



A következő közelítő formulát használva írjon függvényt vagy metódust, amely egy valós szám négyzetgyökét adja vissza! Használja az $x_{k+1} = 1/2 * (x_k + a/x_k)$ sorozatot, amely a négyzetgyökéhez konvergál, ha $x_1 = 1$.

```
public static double squareRoot(double a) {
    double x = 1;
    for (int i = 0; i < 10; i++) {
        x = 0.5 * (x + a / x);
    }
    return x;
}
```

A fenti kódban a `for` ciklus fut 10 alkalommal, és minden futáskor frissíti az x értékét az $x_{k+1} = 1/2 * (x_k + a/x_k)$ képlet alapján. A ciklus futtatása 10 alkalommal elegendő a négyzetgyök kiszámításához, de ha nagyobb pontosságot szeretne, akkor növelnie kell a ciklus futtatásainak számát.



Miért elég 10 futás?

A fenti kódban a `for` ciklus fut 10 alkalommal a négyzetgyök kiszámításához, de ez csak egy példa. Az $x_{k+1} = 1/2 * (x_k + a/x_k)$ sorozat konvergenciája gyors, ami azt jelenti, hogy az x_k értékei hamar közelíteni fogják a valódi négyzetgyök értékét. Ha kevesebb futást használunk, akkor az x_k értékei valószínűleg nem lesznek elég pontosak, de ha több futást használunk, akkor az x_k értékei valószínűleg túl pontosak lesznek. 10 futás az egyensúlyt jelenti a pontosság és a teljesítmény között.

Azonban, el kell mondani, hogy ez a közelítő módszer nem minden esetben ad pontos eredményt, és a négyzetgyökök kiszámításánál jobb alternatívák is léteznek, pl. a `Math.sqrt()` függvény használata javascriptban vagy a `Math.sqrt()` metódus használata Java esetén.
