

1. Kirjoita funktio, joka laskee matriisin ja vektorin tulon. Kun merkitään matriisia

$$A = (a_{ij})_{3 \times 3} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

ja vektoria

$$X = (x_i)_3 = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},$$

lasketaan matriisin ja vektorin tulo $AX = Y = (y_i)_3$ kaavalla $y_i = \sum_{j=1}^3 a_{ij}x_j$ eli

$$AX = Y = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \end{pmatrix}.$$

Funktion prototyyppi voisi olla muotoa

```
void mvmult(double a[][3], double x[], double y[]);
```

missä a on matriisi, x kerrottava vektori ja y vektori, johon tulos sijoitetaan.

2. Kirjoita funktio, joka laskee kuinka monta kertaa jokin merkki esiintyy merkkijonossa. Funktion prototyyppi voisi olla muotoa

```
int ccount(char str[], char c);
```

missä str on tutkittava merkkijono ja c merkki, jota etsitään. Funktio palauttaa kuinka monta kertaa luku löytyi. Vihje: kirjoita funktioon silmukka, jota suoritetaan kunnes merkkijonon loppumerkki '\0' saavutetaan.

3. Kirjoita ohjelma, joka suorittaa lotta-arvonnan. Avuksesi tarvitset funktion, joka antaa satunnaislukuja. Etsi C-kielen standardikirjastosta funktio rand(), ja tutki kuinka se toimii. Tee sitten ohjelma, joka laskee ja tulostaa seitsemän satunnaislukua väliltä 1-39.

Näin toteutettuna on kuitenkin mahdollista, että sama numero esiintyy lottorivissä useammin kuin kerran. Kokeile muuttaa ohjelmaasi siten, että kukin numero voi esiintyä rivissä vain kerran!

4. Lisätehtävä: Anagrammi testeri: Kirjoita funktio, joka testaa ovatko kaksi merkkijonoa toistensa anagrammeja. Funktion prototyyppi voisi olla `int is_anag(char stra[], char strb[]);`