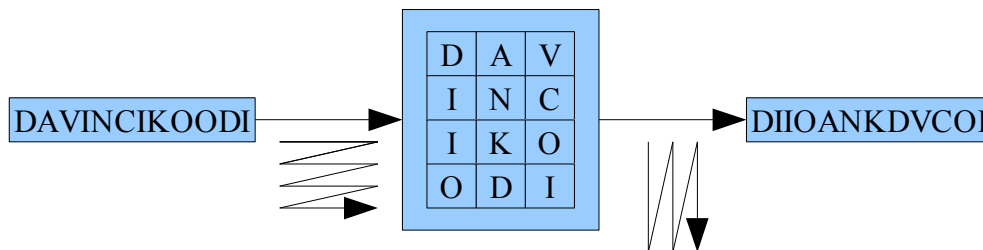


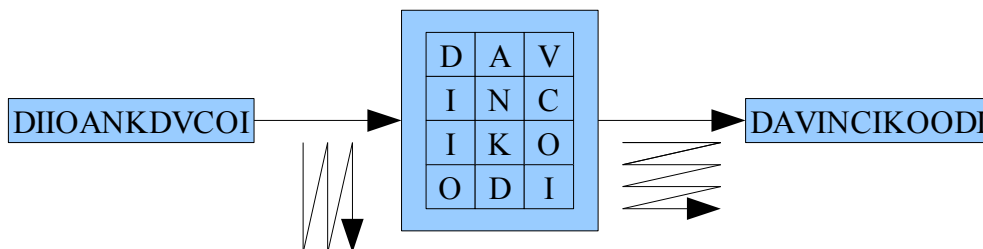
Voit tällä kertaa valita yhden työn kolmesta, tee taas siis vain yksi työ! Tämä on harjoitustyö 3b:

Salakirjoitusta. Rail-fence salakirjoituksessa salattava viesti kirjoitetaan vaakariveittäin kolmisarakkeiseen 2-ulotteiseen taulukkoon. Salattu viesti saadaan lukemalla täytetty taulukko pystyriveittäin. Salaus puretaan tekemällä sama päinvastoin: viesti kirjoitetaan taulukon pystyriveihin, jonka jälkeen taulukko luetaan vaakariveittäin.

Esimerkkinä, olkoon salattava viesti DAVINCIKOODI. Viesti kirjoitetaan aluksi taulukkoon vasemmalta oikealle, ja ylhäältä alas:



Salattu viesti saadaan, kun taulukko luetaan ylhäältä alas, ja vasemmalta oikealle: DIIOANKDVCOI. Salaus puretaan tekemällä samat askeleet, mutta toisin päin.



Kirjoita ohjelma, joka purkaa ja salaa viestejä. Salaus ja sen purku tulee tehdä omissa funktioissaan, joidenka prototyypit voisivat olla:

```
void cipher(char msg[], int len);
void decipher(char msg[], int len);
```

Ensimmäinen argumentti msg on purettava viesti, ja toinen len kertoo viestin pituuden. Viestin pituus oletetaan olevan jaollinen kolmella. Funktio sijoittaa salatun / puretun viestin takaisin merkkijonoon msg.

Pääohjelmassa lue käyttäjältä merkkijono, ja kysy puretaanko vai salataanko se. Esimerkiksi pura viesti TBLDIXHUEITYETRDZJ ja salaa viesti PAKENESINUTONHUOMATTU.

Vihje: Funktioissa tarvitset kaksiulotteisen merkkitaulukon (matriisin), jossa on kolme saraketta ja jokin riittävän suuri määrä rivejä, esim. 100. Salaamisessa täytä matriisi vaakariveittäin salattavalla viestillä, ja sitten kopioi matriisi pystyriveittäin takaisin alkuperäiseen merkkijonoon. Salauksen purkamisessa tee samat askeleet, mutta päinvastaisessa järjestyksessä: täytä matriisi pystyriveittäin, ja kopioi sitten vaakariveittäin.

Kun taulukko tulee käydä läpi *vaakariveittäin* täytyy sinun kirjoittaa silmukka, joka käy läpi *pystyrivi-indeksit*, $i = 0, 1, \dots, \text{len}/3 - 1$ (len on viestin pituus), ja sen sisään toinen silmukka, joka käy läpi yhden *vaakarivin* alkiot, eli indeksit $j = 0, 1, 2$. Kun taulukkoa tarkastellaan *pystyriveittäin*, on silmukoiden järjestys toisin päin: ulompi silmukka käy läpi *vaakarivit*.

Kun purkufunktio on valmis, voit halutessasi kirjoittaa myös salausfunktion. Saat salausfunktion todella helposti vain kopioimalla purkufunktion ja vaihtamalla silmukoiden järjestykset (kaikki sisemmät ulommiksi, ja kaikki ulommat sisemmiksi!).

Viestin pituuden laskemisessa voit käyttää `strlen` funktiota, joka laskee merkkijonon pituuden. Saat sen käyttöön lisäämällä ohjelmaasi `#include <string.h>` rivin.