

Programozás alapja 2.	1. ellenőrző dolgozat.	2015.03.02.	Kurz/Terem: G1/	Elért pontszám:
Név:		Neptun:		

Az első két feladatnál a helyes válasz 1 pont, a hibás -1pont! Az esetleges negatív eredmény is összeadódik a többi feladatra kapott pontokkal. A teljes dolgozat eredménye azonban nem lehet negatív.

1. Jelölje, hogy mely kijelentés(ek) igaz(ak) a C++ referencia típusára ill. a referencia típusú változóra! (2p)

- Teljesen azonos a pointerrel.
 Szerepelhet balértékként.
- Alternatív név.
 Csak lokális változó érhető el vele.
- Nem kell inicializálni.
 Csak paraméterátadásra használható.

2. Jelölje, hogy mely kijelentés(ek) igaz(ak) a C++ enum típusára ill. az enum típusú változóra! (1p)

- Nem kaphat egész értéket.
 Teljesen azonos a C enum típusával.
- Nem szerepelhet balértékként
 Csak bool típusból inicializálható.

3. **Készítsen** C++ nyelven függvényt, ami egész számok tárolására alkalmas dinamikus adatterületet foglal, és feltölti a lefoglalt terület minden elemét egy paraméterként átadott értékkel! A függvény paraméterként vegye át a terület méretét is! A függvény interfészét úgy tervezze meg, hogy azt fel tudja használni a következő feladatban! A dinamikus memória kezeléséhez a C++ nyelvi eszközeit használja! (3p)

4. Az előző feladatban elkészített függvényt **felhasználva** írjon **teljes** C++ programot, ami beolvas két egész számot a standard inputról, majd lefoglal az első számnak, mint méretnek megfelelő nagyságú, egész számok tárolására alkalmas dinamikus adatterületet, amelynek minden elemét feltölti a másodikként beolvasott értékkel! A főprogram ezután írja ki a lefoglalt adatterület első elemét, majd szabadítsa fel a lefoglalt területet! A beolvasáshoz, kiíráshoz a C++ adatfolyamait használja! Feltételezheti, hogy az inputon helyes adatok érkeznek és rendelkezésre áll elegendő dinamikus memória. (4p)

3. Feladat megoldása:

```
int *foglal(unsigned int meret, int ertekek) {
    int *p = new int[meret];
    for (unsigned int i = 0; i < meret; ++i)
        p[i] = ertekek;
    return p;
}
```

4. Feladat megoldása:

```
#include <iostream>
#include "foglal.h" // tfh. ebben van a 3. feadat függvényének prototípusa

int main() {
    unsigned int meret;
    int kezdok;
    std::cin >> meret >> kezdok;
    int *adat = foglal(meret, kezdok);
    std::cout << adat[0] << std::endl;
    delete[] adat;
    return 0;
}
```

Programozás alapja 2.	1. ellenőrző dolgozat.	2015.03.02.	Kurz/Terem: G1/	Elért pontszám:
Név:			Neptun:	

Az első két feladatnál a helyes válasz 1 pont, a hibás -1pont! Az esetleges negatív eredmény is összeadódik a többi feladatra kapott pontokkal. A teljes dolgozat eredménye azonban nem lehet negatív.

1. Jelölje, hogy mely kijelentés(ek) igaz(ak) a C++ referencia típusára ill. a referencia típusú változóra! (1p)

- Szerepelhet balértékként. Teljesen azonos a pointerrel.
 Csak lokális változó lehet. Nem lehet konstans.
 Nem kell inicializálni. Csak paraméterátadásra használható.

2. Jelölje, hogy mely kijelentés(ek) igaz(ak) az alábbi C++ kódrészletre! (2p)

```
void fv(int a = 1, const int& b) { b = a; }
```

- teljesen helyes. hibás, mert default paraméter után már csak default paraméter következhet.
 hibás, mert b nem változtatható meg. hibás, mert minden referenciát inicializálni kell.

3. Készítsen C++ nyelven függvényt, ami valós számok tárolására alkalmas dinamikus adatterületet foglal, és feltölti a lefoglalt terület minden elemét egy paraméterként átadott értékkel! A függvény paraméterként vegye át a terület méretét is! A függvény interfészét úgy tervezze meg, hogy azt fel tudja használni a következő feladatban! A dinamikus memória kezeléséhez a C++ nyelvi eszközeit használja! (3p)

4. Az előző feladatban elkészített függvényt felhasználva írjon C++ programot, ami beolvas két értéket a standard inputról, majd lefoglal az első számnak, mint méretnek megfelelő nagyságú, valós számok tárolására alkalmas dinamikus adatterületet, amelynek minden elemét feltölti a másodikként beolvasott valós értékkel! A főprogram ezután írja ki a lefoglalt adatterület utolsó elemét, majd szabadítsa fel a lefoglalt területet! A beolvasáshoz, kiíráshoz a C++ adatfolyamait használja! Feltételezheti, hogy az inputon helyes adatok érkeznek és rendelkezésre áll elegendő dinamikus memória. (4p)

3. Feladat megoldása:

```
double *foglal(unsigned int meret, double ertekek) {
    double *p = new double[meret];
    for (unsigned int i = 0; i < meret; ++i)
        p[i] = ertekek;
    return p;
}
```

4. Feladat megoldása:

```
#include <iostream>
#include "foglal.h" // tfh. ebben van a 3. feadat függvényének prototípusa

int main() {
    unsigned int meret;
    double kezdok;
    std::cin >> meret >> kezdok;
    double *adat = foglal(meret, kezdok);
    std::cout << adat[meret-1] << std::endl;
    delete[] adat;
    return 0;
}
```