

A/1 (1 pont)

Készíts *Reversal* nevű sabloneljárást egy sorozat megfordítására. A sablon paraméterei egy indefinit tömb típus, valamint annak index- és elemtípusa. A sablonból példányosuló eljárás kapjon egy tömböt paraméterként, és ennek a tömbnek az elemeit kell átrendezni úgy, hogy az elsőt az utolsóval, a másodikat az utolsó előttivel stb. fel kell cserélni. A sablonparaméterek a lehető legáltalánosabbak legyenek. Példányosítsd a sablont és teszteld a megoldást egy főprogramban. Segítség: a sablon megvalósításához használható a diszkrét típusosztály *Succ* és *Pred* attribútuma.

A/2 (1 pont)

Alakítsd át az előző feladatban szereplő sablont úgy, hogy át kelljen adni a sablonnak egy *Swap* eljárást, amely két elemtípusú változó tartalmát felcseréli. Ezzel a felcserélő eljárással kell megvalósítani a *Reversalt*.

A tömb elemtípusaként átadott sablonparaméter akár korlátozott típus is lehessen. A főprogramot módosítsd az új sablonnak megfelelően.

A/3 (2 pont)

Készíts *Bag* (azaz Multiset) adatszerkezetet megvalósító sabloncsomagot. A *Bag* annyiban különbözik a halmaztól, hogy a benne tárolt elemekről multiplicitást is nyilván tartunk. Egy elem törlése csökkenti a hozzá tartozó multiplicitást. Például: ha *x* benne van a bagben ötször, akkor ismét beletéve a multiplicitása hatra változik, míg törölve az *x*-et a multiplicitás négyre csökken. Reprezentáld a *Bag* absztrakt adattípust egy rekorddal, amelynek diszkriminánsa a Bag kapacitását adja meg. A rekordban legyen egy tömb, ami (érték, multiplicitás) párokat tartalmaz. Az értékek típusa a sablon paramétere legyen. Valósíts meg *Add* és *Remove* eljárásokat, valamint *Multiplicity* függvényt az adatszerkezetre.

Készíts hozzá belső iterátort sabloneljárás formájában. A *For_Each* iterátort egy olyan eljárással lehessen paraméterezni, amely egy értéket és egy multiplicitást vár paraméterként, és amit a *For_Each* alkalmaz egy bag minden nem nulla multiplicitású elemére (és annak multiplicitására).

B/1 (1 pont)

Írj egy olyan sablon eljárást, amely eldönti, hogy egy sorozatban előfordul-e többször ugyanaz az érték. Az eljárás kap bemenő paraméterként egy tömböt, kimenő paraméterként pedig egy logikai értéket és két indexpozíciót ad vissza. Akkor adjon vissza igazat, ha két különböző indexpozíción ugyanaz az érték van a tömbben, és ekkor adja vissza a két indexpozíciót is. Ha minden érték csak egyszer szerepel a tömbben, akkor hamisat adjon vissza az eljárás a logikai típusú paraméterében. A tömb típus, az index- és az elemtípusa legyen (a lehető legáltalánosabbra megadott) sablonparaméter.

B/2 (1 pont)

Alakítsd át az előző feladatban szereplő sablont úgy, hogy át kelljen adni a sablonnak egy bináris relációt (a neve legyen alapértelmezés szerint "="). Az eljárás azt döntse el, hogy két különböző indexű tömbelem között fennáll-e a megadott reláció. (Ne feltételezd azt, hogy a reláció szimmetrikus!)

A tömb elemtípusaként átadott sablonparaméter akár korlátozott típus is lehessen. A főprogramot módosítsd az új sablonnak megfelelően.

B/3 (2 pont)

Valósítsd meg a Polinom absztrakt adattípust egy sablon csomaggal. A sablon paramétere a polinom együtthatóinak típusa, valamint a szükséges aritmetikai műveletek erre a típusra. A polinomokhoz (az együtthatók lekérdezése és beállítása mellett) definiálj összeadás, kivonás, szorzás és (egy adott pontban történő) kiértékelés műveletet. A Polinom reprezentációja egy rekord legyen, melynek a diszkriminánsa a polinom maximális fokszámát határozza meg.

Az együtthatókat egy tömbben tárold. A megoldásod teszteléséhez példányosítsd a sablont *Float* együtthatójú polinomokra.