

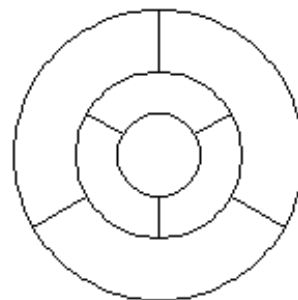
**1. zadatak****ALKA****30 bodova**

Napiši proceduru `ALKA` :a :b :c, koja crta alku kao na slici.

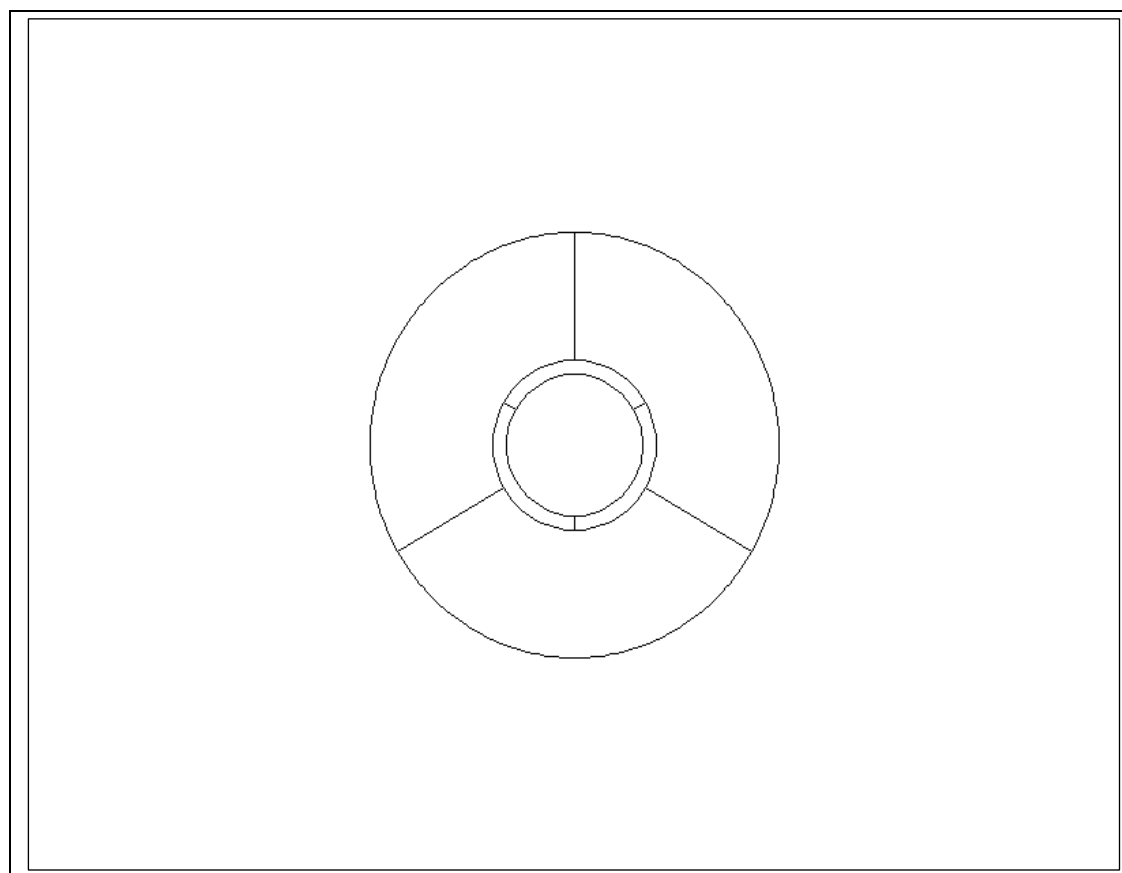
Alka je sastavljena od tri kružnice. Unutarnja i srednja kružnica povezane su s tri linije jednoliko raspoređene u krug. Na sličan način povezane su i srednja i vanjska kružnica, ali obrnuto od unutarnjih linija (vidi sliku desno).

Pozicija lika na ekranu nije bitna, ali je važno da najdonja unutarnja linija i najgornja vanjska linija stoje uspravno ( usporedno s y-osi ).

Radijus unutarnje kružnice je :a, srednje :b, a vanjske :c.



**Primjer:** `cs ALKA 50 60 150`



Program snimate pod imenom **ALKA.LGO**

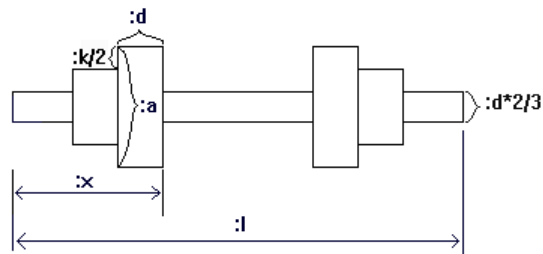
**2. zadatak****UTEG****40 bodova**

Napiši proceduru UTEG :l :x :a :k :d koja crta šipku s utezima.

Šipka je pravokutnik širine :l i visine :d\*2/3. Na udaljenosti :x od oba ruba šipke niže utege prema rubu šipke, i to tako da je prvi uteg visine :a, a svaki sljedeći ima visinu za :k manju od svog prethodnika. Svi utezi su širine :d.

Utege stavljamo sve dok su zadovoljena ova dva uvjeta:

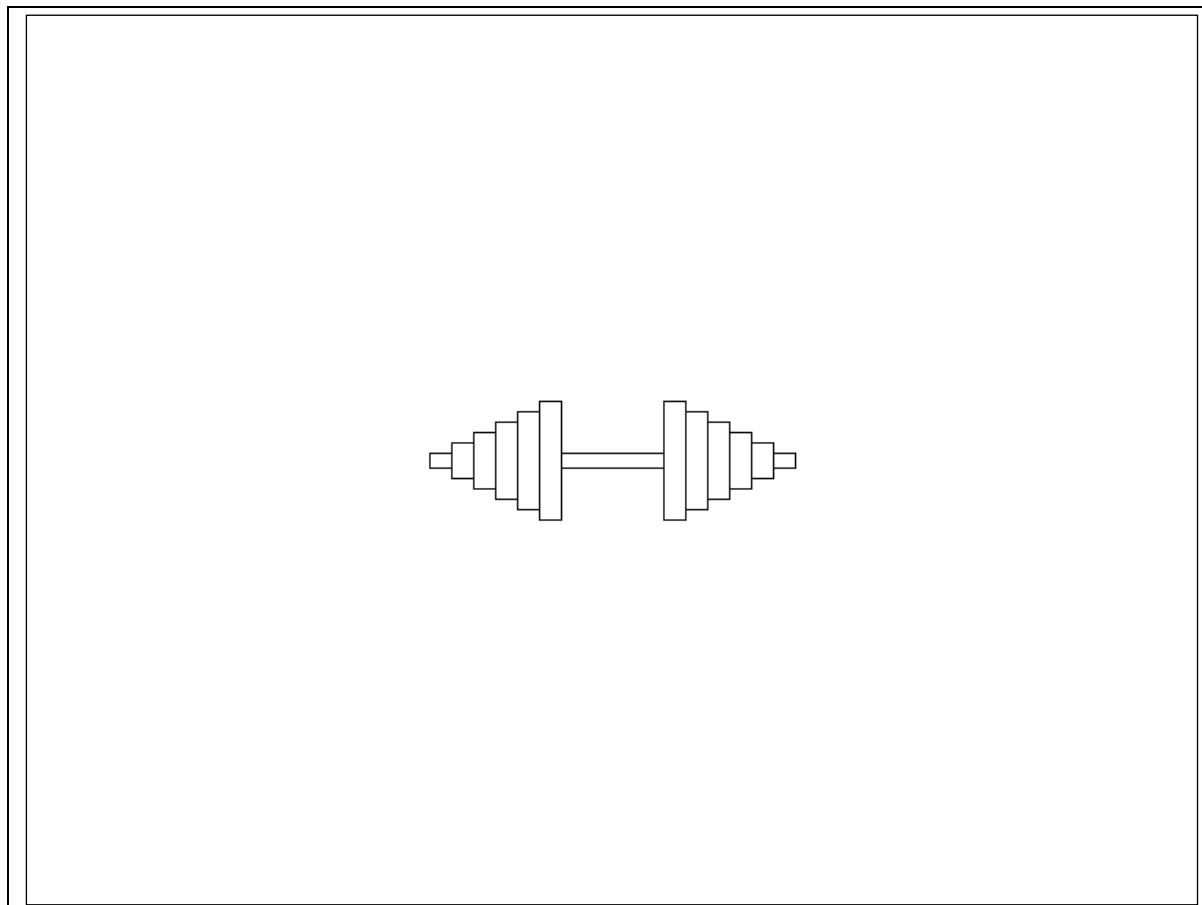
1. Utteg koji stavljamo stane na šipku, tj. komad šipke koji nam je preostao za staviti utteg veći je ili jednak širini uttega
2. Visina uttega koji stavljamo **strogo je veća** od visine šipke



Pozicija i rotacija na ekranu nije bitna.

**Napomena:** :l i :k će biti parni brojevi, dok će :d biti dijeljiv s 3, zbog preciznijeg crtanja.

**Primjer:** cs UTEG 250 90 80 14 15



Program snimite pod imenom **UTEG.LGO**

**3. zadatak**

**POLA**

**60 bodova**

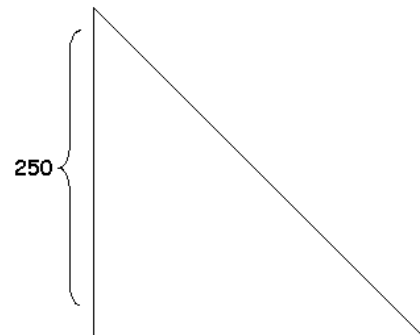
Napiši proceduru `POLA :x :y` koja jednom dužinom podijeli trokut na dva dijela jednake površine.

Početni trokut koji dijelimo je uvijek isti, ima dvije **jednake** stranice, duljine **250**, koje zatvaraju pravi kut.

Vrh pravog kuta je u **središtu ekrana**, a stranice su usporedne s koordinatnim osima.

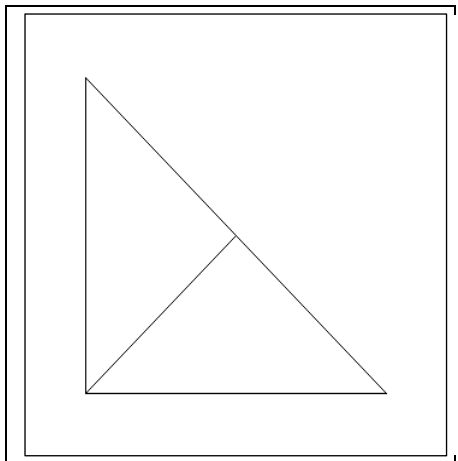
Trokut je potrebno podijeliti na dva dijela jednake površine dužinom kojoj je jedan kraj u točki s zadanim koordinatama `:x` i `:y`, a drugi kraj negdje na stranici trokuta.

Zadana točka **uvijek** će se nalaziti na nekoj stranici trokuta.

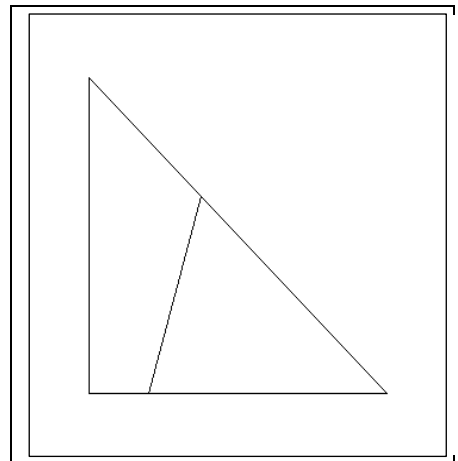


**Primjeri:**

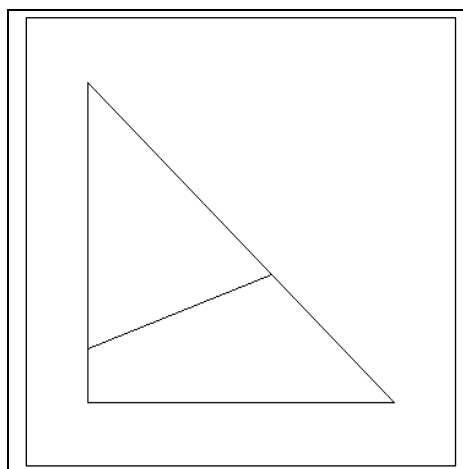
`cs POLA 0 0`



`cs POLA 50 0`



`cs POLA 150 100`



Program snimiti pod imenom **POLA.LGO**

**4. zadatak****KOD****70 bodova**

Mirku je u zadnje vrijeme računalo postalo prepuno velikih datoteka i želi ih svim sredstvima smanjiti. Nakon par dana proučavanja odabrao je algoritam kompresije koji će koristiti, ali treba pomoć već na prvom koraku. Za dani broj pojavljivanja pojedinih simbola u datoteci, potrebno je izračunati zamjenski niz binarnih znamenki. Vas je zamolio da to napravite umjesto njega.

Koraci u postupku određivanja zamjenskih nizova je sljedeći:

1. Dodijeli svakom simbolu oznaku
2. Odaberi dvije oznake takve da je broj pojavljivanja simbola s tim oznakama najmanji mogući (neka su to oznake A i B te je **zbroj** pojavljivanja svih simbola s oznakom A manji od onog s oznakom B)
3. Svim simbolima s oznakom A na kraj zamjenskog niza nalijepi 0, a svim simbolima s oznakom B na kraj zamjenskog niza nalijepi 1
4. Svim simbolima s oznakom B, zamijeni oznaku s oznakom A
5. Ponavljaj sve od koraka 2 dok svi simboli ne poprime istu oznaku
6. Svim simbolima okreni zamjenske nizove

Napiši funkciju `KOD :p` koja prima broj pojavljivanja pojedinog simbola te vraća listu zamjenskih nizova za te simbole. Ulazni podaci će biti takvi da se ni u jednom trenutku neće pojaviti dvije oznake s jednakim ukupnim brojem pojavljivanja simbola s tim oznakama.

Primjer određivanja zamjenskih nizova za neka tri simbola. Neke su pojavljivanja simbola [2 3 4]:

- Prvo se dodjele oznake svakom simbolu [1 2 3]
- Dva puta se ponavljaju koraci 2-5:
  - Odabir oznaka 1 i 2 (A = 1, B = 2)
  - Prvi simbol na kraj niza dobiva 0, a drugi simbol na kraj zamjenskog niza dobiva 1
  - Drugom simbolu se oznaka promijeni u 1, pa su sad oznake [1, 1, 3]
  - Odabir oznaka 1 i 3 (A = 3, B = 1)
  - Treći simbol na kraj niza dobiva 0, a prvi i drugi dobivaju 1 na kraj zamjenskog niza
  - Prvom i drugom simbolu se oznaka promijeni u 3, pa su sad oznake [3, 3, 3]
- Trenutni zamjenski nizovi su [01, 11, 0], i prema koraku 6 potrebni ih je okrenuti, pa su konačni zamjenski nizovi [10, 11, 0].

**Primjer 1.**

```
show kod [ 2 3 4 ]
izlaz: [10 11 0]
```

**Primjer 2.**

```
show kod [ 1 2 4 ]
izlaz: [00 01 1]
```

**Primjer 3.**

```
show kod [1 2 4 5 6]
izlaz: [000 001 01 10 11]
```

Program snimite pod imenom **KOD.LGO**