZÍV FESTÉS „PONTONKÉNT”

```

Eljárás RekPont (Konstans x,y:Egész) :
  PontRajzolás(x,y)
  Ha Üres(x-1,y) akkor RekPont(x-1,y)
  Ha Üres(x,y-1) akkor RekPont(x,y-1)
  Ha Üres(x+1,y) akkor RekPont(x+1,y)
  Ha Üres(x,y+1) akkor RekPont(x,y+1)
Eljárás vége.

```

Érdeemes meggondolni, *mekkora hívási verem* szükséges (maximálisan) a rekurzió során.



## SORT ALKALMAZÓ FESTÉS „PONTONKÉNT”

Az alkalmazható sorfogalmak, -műveletek:

```

Típus      TSorElem=Rekord(x,y:Egész)

Eljárás    SorUres{legyen} (Változó s:TSor)
Függvény  UresSor{?} (Konstans s:TSor):Logikai
Eljárás    Sorba(Konstans elem:TSorElem, Változó s:TSor)
Eljárás    Sorbol(Változó s:TSor, elem:TSorElem)
Függvény  SorHossz(Konstans s:TSor):Egész

Eljárás    SorPont(Konstans x,y:Egész) :
  Változó
    s:TSor
    se:TSorElem

```

```

SorÜres(s)
PontRajzolás(x,y); Sorba(TSorElem(x,y),s)
Ciklus amíg nem ÜresSor{?}(s)
  Sorból(s,se); x:=se.x; y:=se.y
  Ha Üres(x-1,y) akkor
    PontRajzolás(x-1,y); Sorba(TSorElem(x-1,y),s)
  Ha Üres(x,y-1) akkor
    PontRajzolás(x,y-1); Sorba(TSorElem(x,y-1),s)
  Ha Üres(x+1,y) akkor
    PontRajzolás(x+1,y); Sorba(TSorElem(x+1,y),s)
  Ha Üres(x,y+1) akkor
    PontRajzolás(x,y+1); Sorba(TSorElem(x,y+1),s)
Ciklus vége
Eljárás vége.

```

Hasonlítsa össze a szükséges *sorméretet* az előbbi módszerbeli veremmérettel (rekurzív hívás-számmal)!



## VERMET ALKALMAZÓ FESTÉS „PONTONKÉNT”

Az alkalmazható veremfogalmak, -műveletek:

**Típus** TVeremElem=**Rekord**(x,y:Egész)

**Eljárás** VeremUres{legyen}(**Változó** v:TVerem)

**Függvény** UresVerem{?}(**Konstans** v:TVerem):Logikai

**Eljárás** Verembe(**Konstans** elem:TVeremElem, **Változó** v:TVerem)

**Eljárás** Verembol(**Változó** v:Verem, elem:TVeremElem)

**Függvény** VeremMelyseg(**Konstans** v:Verem):Egész

**Eljárás** VeremPont (Konstans  $x, y$ :Egész) :

**Változó**

$v$ :TVerem

$ve$ :TVeremElem

VeremÜres ( $v$ )

PontRajzolás ( $x, y$ ) ; Verembe (TVeremElem ( $x, y$ ),  $v$ )

**Ciklus amíg nem** ÜresVerem{?} ( $v$ )

Veremből ( $v, ve$ ) ;  $x := ve.x$  ;  $y := ve.y$

**Ha** Üres ( $x-1, y$ ) **akkor**

PontRajzolás ( $x-1, y$ ) ; Verembe (TVeremElem ( $x-1, y$ ),  $v$ )

**Ha** Üres ( $x, y-1$ ) **akkor**

PontRajzolás ( $x, y-1$ ) ; Verembe (TVeremElem ( $x, y-1$ ),  $v$ )

**Ha** Üres ( $x+1, y$ ) **akkor**

PontRajzolás ( $x+1, y$ ) ; Verembe (TVeremElem ( $x+1, y$ ),  $v$ )

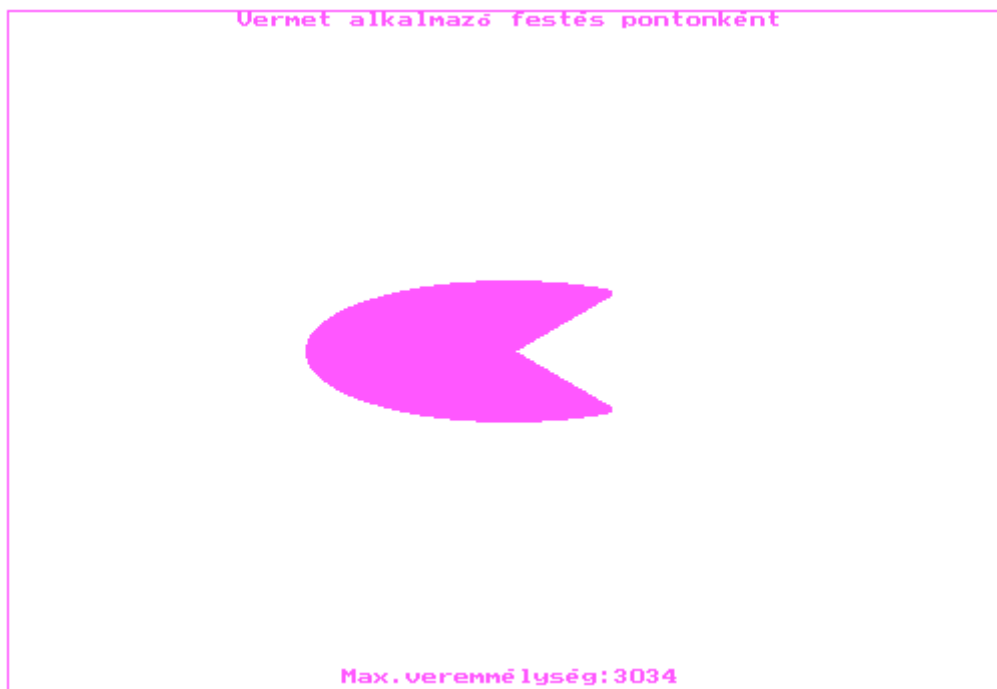
**Ha** Üres ( $x, y+1$ ) **akkor**

PontRajzolás ( $x, y+1$ ) ; Verembe (TVeremElem ( $x, y+1$ ),  $v$ )

**Ciklus vége**

**Eljárás vége.**

Itt is érdekes látni a ténylegesen megkívánt *veremmélységet*.



## REKURZÍV FESTÉS „SZAKASZONKÉNT”

**Eljárás** RekSzakasz (Konstans  $x, y$ :Egész) :

**Változó**

$i, y_1, y_2$ :Egész

Vonalvég ( $-1, x, y, y_1$ ) ; Vonalgég ( $+1, x, y, y_2$ )

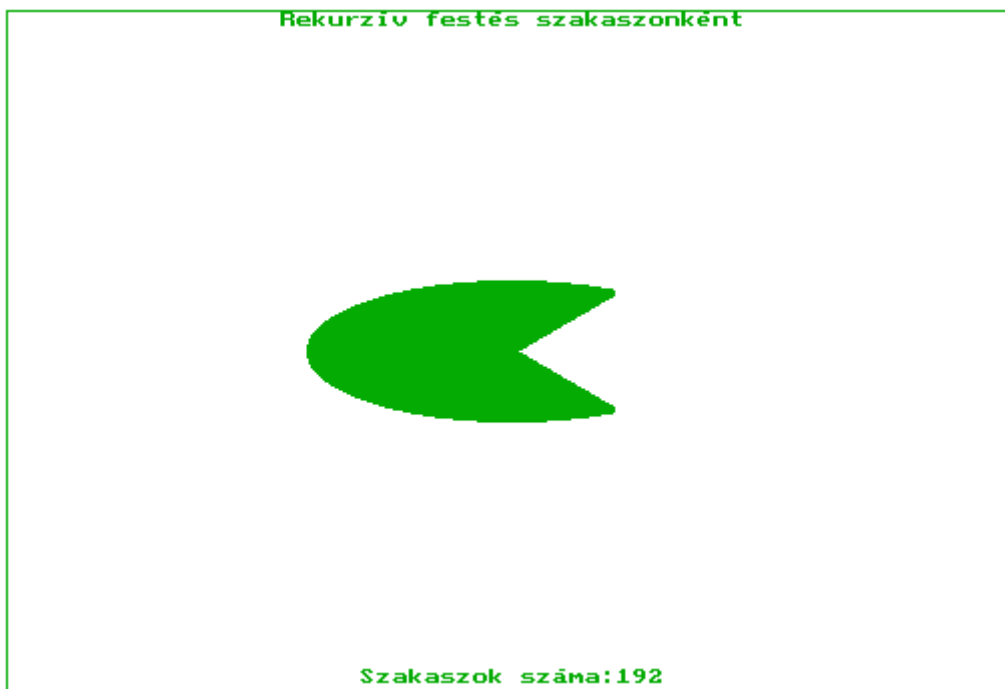
SzakaszKi ( $x, y_1, x, y_2$ )

```

Ciklus i=y1-tól y2-ig
  Ha Üres(x-1,i) akkor RekSzakasz(x-1,i)
  Ha Üres(x+1,i) akkor RekSzakasz(x+1,i)
Ciklus vége
Eljárás vége.

Eljárás Vonalvég(Konstans merre,x,y:Egész, Változó z:Egész):
  z:=y
  Ciklus amíg Üres(x,z+merre)
    z:+merre
  Ciklus vége
Eljárás vége.

```



## „TECHNIKAI” TANÁCSOK

A szükséges kellékek (keretprogram, verem és sor, esetleg graph unit, ill. egavga.bgi driver) letöltéséhez látogasson el a <http://people.inf.elte.hu/szlavi/Grafika/Festések> könyvtárba.