

Voit tällä kertaa valita yhden työn kolmesta, tee taas siis vain yksi työ! Tämä on harjoitustyö 3c:

Soluautomaatti (cellular automaton) on taulukko tiloja 1 tai 0 (solu on elossa tai ei), jotka muuttuvat jonkin yksinkertaisen säännön mukaan. Tässä tehtävässä sinun tulee kirjoittaa ohjelma, joka laskee erään soluautomaatin tiloja.

Jos automaatin tilat ovat taulukossa $A=(a_1, a_2, \dots, a_n)$, saadaan uudet tilat taulukkoon $B=(b_1, b_2, \dots, b_n)$ seuraavilla säännöillä. Olkoon i jokin taulukon indeksi väliltä $2 \dots n-1$:

1. Jos solun a_i naapurit a_{i-1} ja a_{i+1} ovat joko molemmat 1 tai molemmat 0, on solu b_i kuollut, eli nolla
2. Jos a_i :n naapureista a_{i-1} ja a_{i+1} vain toinen on 1, on solu b_i yksi eli elossa.
3. Solut b_1 ja b_n , eli reunoilla olevat solut, ovat 0.

Taulukossa B on sääntöjen soveltamisen jälkeen soluautomaatin uuden 'sukupolven' tilat.

C-kielellä automaatin tilat voidaan sijoittaa kokonaislukutaulukkoon, missä alkiot ovat nollia tai ykkösiä. Kirjoita funktio, joka laskee automaatin uudet tilat yo. sääntöjen mukaan. Funktion prototyyppi voisi olla

```
void cellauto(int a[], int n);
```

missä a on taulukko, jossa on automaatin tilat, ja n on solujen lukumäärä. Funktio laskee uudet tilat johonkin taulukkoon b , ja sen jälkeen kopioi taulukon b taulukkoon a . Pääohjelmassa, tee taulukko soluille, alusta se nolliksi yhtä solua lukuunottamatta. Saat itse valita kuinka monta solua ohjelmassasi käytät (n . 60 on sopiva määrä). Sen jälkeen laske funktiota `cellauto` käyttämällä automaatin tiloja n . 60 sukupolvea eteenpäin. Tulosta jokaisen sukupolven tilat ruudulle, esim. siten, että jos solu on 1, tulosta tähti (*) ja jos 0, tulosta välilyönti.

Alla on kaksi esimerkkikuva, jossa on laskettu 800 solun automaatin tiloja n. 380 sukupolvea. Oman ohjelmasi tulostuksen pitäisi näyttää samanlaista kolmiorakennetta, mutta paljon karkeampana.

