

Tartalomjegyzék

- 12. gyakorlat - Új típusok
 - ◆ 1.1
Feladatok
 - ◆ 1.2
Feladatok
megoldása
 - ◇ 1.2.1
fibonacci
 - ◇ 1.2.2
matrix szorzas 1
 - ◇ 1.2.3
matrix szorzas 2
 - ◇ 1.2.4
atlag

12. gyakorlat - Új típusok

Ezen a gyakorlaton az el?z? heti el?adás anyagát dolgozzuk fel.

Feladatok

Feladatok a CloudCoder-en megtalálhatóak. Ajánlott sorrend:

- fibonacci
- matrix_szorzas_1
- matrix_szorzas_2
- atlag

Feladatok megoldása

fibonacci

```
int fibonacci(int limit, int tomb[], int meret) {
    int i;
    tomb[0] = 1;
    tomb[1] = 1;
    for(i = 2; i < meret && tomb[i-1] + tomb[i-2] < limit; ++i) {
        tomb[i] = tomb[i-1] + tomb[i-2];
    }
    return i;
}
```

matrix_szorzas_1

```
void szoroz(matrix a, matrix b, matrix ret) {
    int i, j, k;
    for(i = 0; i < 2; ++i) {
        for(j = 0; j < 2; ++j) {
            ret[i][j] = 0;
            for(k = 0; k < 2; ++k) {
                ret[i][j] += a[i][k] * b[k][j];
            }
        }
    }
}
```

```
    }
  }
}
```

matrix_szorzas_2

```
struct MatrixTar szoroz(struct MatrixTar a, struct MatrixTar b) {
    int i, j, k;
    struct MatrixTar ret;
    for(i = 0; i < 2; ++i) {
        for(j = 0; j < 2; ++j) {
            ret.adat[i][j] = 0;
            for(k = 0; k < 2; ++k) {
                ret.adat[i][j] += a.adat[i][k] * b.adat[k][j];
            }
        }
    }
    return ret;
}
```

atlag

Amire itt figyelni kell, és sokaknak meglep?, hogy ha a mértani középnél "http://wiki.math.bme.hu1 / N"http://wiki.math.bme.hu-et írunk, akkor azt egész számok osztásaként értelmezi a C. Meg lehet mondani direkt hogy valós számokként kezelje ?ket, ha az egyiket valós számmá konvertáljuk, pl: "http://wiki.math.bme.hu1 / (float)N"http://wiki.math.bme.hu. De egyszer?bb az "http://wiki.math.bme.hu1"http://wiki.math.bme.hu helyett, ami egy egész szám, azt írni hogy "http://wiki.math.bme.hu1.0"http://wiki.math.bme.hu, ami már egy valós szám (a C szemszögéb?l).

```
float atlag(int tomb[], int N, enum atlag_fajta fajta) {
    float sum;
    int i;
    switch(fajta) {
        case SZAMTANI:
            sum = 0;
            for(i = 0; i < N; ++i) {
                sum += tomb[i];
            }
            return sum / N;
        case MERTANI:
            sum = 1;
            for(i = 0; i < N; ++i) {
                sum *= tomb[i];
            }
            return powf(sum, 1.0 / N);
        case NEGYZETES:
            sum = 0;
            for(i = 0; i < N; ++i) {
                sum += tomb[i] * tomb[i];
            }
            return sqrtf(sum / N);
        default:
            return -1;
    }
}
```