

1. Kirjoita ohjelma, joka laskee summan  $S = 1+2+3+\dots+n$  käyttäen kolmea eri toistorakennetta. Luku  $n$  tulee lukea käyttäjältä.
2. Kirjoita ohjelma, joka lukee käyttäjältä kaksi lukua, ja tulostaa ne suuruusjärjestyksessä suuremmasta pienempään jos luvut ovat erisuuret, ja "Luvut ovat yhtäsuuret", jos luvut ovat yhtäsuuret.
3. Kirjoita funktio, joka etsii kokonaislukutaulukosta suurimman alkion. Funktion prototyyppi on `int maximum(int array[], int size);` missä `array` on tutkittava taulukko ja `size` on sen koko. Suurin alkio annetaan paluuarvona. Kirjoita pääohjelma, jossa käytät funktiota.
4. Kirjoita funktio, joka laskee kahden kompleksiluvun tulon. Kompleksiluvut tulee esittää kahden reaaliluvun tietueina, jonka kentät ovat kompleksiluvun reaali- ja imaginaariosat. Funktio saa tulon tekijät argumentteinaan, ja palauttaa tuloksen paluuarvonaan.
5. Kirjoita ohjelma, jossa teet  $10 \times 10$  reaaliluku matriisin `matrix` (kaksiulotteisen taulukon) ja asetat sen yksikkömatriisiksi. Yksikkömatriisissa alkiot  $a_{ii} = 1$  ja kaikki muut ovat nollia, ts.  $a_{ij} = 0$ , kun  $i$  on erisuuri kuin  $j$ . Käytä silmukoita.
6. Kirjoita funktio `swap`, joka vaihtaa kahden reaalilukumuuttujan arvot keskenään. Kirjoita pääohjelma, jossa käytät tätä funktiota.
7. Mitä seuraava ohjelma tulostaa, kun se ajetaan:

```
#include <stdio.h>
main()
{
    int i;
    double x = 0, y = 1, *p;
    p = &x;
    for (i = 0; i < 3; i++) {
        y *= 2;
        printf("%d %f\n", i / 2, *p);
        p = &y;
    }
}
```

8. Mitä seuraava ohjelma tulostaa, kun se ajetaan:

```
#include <stdio.h>
double hokkus(int i, double *x);
main()
{
    int i = 13, j;
    double x, y = 1, a[3] = {1, 2, 3};
    x = hokkus(i, &y);
    j = i / 5;
    printf("%d %f %f\n%f %d %d\n",
           i, x, y, a[1], i % 5, j);
}
double hokkus(int i, double *x)
{
    *x = i + 1;
    i = 3;
    return (2 * *x);
}
```

9. Kirjoita funktio, joka kopioi merkkijonon `a` merkkijonoon `b`. Funktion prototyyppi on `void copystr(char a[], char b[]);`