

ATK Tähtitieteessä 2021 – Harjoitus 3.

1. Tee pääohjelma joka tekee sekä FOR ja WHILE rakenteella taulukon x joka sisältää luvut 0, 1, 2, 3, 4, ..., 100. Lisää IF-rakenne silmukoihin joka tulostaa jotain yllättävää kun $x=50$, ja muutoin pelkän $x:n$ arvon.
2. Tee tehtävä 1. aliohjelmalla ja funktiolla, jotka palauttavat niiden sisällä määriteltävän taulukon x takaisin komentotilaan. Määrittele niiden kutsussa lisäksi kaksi argumenttia:
 - Ensimmäinen argumentti N kertoo kuinka monta lukua haluamme, eli että tulos on luvut 0, 1, ..., N taulukkoon x .
 - Toinen argumentti määrittelee millä luvulla tulostetaan jotain yllättävää.
3. Tehdään edellisen harjoituksen tehtävän 4. heittoliikkeen laskusta aliohjelma, jonka kutsussa voidaan määritellä taulukossa mielivaltaisen monta x - ja y -suuntaista lähtönopeutta. Ohjelman on tarkoitus laskea heittoliikkeen rata jokaisella lähtönopeudella, ja lopuksi piirtää radat samaan ikkunaan.

Ohjelmaa siis pyritään kutsumaan esimerkiksi kahdella x - ja y -suuntaisella alkunopeudella seuraavasti:

```
vx = [2., 3.] & vy= [4., 5.]  
heittoliike, vx, vy
```

tai suoraan

```
heittoliike, [2., 3.], [4., 5.]
```

Heittoliikkeen yhtälöt olivat:

$$x = v_x t$$
$$y = v_y t - \frac{1}{2} g t^2.$$

Missä g on painovoiman kiihtyvyys (9.81 m/s^2), ja v_x, v_y ovat x - ja y -suuntaiset alkunopeudet. Muista määritellä ajan sisältävä taulukko t siten että se sisältää melko tiheävälistä juoksevia numeroita.

Laske heittoliikkeen radat FOR-silmukan sisällä, joka käy läpi jokaisen alkunopeusparin. Käytä silmukan sisällä IF-ELSE rakennetta, jossa ensimmäisellä alkuarvoparilla aukaiset ikkunan heittoliikkeen radan piirtoa varten ja loput radat piirretään tähän samaan ikkunaan. Katso IDL:n manuaalista `n_elements`-proseduuri, tämä voi olla hyödyllinen silmukkaa määriteltäessä.

Lisää ohjelman alkuun IF-rakenne joka ilmoittaa virheestä jos x- ja y-suuntaisia alkunopeuksia on eri määrä. Ehdon täyttyessä ohjelma myös lopettaa itsensä käyttäen STOP-proseduuria ja printtaa samalla x:n ja y:n elementtien lukumäärän.

Ohjelmaa voi tietysti parannella loputtomiin. Esim.:

-Lisää alkuun testi kutsutaanko ohjelmaa ilman yhtään parameteria: tämän voi tehdä *n_params()*- funktiolla (huom: tyhjä sulku). Jos näin on niin kirjoita lyhyt käyttöohje ja poistu ohjelmasta

-Lisää ohjelmaan keywordeja, joiden avulla voi esim. antaa kuvalle otsikon, säädellä plottaus-alueen kokoa tai mahdollistaa uusien kuvaajien piirtämisen entiseen kuvaan.

Käytä *if keyword_set(keywordin_nimi) then ...* -rakennetta

Huomaa että saman asian voi aina tehdä monilla eri tavoilla. Edellä hahmoteltiin aliohjelma, jolla voidaan laskea ja piirtää mielivaltaisen monta rataa samaan ikkunaan. Saman ongelman voi myös pilkkoa osiin: tehdä ensin aliohjelma joka laskee vain yhden radan. Ja kutsua tätä toisesta aliohjelmasta useilla eri alkuparametreilla. Usein tälläine ns. modulaarinen-ohjelmointi helpottaa esim. virheiden etsimistä ja ohjelman myöhempää muokkaamista

IDL:n käyttäminen kotikoneelta yliopiston **paju** koneen kautta

paju-kone näkyy yliopiston palomuurin läpi. **pajussa** ei ole IDL-ohjelmaa, mutta sen kautta voidaan ottaa yhteys työasemaluokan koneisiin:

```
ssh -X omatunnus@paju.oulu.fi
```

pajussa:

```
ssh -X omatunnus@y1124-pcNN
```

NN = työasemaluokan koneen numero

-X tarvitaan X-ikkunoinnin toimimiseksi