

## # B csoport

Az egyes részfeladatok megoldását rendre az `a1`, `a2` és `a3` könyvtárakban kell elkészíteni. Beadás előtt törölni kell a futtatható állományokat, valamint a fordítás során generált `.ali` és `.o` fájlokat. A forrásokat tartalmazó könyvtárakat egy `.zip` állományba kell összecsomagolni, és ezt a zipet kell feltölteni.

Törekedj az elegáns programozásra! Használd ki az Ada tanult lehetőségeit! Fordíts figyelmet az azonosítók megválasztására és a helyes tördelésre. Ezek az apróságok akár egy egész jegyet is jelenthetnek az értékelésben!

### ## 1. feladat (kötelező a kettős érdemjegyhez)

Készíts sablonfüggvényt `Has\_Repetition` névvel, melyet egy indefinit tömb típussal (valamint annak index- és elemtípusával) lehet paraméterezni. (Nevezzük a tömb típust `Vector`-nak.) A sablonfüggvény példányosításával olyan függvényhez jutunk, amely egy `Vector`-t kap paraméterül, és logikai értéket ad vissza: pontosan akkor kapunk igazat, ha a paraméterként kapott `Vector`-ban van olyan elem, amely rákövetkezője egyenlő vele (azaz  $v(i) = v(i+1)$ ).

Készíts főprogramot, amelyben példányosítod a fenti sablonfüggvényt a `String` tömb típussal, és így kapod a `Has\_Double\_Letters` függvényt. Teszteld kimerítően a függvényt. Gondold végig, milyen szélsőséges esetek léteznek.

### ## 2. feladat

Az előző feladat mintájára készítsd el a `Most\_Frequent` sablonfüggvényt. Ez egy paraméterként kapott tömbből megmondja, hogy melyik elem fordult elő legtöbbször a tömbben. Például az "Elmegyek ebédelni!" stringben az 'e' betű a leggyakoribb.

Válasszuk a lehető legegyszerűbb megoldást. Implementáljuk a sablonfüggvényt úgy, hogy a `Multiplicity` függvény maximumhelyét határozzuk meg a tömbben, ahol is a `Multiplicity(Pattern)` megszámolja, hogy a `Pattern` érték hányszor fordul elő a tömbben. A maximumhelyhez tartozó tömbértéket kell visszaadni.

Készíts főprogramot, amelyben példányosítod a fenti sablonfüggvényt, teszteld kimerítően a függvényt.

### ## 3. feladat

Tegyük hatékonyabbá az előző megoldást a következőképpen. A `Most\_Frequent` sablonfüggvényben definiáljunk egy `Multiplicity` tömböt, amelynek indexelése megegyezik a paraméterként kapott tömb indexelésével, és természetes számokat tartalmaz. Töltsük ki a `Multiplicity` tömböt az elejétől a végéig haladva úgy, hogy az `i` indexű elem megmondja, hogy az eredeti tömb `i` indexű eleme hányszor fordult elő a tömb elejétől az `i` indexig bezárólag. (Induljunk el visszafelé az eredeti tömbben, és keressük az `i` indexű elemünk előző előfordulását. Ha nem találunk előző előfordulást, az `i` indexű elem multiplicitása 1 lesz. Ha találunk előző előfordulást, akkor az `i` indexű elem multiplicitása a talált előforduláshoz tartozó multiplicitásnál eggyel nagyobb lesz.) Az így előállított `Multiplicity` tömb maximumhelyét kell már csak megkeresnünk, és az ehhez tartozó tömbértéket visszaadni.

Készíts főprogramot, amelyben példányosítod a fenti sablonfüggvényt, teszteld kimerítően a függvényt.