

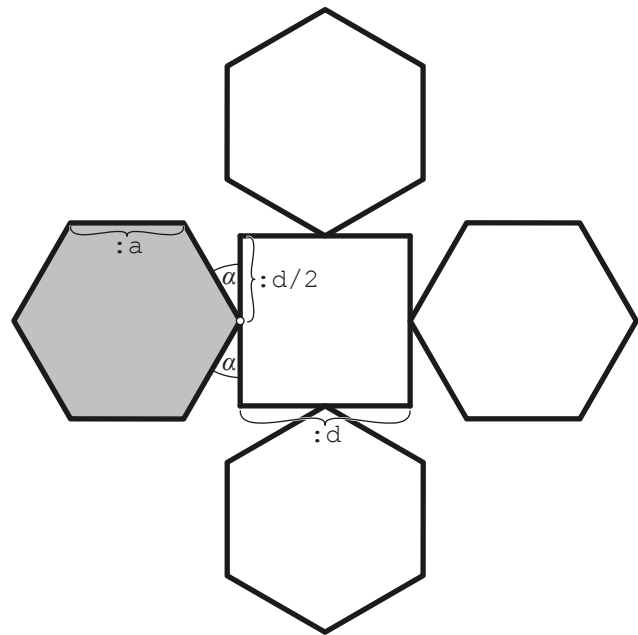
1. zadatak**CVIJET****30 bodova**

Napišite proceduru `CVIJET :n :m :a :d` koja crta *cvijet* kao na slici desno. *Središte cvijeta* je pravilni n -terokut sa stranicama duljine d . Svaku stranicu n -terokuta u polovištu s vanjske strane dodiruje po jedna *latica*. *Latica* je pravilni m -terokut sa stranicama duljine a . Svaka *latica* je tako postavljena, da zatvara jednak kut prema *središtu cvijeta* s obje strane (na slici desno — zatamnjena *latica* i kut α).

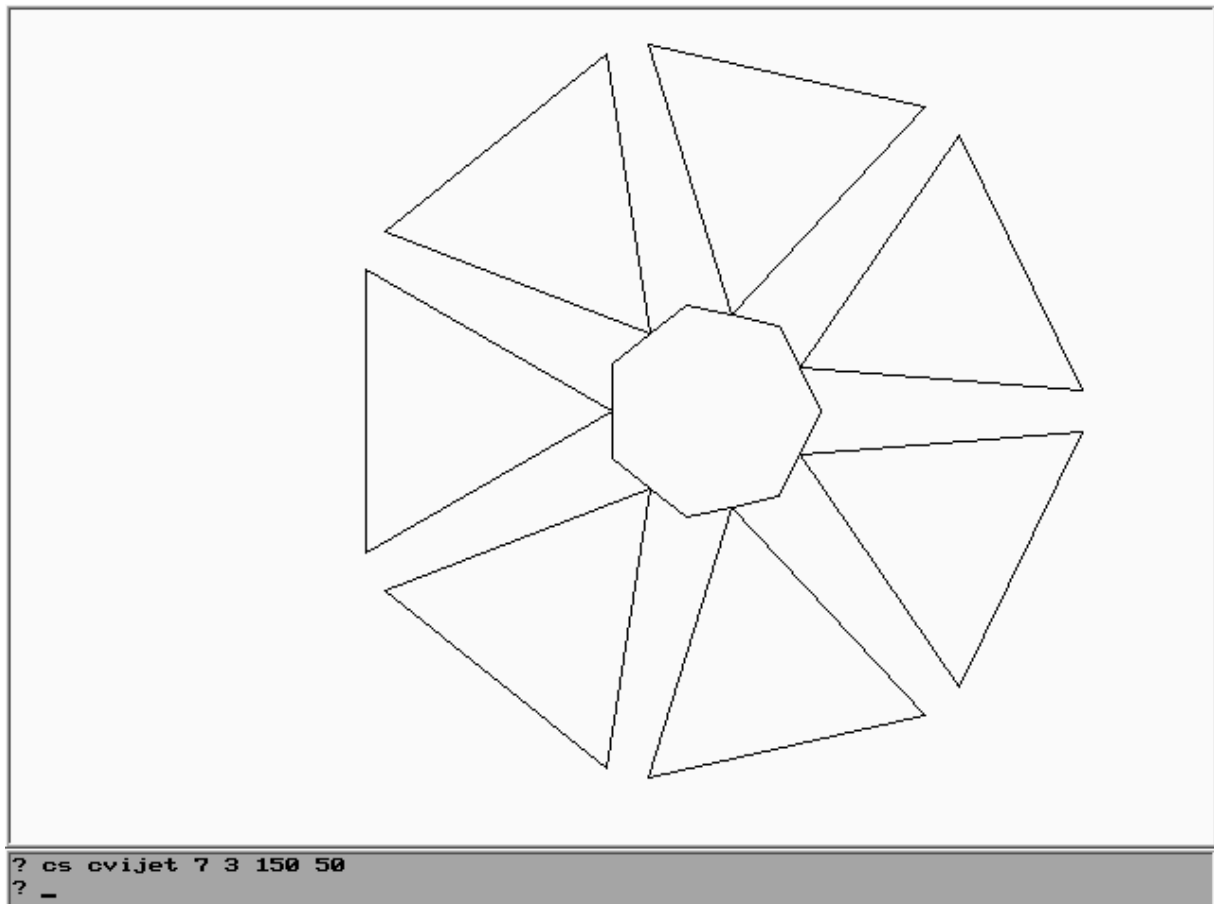
Na slici desno je primjer `CVIJET 4 6 60 90`.

m i n su prirodni brojevi veći od 2.
 a i d su brojevi veći od nule.

Pozicija lika na ekranu nije bitna.



Primjer (vidi sliku dolje): `cs CVIJET 7 3 150 50`



Program snimite pod imenom **CVIJET.LGO**

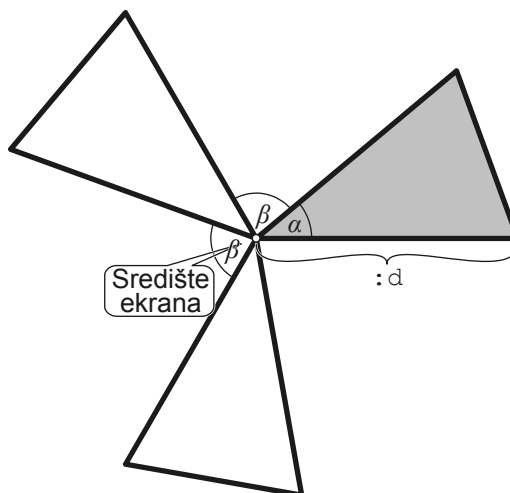
2. zadatak**TROKUTI****40 bodova**

Napišite proceduru `TROKUTI :n :d` koja briše ekran i crta $:n$ jednakih jednakokranih trokuta koji se dodiruju vrhovima suprotnim osnovici (kao na slici desno). Duljina kraka je $:d$. Kut između svaka dva trokuta mora biti jednak (na slici desno kut β), i dvostruko veći od kuta između krakova trokuta (na slici desno kut α).

Na slici desno je primjer kada je $:n = 3$.

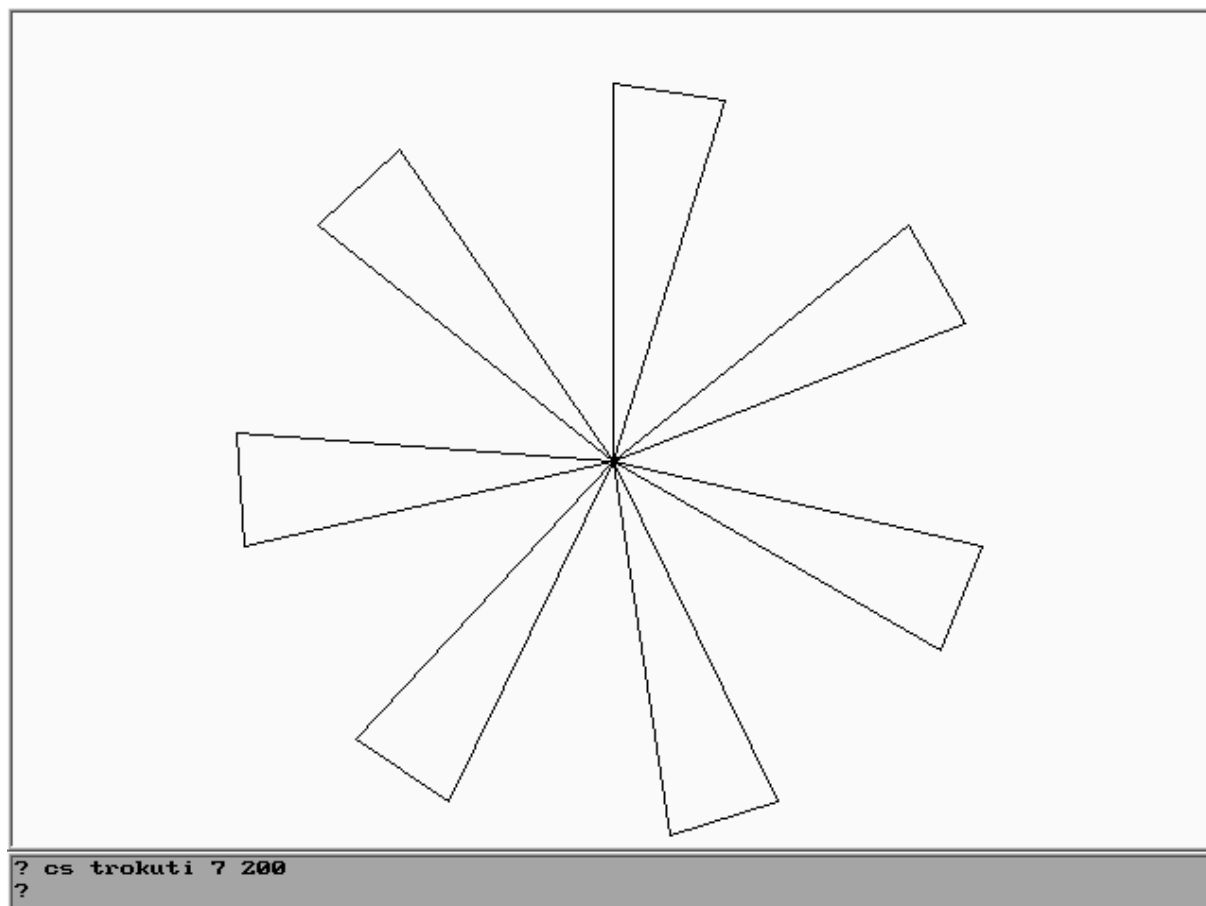
$:n$ je prirodan broj veći od jedan.

$:d$ je broj veći od nule i manji ili jednak od 200 (tako da lik ne prelazi rubove ekrana).



Pozicija lika na ekranu je bitna. Točka u kojoj se dodiruju trokuti treba biti na sredini ekrana.

Primjer (vidi sliku dolje): `TROKUTI 7 200`



Program snimite pod imenom **TROKUTI.LGO**

3. zadatak**ZID****80 bodova**

Zidar Pero zida zid. Šef mu je zadao širinu zida i broj redova (na slici desno imamo 3 reda). Pero ima više vrsta cigala — sve u obliku pravokutnika (na slici desno imamo 4 cigle širine 100, i bar 7 cigala širine 60). Srećom, visina svih cigli je jednaka (na slici desno 35).

Pero počinje zidati zid od donjeg lijevog ruba, slaže ciglu do cigle nadesno (kako pokazuju brojevi na ciglama) i to uzimajući prvo najveće cigle. Kada sazida potrebnu širinu (na slici desno širina zida je 270), započinje novi red iznad upravo sazidanog, ali s desna na lijevo. Sljedeći red zida ponovo slijeva nadesno itd.

Ako je zadnja cigla u redu bila prevelika (na slici desno iscrtkane cigle — broj 3 i 7), Pero ju presječe i sljedeći red nastavlja zidati sa tim odsječenim komadom cigle (osjenčani dio). Kada potroši sve cigle određene veličine, nastavlja zidati s najvećim ciglama od preostalih. Kada završi sa zidanjem, preostale cigle spremi na hrpu za idući zid. (Na slici desno nam je (ako je bilo 7 cigala širine 60) ostala jedna cigla širine 10 (odsječeni komad jedanaeste cigle – na slici desno zatamnjeni dio u gornjem desnom uglu).

Napišite funkciju `ZID :w :n :l :h` koja briše ekran i crta zid širine `:w` i vraća listu cigli koje su još preostale. Ako je cigala bilo točno za zid, funkcija vraća praznu listu. Ako je cigala bilo premalo, funkcija vraća broj 0.

Varijabla `:w` je pozitivan broj.

`:n` je prirodan broj koji označava broj redova cigala.

Lista `:l` se sastoji od bar jedne dvočlane liste. Prvi element dvočlane liste je pozitivan broj koji označava širinu cigle, a drugi je prirodan broj koji označava koliko takvih cigala imamo. Svake dvije dvočlane liste imaju različite prve brojeve (cigle su grupirane po širini).

Varijabla `:h` je pozitivan broj koji označava visinu cigle.

Moguć je i slučaj da je širina neke cigle i više nego dvostruko veća od širine zida.

U gore opisanom primjeru (`ZID 270 3 [[100 4] [60 7]] 35`), funkcija treba vratiti listu `[[10 1]]`.

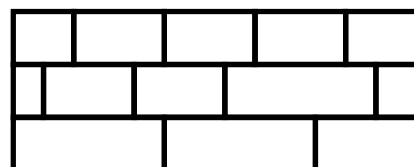
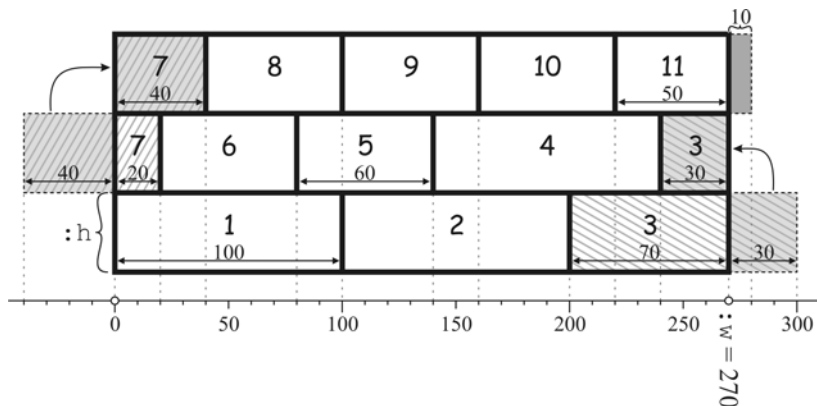
Preostale cigle treba grupirati tako da su podaci o ciglama jednake veličine zapisani u jednoj dvočlanoj listi (uključujući i eventualno preostali odsječeni komad cigle – drugi primjer ispod). Redoslijed grupa cigala nije bitan (u prvom primjeru ispod je prvo jedna manja cigla, pa dvije veće, dok su u drugom prvo dvije veće, pa četiri manje).

Pozicija lika na ekranu nije bitna.

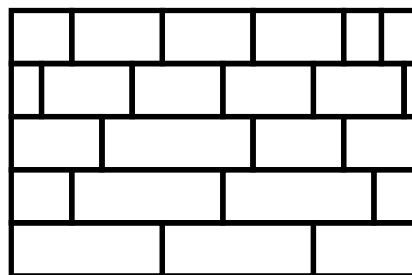
Primjeri:

```
? ZID 270 3 [[100 4] [60 9]] 35
Result: [[10 1] [60 2]]
```

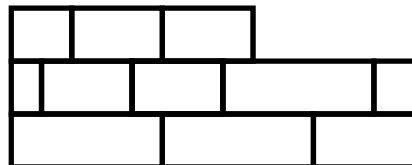
```
? show ZID 270 3 [[100 4] [60 9] [10 3]] 35
[60 2] [10 4]
```



? ZID 270 5 [[100 7] [60 10] [25 2]] 35
Result: []

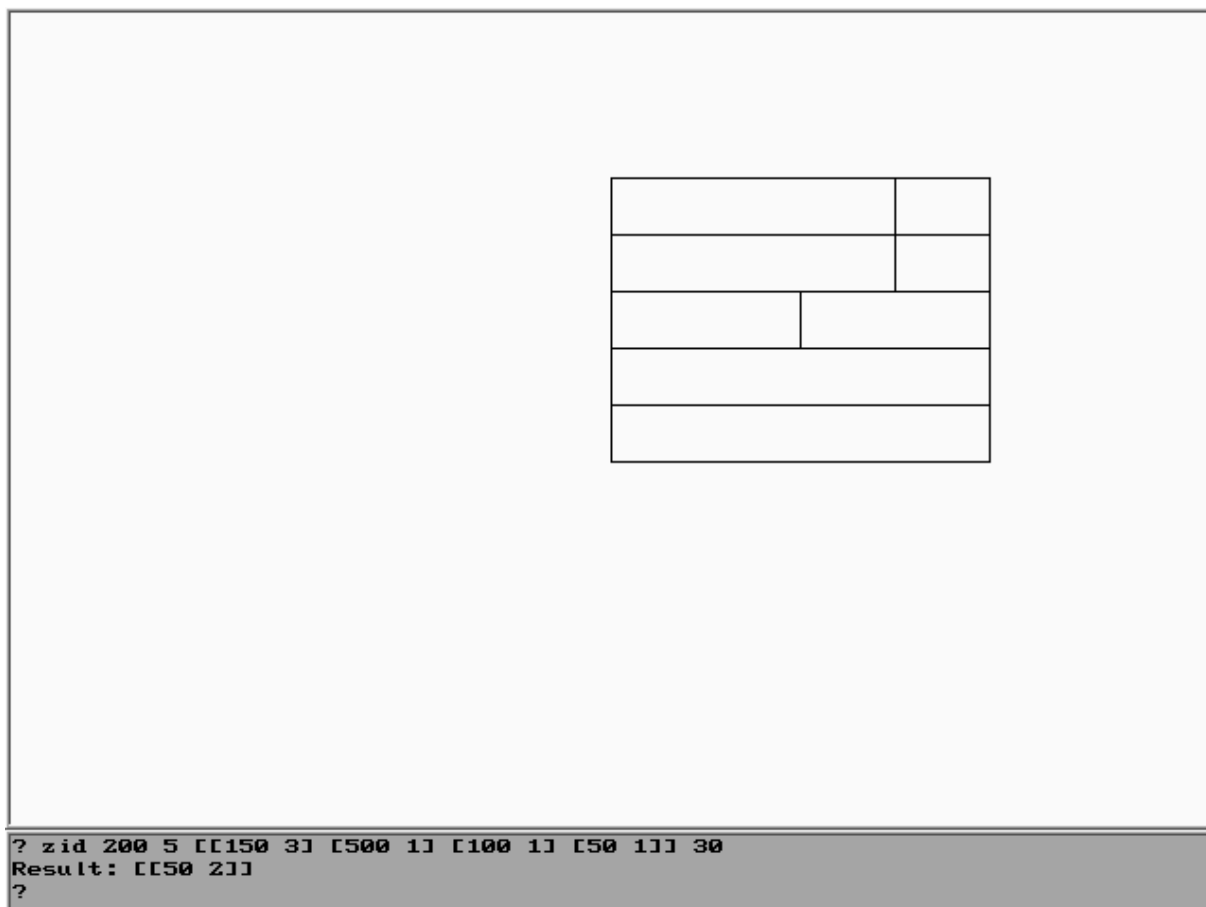


? pr ZID 270 3 [[100 4] [60 5]] 35
0



Primjer (vidi sliku dolje):

? ZID 200 5 [[150 3] [500 1] [100 1] [50 1]] 30
Result: [[50 2]]



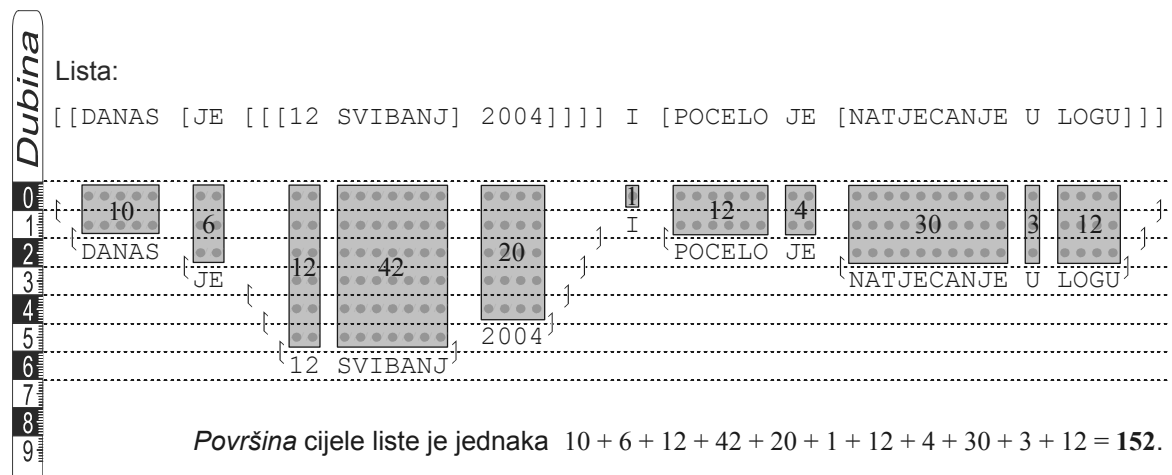
Program snimate pod imenom **ZID.LGO**

4. zadatak**POVRŠINA****50 bodova**

Definirajmo *dubinu* elementa liste kao broj listi unutar kojih se element nalazi. Npr. u listi `[1 2 [4]]` *dubina* brojeva 1 i 2 je 1, dok je *dubina* broja 4 jednaka 2.

Definirajmo *površinu* liste kao zbroj površina svih njenih elemenata. *Površina* prazne liste je jednaka 0. *Površina* riječi je jednaka *dubini* riječi pomnoženom s njenom duljinom. *Površina* broja je jednaka njegovoj *dubini* pomnoženom s brojem znamenaka.

Na slici ispod je skiciran jedan primjer:



Napišite funkciju `POVRSINA :l` koja računa *površinu* liste `:l`.

Lista `:l` se sastoji od prirodnih brojeva, riječi ili lista toga istog tipa.

Primjeri:

```
? POVRSINA []
Result: 0
```

```
? pr POVRSINA [1 2 3]
3
```

```
? show POVRSINA [DANAS JE [12 SVIBANJ 2004]]
33
```

```
? POVRSINA [[[]] []]
Result: 0
```

Program snimite pod imenom **POVRSINA.LGO**