

## 1. Zárthelyi dolgozat (A csoport)

Név: ..... Neptun-kód: .....

1. Írjunk C++ függvényt, amely paraméterül kap egy C string-et, és visszaadja, hogy palindrom-e (azaz balról jobbra olvasva ugyanaz-e, mint jobbról balra olvasva). (8 pont)
2. Definiáljuk az  $a_n$  számsorozatot a következőképp: legyen  $a_0 = 2$ ,  $a_1 = -1$ , továbbá  $n > 1$  esetén legyen  $a_n = (-1)^n a_{n-1} + |a_{n-2}|$ . Írjunk C++ függvényt, amelynek visszatérési értéke a sorozat  $n$ -edik tagja, az  $n$ -et paraméterként lehessen megadni. (10 pont)
3. Írjunk C++ függvényt, amely visszaadja egy paraméterként kapott, `int` értékeket tartalmazó `list`-ben szereplő második legnagyobb értéket, ha pedig nincs ilyen (tehát csak egyféle érték szerepel a listában, vagy az 0 elemű), akkor pedig 0-t. (Az `#include<list>` sort nem szükséges leírni a megoldásban.) (10 pont)
4. A következő C++ kódban 5 hiba van, amik miatt nem fordul le. Keressük meg ezeket. (10 pont)

```
#include<iostream>

class C{
private:
    int a;
    C() :a(0) {}
public:
    C(int _a) :a(_a) {}
    int getVal() { return a; }

    C& operator+=(const C& other)const { a += other.a; }
    bool operator==(const C& other)const { return a == other.getVal(); }
};

int main() {
    C c1;
    C c2(2);
    c2 += c1;
    std::cout << c2.a;
}
```

5. Írjunk egy `Matrix` osztályt, amely egy  $2 \times 2$ -es valós mátrixot képes reprezentálni. Adjuk meg az adattagokat (mind legyen `private`), és írjunk meg a következő `public` konstruktorokat és tagfüggvényeket (legyen mind `inline`): (12 pont)
  - Legyen egy default konstruktor, amely minden tagváltozó értékét 0-ra állítja.
  - Legyen egy konstruktor, amelynek 4 paramétere van, és az ezekben tárolt értékekre beállítja a tagváltozók értékeit.
  - Legyenek olyan tagfüggvények, amelyekkel beállíthatók és lekérdezhetők a mátrix elemei.
  - Írjunk egy `+` operátort, amely két mátrix összegét adja vissza.
  - Írjunk egy `*` operátort, amely a mátrix egy skalárszorosát adja vissza.
  - Írjunk `==` és `!=` összehasonlító operátorokat.

Írjunk az osztályon kívül egy `inline` függvényt, ami hozzáfér a `Matrix` osztály tagváltozóihoz, és visszaadja egy paraméterként kapott `Matrix` transzponáltját.

## 1. Zárthelyi dolgozat (B csoport)

Név: ..... Neptun-kód: .....

1. Írjunk C++ függvényt, amely paraméterül kap egy C string-et, és visszaadja, hogy palindrom-e (azaz balról jobbra olvasva ugyanaz-e, mint jobbról balra olvasva). (8 pont)
2. Definiáljuk az  $a_n$  számsorozatot a következőképp: legyen  $a_0 = 0$ ,  $a_1 = -2$ , továbbá  $n > 1$  esetén legyen  $a_n = (-1)^n a_{n-1} + 2 \cdot |a_{n-2}|$ . Írjunk C++ függvényt, amelynek visszatérési értéke a sorozat  $n$ -edik tagja, az  $n$ -et paraméterként lehessen megadni. (10 pont)
3. Írjunk C++ függvényt, amely visszaadja egy paraméterként kapott, `int` értékeket tartalmazó `list`-ben szereplő második legnagyobb értéket, ha pedig nincs ilyen (tehát csak egyféle érték szerepel a listában, vagy az 0 elemű), akkor pedig 0-t. (Az `#include<list>` sort nem szükséges leírni a megoldásban.) (10 pont)
4. A következő C++ kódban 5 hiba van, amik miatt nem fordul le. Keressük meg ezeket. (10 pont)

```
#include<iostream>

class C{
private:
    int a;
    C(int _a) :a(_a) {}
public:
    C() :a(0) {}
    int getVal() { return a; }

    C& operator+=(const C& other) { a += other.a; }
    bool operator==(const C& other) const { return a == other.getVal(); }
};

int main() {
    C c1;
    C c2(2);
    c2 -= c1;
    std::cout << c2.a;
}
```

5. Írjunk egy `Matrix` osztályt, amely egy  $2 \times 2$ -es valós mátrixot képes reprezentálni. Adjuk meg az adattagokat (mind legyen `private`), és írjunk meg a következő `public` konstruktorokat és tagfüggvényeket (legyen mind `inline`): (12 pont)
  - Legyen egy default konstruktor, amely minden tagváltozó értékét 0-ra állítja.
  - Legyen egy konstruktor, amelynek 4 paramétere van, és az ezekben tárolt értékekre beállítja a tagváltozók értékeit.
  - Legyenek olyan tagfüggvények, amelyekkel beállíthatók és lekérdezhetők a mátrix elemei.
  - Írjunk egy `+` operátort, amely két mátrix összegét adja vissza.
  - Írjunk egy `*` operátort, amely a mátrix egy skalárszorosát adja vissza.
  - Írjunk `==` és `!=` összehasonlító operátorokat.

Írjunk az osztályon kívül egy `inline` függvényt, ami hozzáfér a `Matrix` osztály tagváltozóihoz, és visszaadja egy paraméterként kapott `Matrix` transzponáltját.