Sapientia Erdélyi Magyar Tudományegyetem
Marosvásárhelyi Kar

Felvételi, 2026. szeptember
Szoftverfejlesztés mesterképzés

I. Adatbázisok

Adott a következő relációs modellben leírt adatbázis, mely készruhát áruló cég részleges adatbázisa.

Uzletek (UzletID, UNev, ManSzemSzam);
RuhaCsop (CsopID, CsNev); // nadrágok, szoknyák, télikabát, stb.
Ruhak (RuhaID, RNev, MertEgys, CsopID);
Raktar (UzletID, RuhaID, MennyiségRaktaron, EladasiAr)
Szallitok (SzallID, SzNev, Helyseg, UtcaSzam);
Szallit (SzallID, RuhaID, Datum, SzallitasiAr);

Oldd meg SQL parancs segítségével a következő feladatokat:

a) Add meg az Alexis nevű üzletben található ruhák nevét! **(1 pont)**

b) Add meg azon szállítók nevét, akik csak nadrágot szállítanak! **(1 pont)**

c) Add meg a legolcsóbb esőkabátot, melyből van raktáron! (UNev, RNev, MennyiségRaktaron formájában). **(1 pont)**

d) Add meg a Stefanis nevű üzletben található ruhák értékét csoportokon belül a következő formában: CsNev, Ertek. **(1 pont)**

e) Add meg azon szállítók nevét, kik pontosan 3 féle ruha csoportot szállítanak! **(1 pont)**

f) Add meg minden üzlet esetén annak a ruha csoportnak a nevét, melyből a legnagyobb értékben található ruha az illető üzletben! (UNev, CsNev, Ertek formájában) **(2 pont)**

g) Módosítsd meg minden télikabát eladási árát, tedd egyenlővé a legolcsóbb Alexis üzletben lévő télikabát árával. **(1 pont)**

h) Törölj minden olyan szállítót, aki több mint 2 éve nem szállított. **(1 pont)**

Hivatalból 1 pont.

II. Objektumelvű programozás és szoftvertervezés

1. Mi a különbség a példányszintű és az osztályszintű (statikus) adattagok között? Adjon példát egy olyan osztályra (Java, C++), amely mindkét típusú adattagot tartalmaz és magyarázza a működést. **(1 pont)**

2. Adjon példát **egy-a-sokhoz** (*one-to-many*) tartalmazási kapcsolatra UML diagrammal, illetve ennek megfelelő kódrészlettel. A diagramon csak az osztályok nevét kell feltüntetni, a kód legyen egyszerű, csak a legszükségesebbeket tartalmazza (Getter és setter metódusokat ne írjon!). Magyarázza a kapcsolat implementálását. **(2 pont)**

3. Készítsen egy *egyszeresen láncolt listát.* Használjon 2 osztályt, egyet a listaelemnek és egyet a listának. A listaelem adatként tároljon egy egész számot. A lista biztosítsa a következő műveleteket:
    - Konstruktor - üres lista inicializálása
    - Új elem beszúrása a listafej elé
    - Lista elemeinek kiíratása a standard kimenetre
    - Adott egész szám keresése a listában. **(4 pont)**

4. Mit jelent az **a nyílt-zárt elv** (*Open-Closed Principle*)? Adjon egy példát kóddal, amely **betartja** ezt az elvet, és **magyarázza meg**, hogy miért! **(2 pont)**

Hivatalból 1 pont.

III. Programozás és programozási nyelvek

Koktélrendezés

Írd meg az úgynevezett *koktélrendezési algoritmust* az x[0..n-1] tömb elemeinek növekvő sorrendbe rendezésére. A koktélrendezés a buborékrendezés továbbfejlesztett változata: felváltva halad végig az aktuálisan rendezetlen szakaszon *balról jobbra*, majd *jobbról balra*.

1. feladat – Az algoritmus megvalósítása, mint C/C++ függvény (max. 8 pont)

- **Max. 6 pont:** minden balról jobbra és minden jobbról balra történő „buborékoztatási” menet a teljes tömbön végighalad.
- **Max. 7 pont:** az algoritmus kihasználja, hogy a balról jobbra történő menet után az aktuális szakasz legnagyobb, a jobbról balra történő menet után pedig annak legkisebb eleme a helyére kerül; ennek megfelelően a rendezetlen szakasz határai minden menet után 1-gyel szűkülnek.
- **Max. 8 pont:** az algoritmus minden menetben megjegyzi az *utolsó csere helyét*, és ennek megfelelően módosítja a következő menet határait.

Az algoritmus álljon le, amint algoritmikai garancia van arra, hogy a sorozat már rendezett.

2. feladat – Időbonyolultság (max. 1 pont)

Add meg a koktélrendezési algoritmus időbonyolultságát *legjobb és legrosszabb esetben*. Indokold meg válaszod!

1 pont hivatalból.

Megjegyzés: nem használható segédtömb.

Lehetséges C fejléc:

```
void koktelrendezes(int *x, int n) { ... }
```