

NÉV (olvasható):

NEPTUN kód:

max. 10p, megoldási idő: 15 perc

1. Adott az alábbi Window osztály, amely egy téglalap alakú területet reprezentál a képernyőn. Származtass belőle jelölőnégyzetet reprezentáló Checkbox osztályt, amely pluszként a jelölőnégyzet feliratát és bekapcsolt/kikapcsolt állapotát (logikai érték) tárolja! Készíts konstruktort, amely a jelölőnégyzet minden adatát inicializálja! Definiáld felül a print függvényt úgy, hogy az a jelölőnégyzet minden adatát írja ki! Írj kódrészletet, amelyben létrehozol egy jelölőnégyzetet, és kiírod az adatait! A Window osztályt nem módosíthatod! (4p)

```
class Window {
    int x, y, w, h;
public:
    Window(int x, int y, int width, int height)
        : x{ x }, y{ y }, w{ width }, h{ height } {}
    void print()const {
        std::cout << "x=" << x << ", y=" << y << ", w=" << w << ", h=" << h;
    }
};

class Checkbox :public Window { // 1p
    std::string text;
    bool checked;
public:
    Checkbox(int x, int y, int width, int height, std::string text, bool checked)
        : Window(x, y, width, height), text{ text }, checked{ checked } {} // 1p
    void print()const {
        Window::print(); // 1p
        std::cout << ", text: " << text << ", checked: " << (checked ? "true" : "false");
    }
};

Checkbox g(1, 2, 3, 4, "OK", true); // 1p
g.print();
```

2. Készíts generikus függvényt, amely egy tömböt kap paraméterként, és visszaadja a tömb legkisebb elemének indexét! Készíts specializációt C sztringekre mutató pointerok tömbje esetére! Írj kódrészletet, amelyben kipróbálsd a függvényt double tömbre és C sztringpointer tömbre is! (A kipróbálásnál felhasználhatod a következő sztringpointer tömböt.) (6p)

```
const char* sptomb[4] = { "Dénes", "Aladár", "Cecil", "Béla" };

template<typename T> // 1p
int minkeres(const T t[], int n) { // 1p
    int minindex = 0;
    for (int i = 1; i < n; i++) // alg: 1p
        if (t[i] < t[minindex])
            minindex = i;
    return minindex;
}

template<>
int minkeres(const char* const t[], int n) { // specializáció 1p
    int minindex = 0;
    for (int i = 1; i < n; i++)
        if (strcmp(t[i], t[minindex]) < 0) // strcmp 1p
            minindex = i;
    return minindex;
}
```

```
double dt[6] = { 10, -8, 0, 99, -11, 3 };
const char* sptomb[4] = { "Dénes", "Aladár", "Cecil", "Béla" };
std::cout << dt[minkeres(dt, 6)] << std::endl; // ha mindkét példa jó: 1p
std::cout << sptomb[minkeres(sptomb, 4)] << std::endl;
```