

# Programozás Alapjai 6. próba feladat

## 1. feladatsor

Szoftverfejlesztés Tanszék

2023, Ősz

**Feladat** Töltsd le a bíróról a `minta.zip` állományt, majd tömörítsd ki! A `feladat.c` fájlban megtalálod a feladatok megoldás-kezdeményeit. Bővítsd ezt az alább olvasható feladatok alapján! Lehetőség szerint ellenőrizd megoldásod, majd töltsd fel a `feladat.c` fájlt a bíróra!

**Kiértékelés** A bíró lefordítja a programot, majd lefuttatja azt a feladat pontszámának megfelelő számú tesztessel. Egy tesztet egy bemenet-kimenet pár, amely a megfelelő feladathoz készült. A tesztet akkor helyes, ha az adott bemenethez tartozó kimenet **minden egyes karaktere** megegyezik az előre eltárolt referencia kimenettel. *További feltételek: a program futása nem tarthat tovább 5 másodpercnél, egyszerre nem fogyaszthat többet 16 MiB memóriánál és nem történhet futási hiba (pl. illetéktelen memória hozzáférés).*

**Ellenőrzés** Feltöltés előtt érdemes ellenőrizni a megoldásod.

1. **Fordítás** Ellenőrizd, hogy a programod lefordul-e! A bíró a `gcc -O2 -static -o feladat feladat.c` paranccsal fordít, érdemes ezt használni. A `-Wall` kapcsoló is hasznos lehet.
2. **Példa tesztesetek** Ellenőrizd, hogy a programod helyesen működik-e! A `minta.zip` tartalmaz a bíró által futtatott tesztesetek közül feladatonként egyet-egyet. Az első feladat teszteléséhez másold a programod mellé az `ex1.be` fájlt `be.txt` néven, futtasd le a programod, majd az így kapott `ki.txt` tartalmát hasonlítsd össze az `ex1.ki` fájlban található referencia kimenettel.
3. **Extra tesztesetek** Ellenőrizd a programod működését további példák segítségével! Néhány további tesztet is elérhető, de ezek csupán ellenőrzésre használhatóak, a bíró nem futtatja őket. Ezek használatához futtasd a programod a `-t` vagy `-test` kapcsolóval, például a `./feladat -test` paranccsal. Csak az első feladat teszteléséhez futtasd a programod a `./feladat -t 1` paranccsal.

### 1. feladat: AJANDEK (1 pont)

Október van, szóval már elég közel van hozzánk a karácsony, így elérkezett az idő, hogy a mikulás előkészítse az ideai ajándékokat, hogy aztán betörjön idegen emberek lakásába a kéményükön keresztül, megegye a másnapra szánt süteményt és egyéb édességeket, és egy random, kétes eredetű csomagot hagyjon a nappali közepén, amiben a gyanútlan lakosok esetleg felbotlanak és meghalnak. Segíts a mikulásnak az akciójában!

Készítsd el az 'Ajandekstruktúrát, amelyről tároljuk el, hogy mekkora a doboz mérete, illetve mekkora a tomege. A doboz mérete `cm3`-ban, a tömeg `gramm`-ban van, csak hogy ne kelljen a mikulás manóinak valós számokkal számolniuk, mivel az agyi kapacitásuk megegyezik egy manóéval. Persze ezzel nem sértegetni akartam őket, mivel hát ők manók, szóval ezt a mondatot akár ki is hagyhattam volna.

A struktúra mezőit a fent leírt sorrendben hozd létre!

### 2. feladat: SZÁN (1 pont)

A mikulásnak sikerült előkészítenie az ajándékokat, így indulhat a betörés! Akarom mondani az akció! Mármost az ajándékosztás! Az ajándékokat a mikulás a szánján fogja elszállítani, amelyet a gyönyörűséges rénszarvasok húznak, akikről majd később lehet még mesélek. Azonban a szánnak is vannak korlátai.

Készítsd el a Szanstruktúrát. A szánról tudjuk, hogy legfeljebb hány `cm3`-nyi térfogatú csomagot bír el, illetve maximálisan hány g ajándék szállítható vele. Emellett azt is tároljuk el, hogy hány rénszarvas húzza majd a szánt.

A struktúra mezőit a fent leírt sorrendben hozd létre!

### 3. feladat: DE NEHÉZ A SZÁN (4 pont)

Azért a szeretet ünnepén gondolnunk kell a rénszarvasokra is, akik minden erejükkel azon vannak, hogy a

mikulás akcióját segítsék. Háláljuk meg nekik ezt azzal, hogy nem szállítatunk velük túl sok rakományt egyszerre!

Valósítsd meg a 'maximalis\_rakomany\_tomeg' függvényt, amely paraméterben a szánt várja, illetve a mikulás tömegét kg-ban (1 tizedesjegy pontossággal, 1 kg = 1000 g).

A rakomány maximális tömegét úgy számítjuk ki, hogy a szán által elbírta maximális tömegből kivonjuk a mikulás tömegét. Azonban ha a szán 5-nél kevesebb rénszarvas húzza, akkor ezt a tömeget el kell osztani 2-vel (lefelé kerekítve).

Példa input:

Szán maximális tömege: 150.000 g

Rénszarvasok száma: 3

Mikulás tömege: 120 kg

Példa output:

15000

Magyarázat:

A szán 150.000 g-nyit tud szállítani, ebből 120.000 g a mikulás, marad 30.000 a rakománynak. Viszont 5-nél kevesebb (3) rénszarvas húzza, így ezt el kell osztani 2-vel, marad 15.000 g (ami 15 kg).

#### 4. feladat: A GYEREKEK EGYRE JOBBAK (4 pont)

Miközben a mikulás és titkos csapata az akció részleteit dolgozza ki, a manók különböző statisztikákat készítenek. Azt tapasztalták, hogy minden évben a jó gyerekek száma ugyanannyival növekszik. Ennek okát pontosan nem tudják, de valószínűleg az lehet a háttérben, hogy a gyerekek egész nap a telefonjukat nyomkodják, így nincs is lehetőségük rosszkodni, mint a régi szép időkben.

A manók szeretnék a mikulásnak prezentációt készíteni az elkövetkező évek várható adatairól. Segíts nekik ebben!

Valósítsd meg a 'jo\_gyerekek' függvényt, ami első paraméterben azt várja, hogy a megfigyelés kezdetén (2000-ben) hány jó gyerek volt. A második paraméter (növekmény) megmondja, hogy mennyivel növekszik a jó gyerekek száma évenként. A harmadik paraméter pedig megmondja, hogy melyik a kérdéses év, amikor ki szeretnénk deríteni, hogy várhatóan hány jó gyerek lesz. A függvény adja vissza, hogy abban az évben várhatóan hány jó gyerek lesz.

Példa input:

1000, 15, 2007

Példa output:

1105

Magyarázat:

2000-ben 1000 jó gyerek volt, és minden évben 15-tel növekszik, tehát (2001: 1015; 2002: 1030; 2003: 1045; 2004: 1060; 2005: 1075; 2006: 1090; 2007: 1105).

#### 5. feladat: INFLACIO (6 pont)

A mikulás manói nem csak a gyerekek számában vettek észre összefüggéseket, hanem az ajándékok áraiban, ugyanis azt tapasztalták, hogy az infláció (illetve a jó gyerekek számának növekvése) miatt minden évben ugyanannyi növekszik az ajándékok költsége az előző évhez képest. A kérdés, hogy egy adott időszakban összesen mennyit költöttek Mikulásék ajándékokra, ebben kérjük az ön segítségét, hiszen ők nem matematikusok, ön viszont lehet, hogy igen.

Először is legyen kedves elkészíteni az 'Idoszakstruktúrát, amely egy időszakot ír le, tehát két mezővel rendelkezik (az időszak kezdetét, illetve végét jelölő évek).

Ha ezzel elkészült, utána tesszen megvalósítani az 'ossz\_koltseg' függvényt, amelynek 3 paramétere van:

- az ajándékok költsége 2000-ben
- a növekmény, ami megmondja, hogy minden évben hány
- egy időszak, ami a kezdeti és végső évszámot tartalmazza.

A függvény számítsa ki az ajándékok összköltségét a megadott időszakban!

Példa input:

15000, 10, 2003-2005

Példa output:

66084.15

Magyarázat:

2000-ben összesen 15.000 \$-ba kerültek az ajándékok, amely költség minden évben 10A 2003-2005-ös időszakra vagyunk kíváncsiak (2001: 16500 ; 2002: 18150 ; 2003: 19965 ; 2004: 21961.5 ; 2005: 24157.65). Ebből az utolsó 3 (2003-2005) összegére vagyunk kíváncsiak, ami összesen 66.084,15.

**6. feladat: BUNCSELEKMENY** (4 pont)

Kiderült, hogy a mikulás, aki a betöréseket tervezte, igazából nem a valódi mikulás volt, hanem egy szélhámos, és csak álca volt a mikulásruha és az ajándékok, igazából elvitte a gyanútlan lakosok értékeit :( Szerencsére a rendőrség már intézkedik, elkapták az álmikulást és börtönbe zárták.

A rendőrség szeretné meghatározni, hogy összesen mennyi kárt okozott az álmikulás. Tedd jóvá a korábbi cselekedetedet, miszerint segítetted az álmikulás akcióját, és most segítsd a rendőrök munkáját. Utána majd dönts el, hogy melyik oldalhoz szeretnél tartozni!

Valósítsd meg a 'karbecsles'függvényt, amely paraméterben megkapja a károk mértékét egy tömbben, illetve a tömb méretét. A függvény adja vissza az össz-kárt ezek alapján.

Figyelem! A rendőrség nagyon szereti a rekurziót! A függvényt rekurzív módon valósítsd meg! A megállási feltétel a legkorábbi olyan eset legyen, ahol számítás nélkül lehet tudni a visszatérési értéket!