

NÉV (olvasható):

NEPTUN kód:

a) Készíts függvényt, amely egy másodfokú egyenlet három együtthatóját kapja, és referencia paraméterek segítségével visszatér az egyenlet gyökeire. A függvény egy logikai értéket adjon visszatérési értéként, amely hamis, ha az egyenletnek nincs valós gyöke, és igaz, ha van. (4p)

b) Írj main függvényt, amelyben kipróbáld az a) feladatban létrehozott függvényt. (2p)

c) Írj C++ programot, amely egy, a felhasználó által megadott méretű, valós számok tárolására alkalmas dinamikus tömböt véletlen értékekkel tölt fel, kiírja a tömb tartalmát, majd felszabadítja a tömböt. A dinamikus memória kezeléséhez a C++ operátorait használd! (4p)

max. 10p, megoldási idő: 14 perc

```
#include <stdio.h>
#include <math.h>

bool masodfoku(double A, double B, double C, double &x1, double &x2) {
    // 1p: referencia helyes megadása
    // 1p: jó visszaadott típus (bool vagy egész)
    double d = B*B - 4 * A*C;
    if (d < 0)
        return false; // 1p: helyes visszaadott érték (false vagy 0)
    x1 = (-B + sqrt(d)) / (2 * A); // 1p: referencia helyes használata
    x2 = (-B - sqrt(d)) / (2 * A);
    return true;
}

int main() {
    double x1, x2; // 1p: referencia értéke és helyes használata: nem jár, ha csinál felesleges
    // referencia változót a main-ben.
    if (masodfoku(1, -5, 6, x1, x2)) // 1p, ha nem cifrázza a logikai érték vizsgálatát, azaz
        // nincs ==true vagy ==1
        printf("%g, %g\n", x1, x2);
    return 0;
}

#include <stdio.h>
#include <stdlib.h>

int main() {
    unsigned n;
    scanf("%d", &n);
    double *t = new double[n]; // 2p: helyes tömbfoglalás new-val. Részpontoszám nincs.
    for (unsigned i = 0; i < n; i++)
        t[i] = rand(); // 1p: rand függvényt ismeri és jól használja
    for (unsigned i = 0; i < n; i++)
        printf("%g\n", t[i]);
    delete[] t; // 1p: ha a delete[] operátort használja, jól
    return 0;
}
```