

Frissen létrehozva. A feladatsor az észrevételek alapján még minden bizonnyal frissülni fog.

Bevezetés az informatikába – Gyakorló feladatsor (GI BSc, Info tanár)

Ezen feladatsor célja kellő „tápanyagot” adni az otthoni gyakorláshoz. A feladok megoldásával a kezdő programozók is bátran próbálkozhatnak, ugyanakkor számos feladatnál több mindenre oda kell figyelni, mint amit a gyakorlaton megszokhatott. Ha szükséges, olvassa át többször is a feladat szövegét és haladjon lépésenként. A feladatok sorszáma után írt ♦ száma utal a feladat nehézségére. A ♦ jel után megjelenő ? azt jelzi, hogy a segítő információk nélkül a feladat már egy magasabb kategóriába tartozna.

1. (♦) Számolja ki, mekkora volt a hajó elmozdulása, ha X km-et haladt keleti, majd Y km-et északi irányba. Az X és Y értékét (valós) a billentyűzetről olvassa be a program.

Elmozdulás alatt a kiindulópont és a végpont távolságát értjük.

2. (♦) Készítsen programot, mely hidegtálrendelésénél segít a vevőknek a mennyiség meghatározásában. Minden hidegtál 25 dkg köretet és 20 dkg feltétet tartalmaz.
 - a. A program kérje be a felnőttek, kamaszok, a nagyon idősek és a kisgyerekek számát.
 - b. Számolja ki a köretből és a feltétből szükséges mennyiséget: (ezt nem kell megjelenítenie)
 - i. kisgyerekek/nagyon idősek: 6 dkg köret, 3 dkg feltét
 - ii. felnőtt: 10 dkg köret, 5 dkg feltét
 - iii. kamasz: 12 dkg köret, 6 dkg feltét
 - c. Határozza meg, hogy hány hidegtálat kell rendelnie a vevőnek, hogy mindenkinek legalább annyi étel jusson (köretből és feltétből is), mint a javasolt mennyiség, és a kapott értéket jelenítse meg a képernyőn. A hidegtálat darabra árulják, nem kérhet belőle „tört” részt.

3. (♦) Számolja ki, egy három termékből álló termékkosár ára alapján, hogy az ideai ár hány százaléka a 2010-es árnak (egész szám). A termékek ideai és 2010-es árát is a billentyűzetről olvassa be. (A kiírás százalékban történjen, a törtrészt 2 tizedes pontossággal megjelenítve.)

Segítség: A 3-3 egész értéket kell beolvasnia évenként, de figyeljen rá, hogy a végeredmény már egyetlen valós érték. A termékkosár ára alatt a benne szereplő termékek árának összegét értjük.

4. (♦♦♦) Készítsen programot a hallgatók zh eredményeinek kezelésére, ahol minden diákot a sorszáma (egész érték) azonosít. A program soronként olvasson be egy-egy sorszám, pontszám párt. A beolvasásnak vége, ha a sorszám nulla értékű. Jelenítse meg a következő értékeket:
 - a. hány diák ért el legalább 10 pontot,
 - b. mennyi volt az átlagos pontszám,
 - c. azon diák sorszámát, aki a legjobb eredményt érte el. (Feltételezheti, hogy csak egy ilyen van.)

Segítség: Nincs szükség az adatok eltárolására, elegendő néhány változó a diákszám, 10-nél több pontot elérő diákszám, pontszám összeg és a legjobb sorszám tárolására.

Haladóknak: (♦♦♦♦)

Egész értékű sorszám helyett a neptunkódot használja a diákok azonosítására.

Ne éljen a c) pontnál megadott feltételezéssel. (Ekkor már tárolás szükséges.)

Frissen létrehozva. A feladatsor az észrevételek alapján még minden bizonnyal frissülni fog.

5. (♦♦) Olvasson be egy maximum 10 hosszúságú karaktersorozatot, melyben csak számjegyek szerepelnek. Az Ön dolga, hogy minden egyes számjegy helyett eggyel nagyobb számjegyet írjon. A 9-es számjegy helyett pedig 0-át szerepeltessen.

Segítség: Karakterekkel dolgozik. Ügyeljen arra, hogy a 9 nem azonos a '9'-cel.

6. (♦) A nagy receptje szerint 1kg lisztből 80db kis pogácsa lesz. Készítsen programot, mely bekéri a liszt mennyiségét kg-ban (valós) és megjeleníti a belőle süthető pogácsák számát. A nagy csak egész pogácsát süt!
7. (♦?) Készítsen programot, mely a vásárlókat tájékoztatja a gyümölcsök áráról az alábbi táblázat alapján. A program olvassa be a gyümölcs nevét, és írja ki a táblázatban szereplő árát. Ha a felhasználó más gyümölcs nevét írja be, jelenítsen meg „Nincs” üzenetet. (A gyümölcsnevek max. 20 karakter hosszúak.)

Név	Ár
alma	250
barack	500
eper	1600
meggy	300

(Két sztring összehasonlítására a string.h-ban lévő

*[int strcmp\(const char *str1, const char *str2\);](#)*

függvény használható. A függvény nullát ad vissza, ha a paraméterként átadott sztringek azonosak.

char szo[21];

int ertek = strcmp(szo, "alma"); //az ertek változó akkor nulla, ha a beolvasott szó alma.)

8. (♦) Számolja ki, hány óra múlva éri utol az X teknős az Y teknőst, ha X teknős v_1 , Y teknős v_2 sebességgel halad, és Y teknősnek 0.1 km előnye van az indulásnál. A teknősök sebességét a billentyűzetről olvassa be (km/h mértékegységet feltételezve). Ha az X teknős soha nem fogja utolérni a Y-t, akkor jelenjen meg a „soha” üzenet a képernyőn, különben a szükséges idő (órában).

*Segítség: megtett út = idő * sebesség. Írja fel a képletet mind a két teknősre, és a két egyenletből fejezze ki az időt. A Y teknősnél ne felejtse el az előnyt is hozzáadni a megtett úthoz.*

9. (♦♦) Tegyük fel, hogy a vonat 100km/h utazósebességgel közlekedik a megállók közti szakaszokon, viszont a megállóba való beérkezés és az onnan való elindulás is lassú, ilyenkor 2-2 km-en keresztül az átlagsebesség csak 50km/h.
- Olvasson be egész értékeket egészen addig, ameddig 4 alatti értéket nem olvas. Ezek a megállók közti távolságot jelentik km-ben.
 - Számítsa ki, hogy hány óra alatt teszi meg a vonat a legelsőtől a legutolsó megállóig a teljes távot, és jelenítse meg a képernyőn 2 tizedes jegy pontossággal.

P1: input: 10 20 15 0 output: 0.5

Indoklás: 10-4 + 20-4 + 15-4 km-en halad utazósebességgel a vonat, míg 4 + 4 + 4km-en át 50km/h-val.

Frissen létrehozva. A feladatsor az észrevételek alapján még minden bizonnyal frissülni fog.

Segítség: $összes_idő = utazósebességgel_megtett_út / utazósebesség + lassan_megtett_út / lassú_sebesség.$

10. (♦) A Google Map egyik szolgáltatása, hogy a választott útvonalnál megjeleníti a szükséges időt is. Tegyük fel, hogy a térképünk részletessége azt is lehetővé teszi, hogy a választott útvonalon megismerjük a kereszteződések típusait. Készítsen programot, mely inputként bekéri a távolságot (valós érték, kilométerben megadva), bekéri a közlekedési lámpával ellátott kereszteződések és a „sima” kereszteződések számát. Becsülje meg a menetidőt és írja ki a képernyőre a következő értékekkel számolva:
- sebesség: 60km /h,
 - közlekedési lámpával ellátott kereszteződés: + 3 perc (átlagérték),
 - „sima” kereszteződés: +6 perc.

Ha a számolt idő meghaladja az egy órát, akkor jelenítse meg
órában **két tizedes jegy** pontossággal,
különben
percekben, egészen **kerekítve**.

Segítség: ne felejtse el átváltani a percekben megadott értékeket órába a számításnál, vagy fordítva.

11. (♦?) Készítsen kerekítő programot. Az 1, 2 forintos címletű érmék bevonása után a kerekítés szabályai az alábbi formában változtak meg hazánkban készpénzes fizetésnél (feltételezze, hogy századokra kerekített értéket kap meg):
- a 0,01 forinttól 2,49 forintig végződő összegeket lefelé, a legközelebbi 0-ra,
 - a 2,50 forinttól 4,99 forintig végződő összegeket felfelé, a legközelebbi 5-ösre
 - az 5,01 forinttól 7,49 forintig végződő összegeket lefelé, a legközelebbi 5-ösre
 - a 7,50 forinttól 9,99 forintig végződő összegeket felfelé, a legközelebbi 0-ra

kell kerekíteni.

Segítség:

- Valós számot olvasson be.
- Ossza el öttel.
- Kerekítse egészen az eredményt. (Már tárolhatja egészként.)
- Szorozza meg öttel az eredményt.
- Írja ki az eredményt. (Egész szám).

12. (♦♦?) Olvasson be egy egész számot ($n \geq 2$) és írja fel a prímtényezősz alakját.

Segítség: Egy egyszerű módszer a számításához:

input: n

$x \leftarrow 2$

Ameddig n nagyobb, mint 1

Ha n osztható x -szel, akkor:

a) Jelenítse meg/tárolja el az x értékét (prímtényező)

b) $n \leftarrow n/x$

Különben

c) $x \leftarrow x+1$

Frissen létrehozva. A feladatsor az észrevételek alapján még minden bizonnyal frissülni fog.

13. (♦♦?) Készítsen programot, mely segít Hókuszpóknak elkapni Okoskát. Okoska a mai naptól kezdve minden n ., Hókuszpók minden m . napon jár a bűzös mocsárnál. Olvassa be az n és az m értékét vesszővel elválasztva a billentyűzetről (egészek), és írja ki a képernyőre, hány nap múlva kell Hókuszpóknak magával vinnie a törpcspadát (a legkisebb értéket adja meg).

Pl: input: 2,3 output: 6
Pl: input: 6,9 output: 18
Pl: input: 36,6 output: 36

Segítség: A fekadathoz a legkisebb közös többszöröst kell meghatározni:

Egy egyszerű mód a számításra:

input: a, b

$x \leftarrow a$

$y \leftarrow b$

Ameddig x nem azonos y -nal

Ha x kisebb, mint y , akkor $x \leftarrow x+a$,

Különben $y \leftarrow y+b$

Nagyobb számoknál hatékonyabb ez a megoldás: (♦♦♦♦)

input: a, b

1. Határozza meg mindkét szám prímtényező felbontását. Tárolja el azt is, melyik prímtényezőt hányszor kell felhasználni a szám előállításához:

$$a = p_1^{k_1} p_2^{k_2} \dots p_n^{k_n}$$

$$b = q_1^{h_1} q_2^{h_2} \dots q_m^{h_m}$$

2. Készítsen egy listát, amiben minden prímtényező egyszer szerepel. Ha mindkét számnál szerepel a prímtényező, akkor a nagyobb kitevővel vesszük fel a listába.

3. Ha a lista elemeit $(a_1, b_1), (a_2, b_2), \dots, (a_j, b_j)$ jelöli, akkor a legkisebb közös többszörös: $a_1^{b_1} a_2^{b_2} \dots a_j^{b_j}$

14. (♦♦?) Készítsen programot, mely beolvas két törtet és

- a. kiírja a törtek összegét tört alakban egyszerűsítés nélkül.

input: output:
1/6 7/18
2/9

Segítség: A közös nevező a két szám legkisebb közös többszöröse.

- b. kiírja tört alakban egyszerűsített alakban.

Segítség: Az egyszerűsítéshez határozza meg a számláló és a nevező legnagyobb közös osztóját, és azzal ossza le mind a két értéket. Ha nem talált közös osztót, akkor a tört nem egyszerűsíthető.

15. (♦♦) Készítsen egy „kassza” programot. Minden terméket azonosítóval tárolunk, mely egy egész érték (minden termékre egyedi).

- a. Olvasson be <azonosító>, <vásárolt mennyiség> párokat egészen addig, ameddig a pár első értéke nem nulla. (Az egyszerűség kedvéért minden terméket darabra árulunk, tört rész nem értelmezett.)
- b. Az alábbi táblázat szerint határozza meg, hogy mennyit kell fizetni a vevőnek a vásárolt termékekért. A bolti kerekítés szabályai szerint írja ki a fizetendő összeget.

Frissen létrehozva. A feladatsor az észrevételek alapján még minden bizonnyal frissülni fog.

Azonosító	Ár	Utolsó számjegy alapján	Kerekítési mód	Érték -> Fizetendő
1	250	3-nál szigorúan kisebb	lefele, tízesre	102Ft -> 100Ft
2	500	3 és 7 között (határokkal együtt)	legközelebbi ötösre	103Ft -> 105Ft
3	1600	7-nél szigorúan nagyobb	felfele, tízesekre	109Ft -> 110Ft

16. (◆◆◆?) Készítsen programot, mely egy árazón meghatározza, hogy melyik számlálón hányat kell forgatni ahhoz, hogy a kívánt összeg szerepeljen majd az árcédulán. A címkéző 5 számjegyet képes megjeleníteni. Minden számjegyhez egy-egy fogaskerék tartozik, melyeken a decimális számjegyek szerepelnek növekvő sorrendben 0, 1, ...,9, majd egy üres mező áll.

- Olvasson be két szóközzel elválasztott egész számot. Az első szám az árcímkézőn most szereplő árat, a második a kívánt árat jelenti.
- Határozza meg minden fogaskerékre a szükséges forgatások számát. (A kerekeket csak a nagyobb számjegy irányába lehet tekerni, majd az üres mező következik, és utána fordul át újra a 0-s értékre.)
- Jelenítse meg sorrendben, a legfelső helyi értékű jegytől indulva, a szükséges forgatások számát.

Pl: (az üres helyeket x-szel jelölve a láthatóság érdekében)

```
input: 125 555    output:    00430
magyarázat:
  output címke: xx555
  inputcímke:   xx125
  tekerés:     00430
  (a tekerés a számjegyek különbsége)
```



```
input: 149 25    output:    00916
magyarázat:
  output címke: xxx25
  inputcímke:   xx149
  tekerés:     00986
  (átfordulásnál a tekerés a számjegyek különbsége + 10)
```

Tipp: Az üres mezőt tekintse 10-es értéknek a számolás során.

17. (◆◆) Készítsen programot az olajár változásának szimulálására. A modell szerint az olaj világszerte minden páratlan sorszámú nap 5%-kal, páros sorszámú nap véletlenszerűen változik +/-10%-kal.

- Kérje be a kiindulási napra (**1. nap**) vonatkozó olajárát (egész) és egy n (egész) értéket.
- Határozza meg az n . napra vonatkozó olajárát. (Az n . napi emelkedés már ne legyen benne.)
- Mivel a szimuláció véletlenszerű elemeket tartalmaz, ezért 3 szimuláció eredményét is jelenítse meg. A kiírás minden egyes szimulációnál:

<szimuláció_sorszama> szimuláció <érték> Ft/liter
formában történjen 2 tizedes pontossággal.

Pl:
1. szimuláció 450.81 Ft/liter
2. szimuláció 420.09 Ft/liter
3. szimuláció 550.21 Ft/liter

Frissen létrehozva. A feladatsor az észrevételek alapján még minden bizonnyal frissülni fog.

18. (◆◆?) Készítsen programot, mely különböző egyszerű statisztikákat határoz meg a hőmérsékleti adatok alapján. Az adatok (egész számok) szóközzel elválasztva érkeznek, maximum 10 darab. A sorozatot EOF jel zárja.

A program írja ki a következőket:

- a) beolvasott értékek száma
- b) az átlaghőmérséklet
- c) az átlaghőmérsékletnél melegebb napok száma
- d) azon napok száma, amikor fagypont alatt (0 fok) volt a hőmérséklet
- e) a legmagasabb hőmérsékletű napok sorszáma
- f) az i . és $(i-1)$. nap hőmérsékletkülönbsége ($i = \{1, 2, \dots, \text{elemszám}-1\}$)

Segítség:

A scanf függvény visszatérési értéke EOF, ha véget ért a sorozat.

Vegyen fel egy változót, ami minden beolvasás után azt tartalmazza, hogy hány elem van a tömbben. (Kezdetben 0 legyen az értéke. Ha beolvasott egy elemet a tomb[elemszam] helyre, akkor növelje az elemszámot eggyel.)

Ügyeljen rá, hogy csak azokat a tömbelemeket dolgozza fel, ahol van érték.

19. (◆◆◆) Készítsen programot, mely egységesíti a cikkeknek adott címeiket. A cél, hogy minden egyes szó nagy kezdőbetűvel kezdődjön. A címbeli szavak minden esetben szóközzel vannak elválasztva egymástól.
- a. Olvasson be egy sort a billentyűzetről (max. 40 karakter).
 - b. Írja ki a leírásnak megfelelően átalakított szöveget.

pl:

```
input: szorgalmasak A debreceni gi-sek
output: Szorgalmasak A Debreceni Gi-sek
```

Segítség: A beolvasáshoz gets / fgets függvények valamelyikét használja, a scanf az első szóköznél megakad.

Pl: char szo[41]; fgets(szo,41, stdin);

Ügyeljen arra, hogy ne fusson túl se negatív, se pozitív irányba a tömbindex. Az első betűt nagybetűsíteni kell, ott nem kell ellenőrizni, hogy van-e szóköz előtte.

Haladóknak: (◆◆◆◆) Oldja meg, hogy a névelőket (a, az, egy) ne nagybetűsítse a program, kivéve, ha a cím azzal kezdődik.

20. (◆◆) Dekódoljon egy titkosított üzenetet. A titkosító nagyon kezdetleges módszert választott, egész egyszerűen „felcserélte” az n és m , valamint az a és e betűket az üzenetben. (n helyett mindig m -et írt, m helyett mindig n -et, stb.)

Pl: elmentemaludni -> alnamtaneludmi

Az Ön feladata az üzenet visszaalakítása és megjelenítése. Az üzenet maximum 255 karakter hosszúságú, és csak betűket tartalmaz (mert a titkosítás során a szóközöket eltávolították és kisbetűsítették az üzenetet).

Input: alnemtaneludmi output: elmentemaludni

Frissen létrehozva. A feladatsor az észrevételek alapján még minden biztonnyal frissülni fog.

21. (♦♦) Készítsen programot, mely képes meghatározni azt, hogy egy fertőzés hány nap múlva éri el a kritikus szintet. Az aktuális napot 0-val jelöljük. A program olvassa be a billentyűzetről:
- a fertőzött személyek számát (akik már betegek)
 - a fertőzés terjedésének sebességét: 1 fertőzött hány új személyt fertőz meg 1 nap alatt,
 - a fertőzöttek kritikus számát.

Minden érték 0-nál szigorúan nagyobb egész szám. A fertőzést nem lehet több helyről elkapni, a megfertőzött személyek különbözőnek tekinthetők.

Pl:

```
input: 10, 3, 5   output: 0
//a fertőzöttek kritikus száma már most meghaladja a kritikus értéket

input: 2, 3, 15  output: 2
//fertőzöttek: 0. nap 2 fő, 1. nap 6 fő, 2. nap 18 fő
```