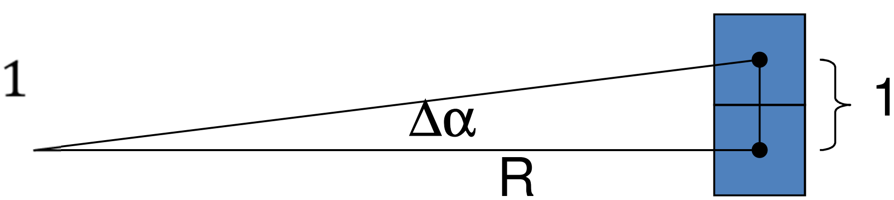
Računarska grafika - vežbe

9 – Rasterizacija kružnice

Rasterizacija kružnice (1/10)

- Pretpostavka:
  - kružnica se crta u koordinatnom početku
- Trivijalan algoritam
  - $x = R\cos(\alpha)$, $y = R\sin(\alpha)$, $\alpha \in [0,2\pi]$
  - menjajući α u malim koracima,
    dobija se skup tačaka koje aproksimiraju kružnicu
- Loše osobine:
  - trigonometrijske funkcije su spore
  - korak $\Delta\alpha$ je inverzno proporcionalan vrednosti R

$$\Delta\alpha = arctg\left(\frac{1}{R}\right) \approx \frac{1}{R}, \text{za } R \gg 1$$



Rasterizacija kružnice (2/10)

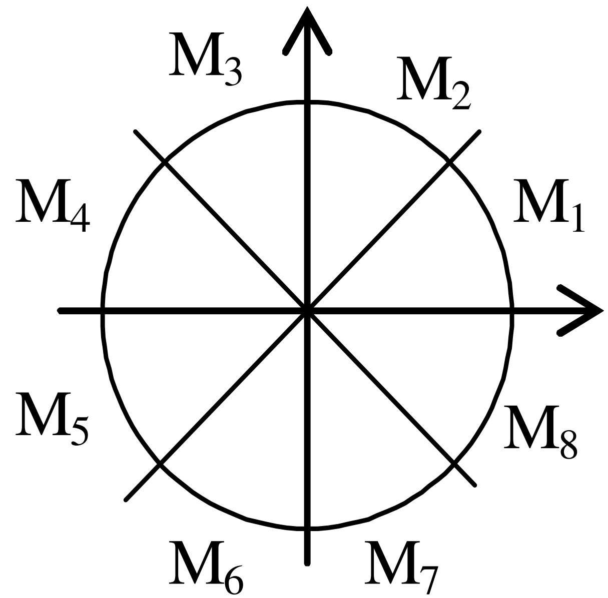
- Kružnica sa centrom u tački (cx,cy) se crta funkcijom:

```
void kruznica(int cx, int cy, int r){
    float deltaAlfa = atan(1.0f/r);
    for(float alfa=0; alfa<6.28; alfa+=deltaAlfa)
        crtajTacku(cx+r*cos(alfa),cy+r*sin(alfa));
}
```

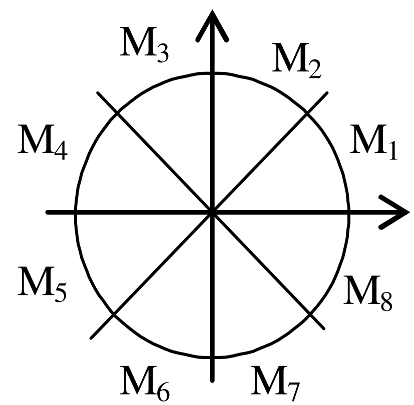
Rasterizacija kružnice (3/10)

- Efikasniji algoritam
  - koristi centralnu simetriju kružnice
  - računa koordinate tačaka za 1/8 kružnice, ostale preslikava

$M_1 = (x,y)$	$M_5 = (-x,-y)$
$M_2 = (y,x)$	$M_6 = (-y,-x)$
$M_3 = (-y,x)$	$M_7 = (y,-x)$
$M_4 = (-x,y)$	$M_8 = (x,-y)$



Rasterizacija kružnice (4/10)

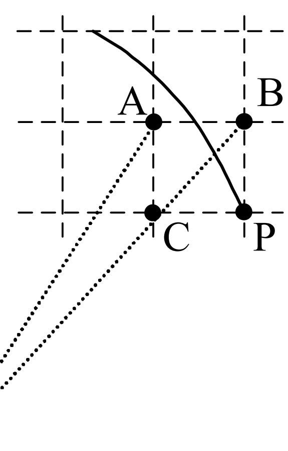



```
void crtaj8Tacaka(float cx, float  cy, float  dx, float dy){
   crtajTacku(cx+dx, cy+dy); // M1
   crtajTacku(cx-dx, cy+dy); // M4
   crtajTacku(cx+dx, cy-dy); // M8
   crtajTacku(cx-dx, cy-dy); // M5
   crtajTacku(cx+dy, cy+dx); // M2
   crtajTacku(cx-dy, cy+dx); // M3
   crtajTacku(cx+dy, cy-dx); // M7
   crtajTacku(cx-dy, cy-dx); // M6
}

void kruznica(int cx, int cy, int r) {
   float deltaAlfa = atan(1.0f/r);
   for(float alfa=0; alfa<0.79; alfa+=deltaAlfa)
        crtaj8Tacaka(cx, cy, r*cos(alfa), r*sin(alfa));
}
```

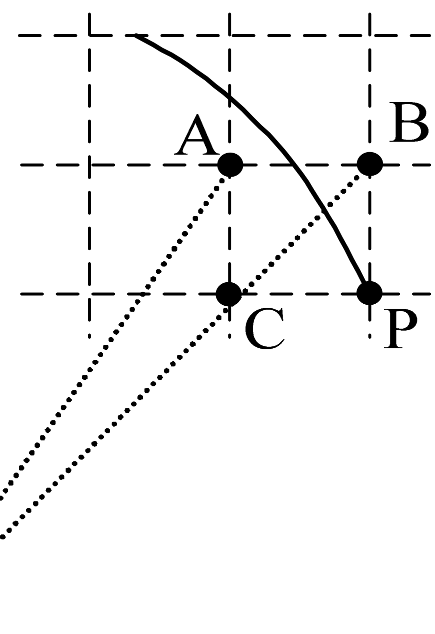
Rasterizacija kružnice (5/10)

- Bresenham/Michener-ov algoritam
  - posmatra se prvi oktant kružnice
  - počinje tačkom (R, 0) i napreduje
    u smeru nasuprot kretanja kazaljke časovnika, do ugla 45°.
  - prolazi kroz tačku P
  - odlučuje se o izboru između dve kandidat-tačke (A i B)
    između kojih prolazi kružna linija

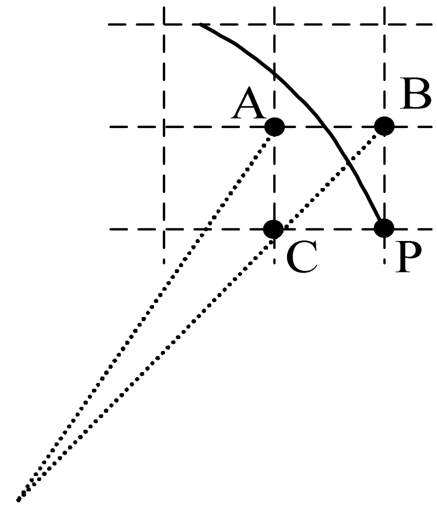


Rasterizacija kružnice (6/10)

- Pretpostavka: kružnica je prošla kroz tačku P(x,y)
- Treba odlučiti da li će “proći” kroz A(x-1, y+1) ili B(x, y+1)
- Da li je moguće da “prođe” kroz C?
  - ne, jer posmatrani luk pripada prvom oktantu



Rasterizacija kružnice (7/10)



- Uvodi se pojam kvadradtne greške u tački K
  - odstupanje u odnosu na geometrijski tačnu kružnicu

    $e(K) = x_k^2 + y_k^2 - R^2$

    $e(K) > 0$, za K izvan kruga

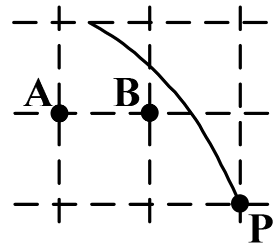
    $e(K) = 0$, za K na kružnici

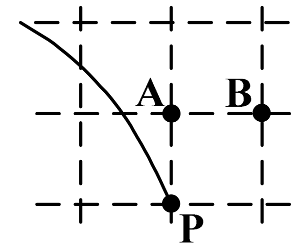
    $e(K) < 0$, za K unutar kruga

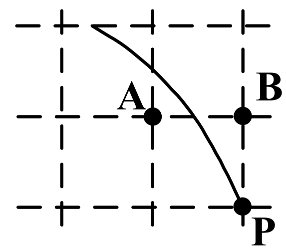
U tački A (e<0):	U tački B (e>0):
$e(A) = (x-1)^2 + (y+1)^2 - R^2$	$e(B) = x^2 + (y+1)^2 - R^2$

Rasterizacija kružnice (8/10)

- Algebarski zbir grešaka je: d=e(A) + e(B)
  - A, B unutar kruga (teoretski): e(A)<0, e(B)<0 ⇨ d<0, bira se B
  - A, B izvan kruga (teoretski): e(A)>0, e(B)>0 ⇨ d>0, bira se A
  - A unutar, B izvan kruga: e(A)<0, e(B)>0
    - |e(A)| > |e(B)| ⇨ d<0, bira se B
    - |e(A)| < |e(B)| ⇨ d>0, bira se A
- Uočava se zakonitost:
  d<0 ⇨ bira se B; d>0 ⇨ bira se A
- Ako je izabrano A, dekrementira se X;
  Y se uvek inkrementira





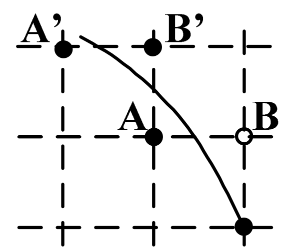


Rasterizacija kružnice (9/10)

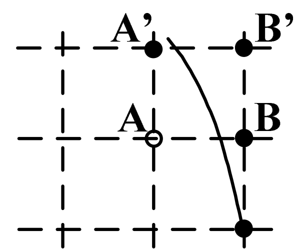
- **U i-tom koraku:**
  - $d_i = (x-1)^2 + (y+1)^2 - R^2 + x^2 + (y+1)^2 - R^2$
    $= 2x^2 + 2y^2 - 2R^2 + 3 - 2x + 4y$

- **U početnom koraku:**
  - $d_0 = d_{i|(R,0)} = \mathbf{3-2R}$

- **U (i+1) koraku:**
  - za $d_i > 0$ (A):
  - $d_{i+1} = (x-2)^2 + (y+2)^2 - R^2 + (x-1)^2 + (y+2)^2 - R^2$
    $= 2x^2 + 2y^2 - 2R^2 + 3 - 2x + 4y + 10 - 4x + 4y$
    $= \mathbf{d_i + 4(y-x) + 10}$



  - za $d_i < 0$ (B)
  - $d_{i+1} = (x-1)^2 + (y+2)^2 - R^2 + x^2 + (y+2)^2 - R^2$
    $= 2x^2 + 2y^2 - 2R^2 + 3 - 2x + 4y + 6 + 4y$
    $= \mathbf{d_i + 4y + 6}$



Rasterizacija kružnice (10/10)

```
void kruznica(int cx, int cy, int r) {
   int d = 3-2*r;
   int x=r, y=0;
   while(x>y) {
        crtaj8Tacaka(cx, cy, x, y);
        if(d<0) d+=4*y+6;
        else {
              d+=4*(y-x)+10;
              x--;
        }
        y++;
   }
   if(x==y) crtaj8Tacaka(cx, cy, x, y);
}
```

Zadatak 1

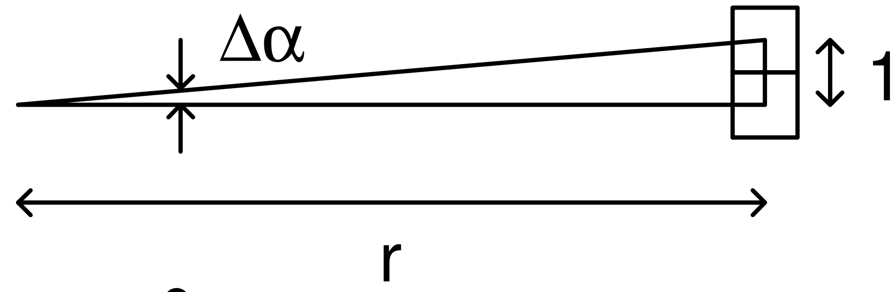
a) U jednostavnom algoritmu za crtanje kružnice uz pomoć trigonometrijskih funkcija izvesti zavisnost koraka ugla $\Delta\alpha$ od poluprečnika kružnice. Korak ugla mora biti maksimalan, ali istovremeno mora da obezbedi neprekidnost kružnice. Kružnica je neprekidna ukoliko je udaljenost svake njene dve susedne tačke (za dva ugla čija je razlika $\Delta\alpha$) po svakoj od koordinata najviše 1. Pretpostaviti da je r dovoljno veliko i da važi: $\sin(\Delta\alpha) = \Delta\alpha$.

b) Naći odnos broja poziva procedure za crtanje tačke jednostavnog trigonometrijskog algoritma kome je korak ugla postavljen kao pod a) i broja poziva procedure za crtanje tačke Bresenham/Michener-ovog algoritma za crtanje kružnice istog poluprečnika. Odnos dati za dovoljno velike poluprečnike kružnice

c) Koji efekat nije uzet u obzir odnosom izračunatim u tački (b), a značajno doprinosi relativnoj prednosti brzine Bresenhamovog algoritma?

Zadatak 1 – rešenje (1/2)

a) $\Delta\alpha = \mathrm{arctg}(1/r) \approx 1/r$



b) Jednostavan algoritam: $n_j = \dfrac{2\pi}{\Delta\alpha} = 2\pi r$

Bresenham/Michener-ov algoritam:
1/8 kruga se crta uz inkrementiranje y, sve dok x<y

$$r^2 = x^2 + y^2 = 2y^2 \Rightarrow y = \frac{r}{\sqrt{2}} \Rightarrow n_b = \frac{8r}{\sqrt{2}}$$

Dakle, odnos je: $\dfrac{n_j}{n_b} = \dfrac{2\pi r}{4\sqrt{2}r} \approx 1.11$

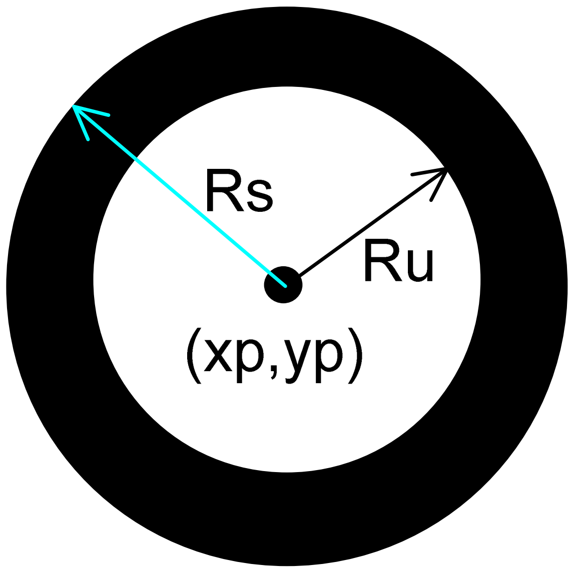
Zadatak 1 – rešenje (2/2)

c) Nije uzeta u obzir činjenica:

Bresenham-ov algoritam radi samo sa celim brojevima, jednostavan algoritam radi sa realnim brojevima i koristi trigonometrijske funkcije

Zadatak 2

- Korišćenjem Bresenhamovog algoritma za crtanje kružnice kao osnove i rutine za crtanje linije **`Line(x1,y1,x2,y2)`** koja crta liniju od tačke (x1,y1) do tačke (x2,y2), napisati program za crtanje popunjenog prstena sa centrom u tački (xp,yp), spoljnog poluprečnika Rs i unutrašnjeg poluprečnika Ru (smatrati da je Rs>Ru).

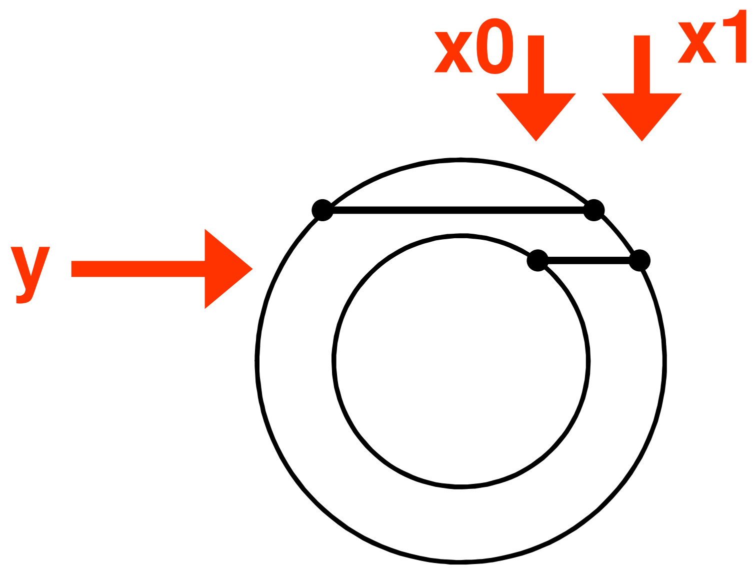


Zadatak 2 – rešenje (1/5)

Ideja: napraviti matricu koordinata `tacke[MaxY][2]`

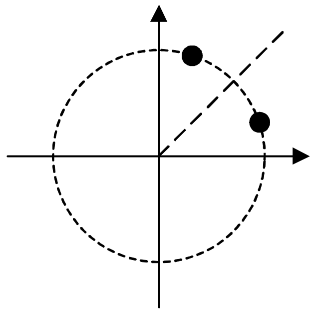
- popunjavati u opsegu unutrašnjeg kruga: `tacke[y][0]`, a zatim spoljašnjeg kruga: `tacke[y][1]`
  - `tacke[y][0]` će ostati ? za y-opseg koji ne obuhvata unutrašnji krug
- u opsegu unutrašnjeg: za dato `y` crta se od `tacke[y][0]` do `tacke[y][1]`
- u opsegu spoljašnjeg: za dato `y` crta se od –`tacke[y][1]` do `tacke[y][1]`

y	x0	x1
0	3	6
1	2	6
...	...	...
MaxY	?	4



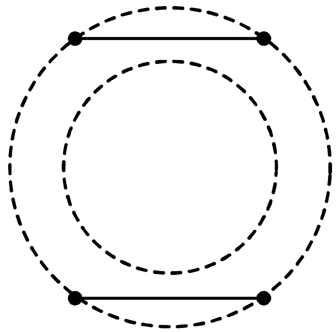
Zadatak 2 – rešenje (2/5)

```
/* smesta u niz izracunatu i njoj simetricnu tacku
   simetrala je prava y=x */
void ubaci2(int tacke[][], int index,
                 int x, int y) {
  tacke[y][index] = x;
  tacke[x][index] = y;
}
```

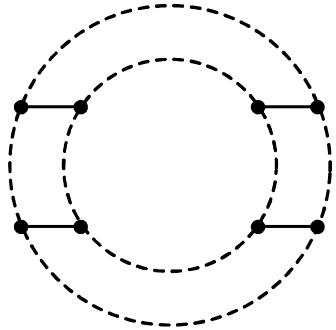


Zadatak 2 – rešenje (3/5)

```
void crtaj2(int tacke[][], int y, int xc, int yc){
   Line(xc-tacke[y][1], yc+y,
        xc+tacke[y][1], yc+y);
   Line(xc-tacke[y][1], yc-y,
        xc+tacke[y][1], yc-y);
}
```



```
void crtaj4(int tacke[][], int y, int xc, int yc){
   Line(xc-tacke[y][1], yc+y,
        xc-tacke[y][0], yc+y);
   Line(xc+tacke[y][0], yc+y,
        xc+tacke[y][1], yc+y);
   Line(xc-tacke[y][1], yc-y,
        xc-tacke[y][0], yc-y);
   Line(xc+tacke[y][0], yc-y,
        xc+tacke[y][1], yc-y);
}
```



Zadatak 2 – rešenje (4/5)

```
void prsten(int xp, int yp, int Ru, int Rs){
      int tacke[MAX_INDEX][2];
      int x,y,d;

      // unutrasnji krug
      x=Ru; y=0;
      d=3-2*Ru;
      while(x>=y) {
            ubaci2(tacke,0,x,y);
            if( d < 0 ) d += 4*y+6;
            else {d += 4*(y-x)+10;
                  x--;
            }
            y++;
      }
```

Zadatak 2 – rešenje (5/5)

```
  // spoljasnji krug
  x=Rs; y=0;
  d=3-2*Rs;
  while(x>=y) {
        ubaci2(tacke,1,x,y);
        if(d < 0 )  d += 4*y+6;
        else {d += 4*(y-x)+10;
              x--;
        }
        y++;
  }
  // crtanje do visine unutrasnjeg poluprecnika
  for(y=0; y<=Ru; y++)    crtaj4(tacke, y, xp, yp);
  // crtanje ostatka
  for(y=Ru+1; y<=Rs; y++) crtaj2(tacke, y, xp, yp);
}
```

Primeri ispitnih zadataka