

FELADATOK

I. Utasítások

1. Olvassunk be egy karaktert! Írjuk ki, hogy az illető karakter kisbetű-e!
2. A billentyűzetről beolvassunk egy egész számot. A programunk vizsgálja meg, hogy a szám osztható-e 12-vel, és írjon ki a képernyőre egy üzenetet!
3. Olvassunk be egy évszámot! Állapítsuk meg, hogy az illető év szökőév-e!
4. Állapítsuk meg egy beolvasott karakterről, hogy magánhangzó vagy sem!
5. Számítsuk ki az $f : \mathbf{R} \rightarrow \mathbf{R}$ függvény értékét az x pontban, ha az f a következőképpen van értelmezve:
 - $$f(x) = \begin{cases} 2, & \text{ha } x < -1 \\ x^2 - 1, & \text{ha } -1 \leq x < 1 \\ \sqrt{x}, & \text{ha } x \geq 1 \end{cases}$$
6. Olvassunk be két egész számot, a nagyobbat osszuk el a kisebbel, majd írjuk ki a hányadost és a maradékot!
7. Számítsuk ki egy általunk beolvasott számnak az abszolút értékét az Abs függvény segítségével!
8. Írjunk algoritmust, amely megoldja a következő egyenletrendszert:
 - $$\begin{cases} ax + by = c \\ dx + ey = f \end{cases}$$
9. Ellenőrizzük, hogy három beolvasott természetes szám pithagorászi szám vagy sem (azaz teljesítik-e a következő három feltétel közül az egyiket: $a^2=b^2+c^2$, $b^2=a^2+c^2$, $c^2=a^2+b^2$)!
10. Kérjen az algoritmus két számot, majd írja ki a köztük levő relációt! Például, ha a két szám 5 és $-3 \cdot 7$, akkor írja ki, hogy $5 > -3 \cdot 7$.
11. Olvassunk be két egész számot! Állapítsuk meg, hogy a második osztója-e az elsőnek!
12. Határozzuk meg a 2^x szám utolsó számjegyét! Az x természetes szám és a billentyűzetről olvassuk be.
13. Írjunk programot, amely egy, 1 és 7 közötti számot olvas be, majd a hét napjai közül kiírja az ennek megfelelőt!
14. Oldjuk meg az $ax = b$ egyenletet, ahol a és b valós együtthatók!
15. Olvassunk be két egész számot. Állapítsuk meg, hogy ezek egymás után következő számok, vagy nem!
16. Írjunk programot, amely bekéri, hogy a hétnek hányadik napja van, majd írja ki, hogy ez milyen nap! (Az első nap a Hétfő.)
17. Egy vállalatnál minden dolgozó a belépéstől számított 5 évenként külön jutalmat kap. A program kérje be a szolgálatban eltöltött időt (év) és írja ki, hogy jár-e a jutalom.
18. Írjunk programot, amely kérjen be három egész számot és írja ki, hogy van-e közöttük két azonos szám!
19. Tetszőleges, legfeljebb 4 jegyű egész számot írjunk fel római számokkal!
20. Készítsünk programot, amely egy hónapról megmondja, hogy melyik évszakba esik itt, Európában!
21. Készítsünk programot, amely egy 2006-os dátumról (hó, nap) megmondja, hogy milyen napra esik!

22. Készítsünk számítógép programot, amely az egész számok közötti alapműveletek (+, -, *, /, %) elvégzésére alkalmas! Védjük ki a nullával való osztást!
23. A rendőrség a 6-os úton, Marosvásárhelyhez közel méri a mellettük elhaladó autók sebességét. Készítsünk programot, amely bekéri megengedett sebesség értékét és az elhaladó jármű sebességét (km-ben), és abban az esetben, ha a jármű túllépte a megengedett sebességhatárt, írjuk ki annak mért sebességét!
24. Készítsünk programot, amely bekéri egy nyitott fűtési rendszerben keringő víz mennyiségét köbméterben, és az eltelt időt (óra, perc, másodpercben), majd megmondja, hogy mennyi vizet kell utántölteni, hogy ismét tele legyen a tartály! A rendszerből óránként 2 ezrelék párolog el.
25. Egy diszkóba aszerint lehet belépőjegyet kapni, hogy hány éves az illető. A belépőjegy egyben tombola is, fizetni érte csak a kapusnál kell majd, a legfeljebb húszévesek diákjegyet kapnak (20 RON), az idősebbek felnőtt jegyet (30 RON). A kapuban a diákjeggyel rendelkezők közül a lányoknak nem kell fizetniük a belépőjegyért, mindenki másnak a jegyen szereplő díjat. Készítsünk programot, amely a szükséges adatok bekérése után megállapítja és kiírja, hogy az éppen belépő vendég fizet-e és ha igen, mennyit!
26. Készítsünk programot, amely bekér két tetszőleges egész számot! Ha a két szám különbsége nem több 10-nél, akkor írassuk ki a két szám szorzatát, különben a kisebb szám és a nagyobb szám hányadosát 4 tizedesjegy pontossággal.
27. Szerencsejátékot játszunk! A gép generál egy 101 és 200 közé eső véletlen számot. Tippeljük meg ezt az értéket! Ha éppen eltaláltuk, akkor 10 lej a jutalom, ha maximum 10 egységnyi az eltérés, akkor 1 lej a nyeremény és maximum 20 egység eltérés esetén is nyertünk 1 lejt. Írassuk ki a gép által generált számot, a tippelt számot és a nyeremény összegét!
28. Számítsuk ki, hány szökőév volt/van két évszám között!
29. Ha 50 dollárt a bankba teszünk, és a kamatláb $k\%$, mennyi pénzünk lesz n év múlva?
30. Írjunk programot kamatos kamat számítására!
31. Döntsük el egy beolvasott n -ről, hogy négyzetszám-e?
32. Olvassunk be számpárokat addig, amíg a két szám különbsége 0 nem lesz. Írjuk ki azt a párt, amelyben a tagok közti eltérés abszolút értéke a legnagyobb!
33. Egy beolvasott n számról döntsük el, hogy 2 hatványa-e!
34. Egy n számról döntsük el, hogy Fibonacci-szám-e!
35. Írjunk programot, amely egy tetszőleges n számról megmondja, hogy hány jegyű!
36. Egy tetszőleges valós számot addig harmadolunk, míg az eredmény 0,01-nél kisebb lesz. Írassuk ki a harmadolások számát!
37. Olvassunk be n darab számot, majd írjuk ki, hogy a beolvasottak hány százaléka páros, és hány százaléka páratlan!
38. Olvassunk be n darab számot, majd számoljuk meg, hány volt köztük negatív, hány pozitív és hány nulla!
39. Olvassunk be n darab egész számot, majd írjuk ki a negatívak közül a legnagyobbat, a pozitívak közül pedig a legkisebbet!
40. Olvassunk be a billentyűzetről n tanuló félévi jegyeit egy tantárgyból, majd írjuk ki az átmenő jegyek átlagát, valamint a bukók számát!
41. Olvassunk be n darab számot! Számoljuk meg, hány osztható közülük 3-mal, 5-tel és 15-tel!
42. Olvassunk be n darab számot. Számoljuk meg, hány olyan szám van a beolvasottak között, amelynek 11-gyel való osztási maradéka 5, és az első számjegye az utolsó számjegyének kétszeresével egyenlő!
43. Határozzuk meg két egész szám legnagyobb közös osztóját ismételt kivonással!

44. Olvassunk be egész számokat 0 végjelig! Határozzuk meg a beolvasott páros számok számtani középátlányosát!
45. Olvassunk be * végjelig egy tetszőleges karaktersorozat, és állapítsuk meg a magánhangzók számát!
46. Olvassunk be számokat 0 végjelig! Határozzuk meg a negatív és a pozitív számok összegét (külön-külön), majd írjuk ki a legnagyobb számot és annak sorszáma!
47. Írjunk programot, amely egy tizenhatos számrendszerbeli számot átalakít tízes számrendszerbelivé!
48. Olvassunk be egy dátumot: év, hónap, nap! Írjuk ki, hogy ez a dátum az év hányadik napja! Vigyázzunk a szökőévek esetében a február 29-ére és az ezt követő napokra!
49. Olvassunk be számokat 0 végjelig, és írjuk ki azok kettes számrendszerbeli alakját!
50. Készítsünk programot, amely kiírja a prímekeket 1000-ig!
51. Határozzuk meg és írjuk ki két beolvasott egész szám közötti számok összegét!
52. Írjunk programot, amely leegyszerűsíti az $\frac{m}{n}$ törtet (m és n természetes számok)!
53. Írjuk ki az összes háromjegyű prímszámot, amelyeknek a tükörképe is prím!
54. Adott egy szám p számrendszerben ($p < 10$). Írjuk ki a számot a 10-es számrendszerben!
55. Olvassunk be egy természetes számot! Ha nem Fibonacci-szám, bontsuk fel Fibonacci-számok összegére!
56. Számítsunk át órára, percre és másodpercre egy másodpercben megadott időt!
57. Fejezzük ki méter, centiméter és milliméter segítségével egy milliméterben megadott távolságot!
58. Olvassunk be egy természetes számot! Írjuk ki a benne szereplő páros számjegyek számát!
59. Adottak egy tetszőleges pont koordinátái. Írjunk programot, amely megmondja, melyik síknegyedben van a pont!
60. Készítsünk programot, amely kiírja az N -es szorzótáblát ($1 \times N =$, $2 \times N =$, ...)!
61. Döntsük el egy természetes számról, hogy Armstrong-féle szám-e! Egy számról azt mondjuk, hogy *Armstrong-féle*, ha egyenlő a számjegyei köbének összegével.
62. Beolvasunk egy egész számot és egy számjegyet. Számoljuk ki hányszor fordul elő az adott számjegy a számban!
63. Írjunk programot, amely megszámlolja, hogy hány nullás számjegy van egy adott n szám négyzetében!
64. Számítsuk ki n darab beolvasott szám számtani és mértani középátlányosát!
65. Számítsuk ki n darab beolvasott szám közül az $[a, b]$ intervallumba eső negatív számok összegét és szorzatát!
66. Írjunk programot, amely megállapítja, hogy legkevesebb hány számot kell összeadni ahhoz, hogy az $S = 1 + 2 + 3 + \dots$ összeg nagyobb legyen egy adott n számnál!
67. Adott n természetes szám esetén, határozzuk meg az $n!$ (n faktoriális) értékét!
68. Írassuk ki a következő számsorozatok első n darab tagját:
 - a) 3, 33, 333, 3333, ...
 - b) 3, 8, 13, 18, 23, 28, ...
 - c) 1, -2, 3, -4, ...
69. Számítsuk ki a következő kifejezéseket:
 - a) $S = 0 + 2 + 4 + \dots + 2n$;
 - b) $S = 1^3 + 2^3 + 3^3 + \dots + n^3$;

- c) $S = 1 - 2 + 3 - 4 + \dots + (-1)^{n+1}n$;
- d) $S = 1 + \frac{1+2}{1 \cdot 2} + \frac{1+2+3}{1 \cdot 2 \cdot 3} + \dots + \frac{1+2+\dots+n}{1 \cdot 2 \cdot \dots \cdot n}$.

70. Írjuk ki az $a_n = \left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n-1}\right)$ sorozat első n tagját és azok összegét!
71. Írjunk programot, amely a legkevesebb bankjegy segítségével kifizet egy adott összeget, ha tudjuk, hogy csak 500, 100, 50, 10, 5, 2 és 1 címletű bankjegyeink vannak!
72. Írassuk ki az összes n -nél kisebb ikerprímszámot! *Ikerprímszámnak* nevezünk két prímszámot, ha egymás után következő páratlan számok. Ikerprímek például: (3, 5), (5, 7), (11, 13).
73. Egy természetes számot *majdnem prímnek* nevezünk, ha felírható két (nem feltétlenül különböző) prímszám szorzataként. Ha két szomszédos természetes szám is majdnem prím, akkor nevezzük ezeket *iker-majdnem-prímeknek*. Ilyen például a 9 és a 10, hiszen $9 = 3 \cdot 3$, illetve $10 = 2 \cdot 5$, azaz mindketten majdnem prímek, és szomszédosak. Írjunk programot, amely egy megadott határig előállítja az összes iker-majdnem-prím párokat!
74. Egy beolvasott n számról döntsük el, hogy tökéletes-e! Egy szám *tökéletes*, ha egyenlő a nála kisebb pozitív osztóinak összegével. Például 28 tökéletes, mivel $28 = 1 + 2 + 4 + 7 + 14$.
75. Írjunk programot, amely megtalálja az összes egymilliónál kisebb tökéletes számot!
76. Két beolvasott számról döntsük el, hogy barátságosak-e! Két természetes szám *barátságos*, ha az egyik szám önmagánál kisebb pozitív osztóinak összege egyenlő a másik számmal, és fordítva. Ilyen számpár például a 220 és a 284, hiszen a 220 nála kisebb osztói: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110, és ezek összege 284; a 284 nála kisebb osztói: 1, 2, 4, 71, 142, és ezek összege 220.
77. Írassuk ki az összes n -nél kisebb barátságos számot!
78. Suryanarayana indiai matematikus *nagyon tökéletesnek* nevezte azt a természetes számot, amely osztói összegének osztóit összeadva, a szám kétszeresét kapjuk. Ilyen például a 16, mert osztóinak összege 31, a 31 osztóit összeadva pedig 32-t kapunk. Írjunk programot, amely nagyon tökéletes számokat keres!
79. *Palindrom* számokról már esett szó. A feladat az, hogy nem palindrom számokból állítsunk elő palindromokat a következőképpen: fordítsuk meg a számjegyek sorrendjét, s az így kapott számot adjuk az eredetihez. Ezt az eljárást addig folytassuk, amíg az összeg tükörszám nem lesz! Például a 86-ból indulva: $86 + 68 = 154$, $154 + 451 = 605$, $605 + 506 = 1111$.
80. Léteznek olyan számpárok, amelyeknek szorzata nem változik, ha számjegyeiket felcseréljük. Például 12 és 24, mivel $12 \cdot 42 = 504 = 21 \cdot 24$. Vajon van-e még ilyen számpár?

II. Tömbök és mutatók

1. Olvassunk be egy egész számokat tartalmazó tömböt! Írjuk ki az elemek közül a legkisebbet és a legnagyobbat, valamint ezek indexét!
2. Építsünk fel egy vektort az első n ($1 \leq n \leq 30$) 3-mal osztható számból!
3. Számítsuk ki egy egész elemeket tartalmazó tömb páros és páratlan elemeinek az átlagát!
4. Olvassunk be egy n ($1 \leq n \leq 100$) valós számot tartalmazó vektort! Határozzuk meg a negatív és a nem negatív elemek összegét!
5. Hozzunk létre egy egész számokat tartalmazó tömb elemeiből egy másik tömböt, amelynek elemei legyenek az adott tömb megfelelő elemeinek a négyzetei!
6. Olvassunk be n ($1 \leq n \leq 100$) természetes számot! Hozzunk létre ezekből egy új vektort, amely az előző vektor páratlan elemeit tartalmazza!

7. Számítsuk ki egy n ($1 \leq n \leq 50$) természetes számot tartalmazó tömb azon elemeinek összegét, amelyek oszthatók egy beolvasott a számmal!
8. Adott egy egész számokból álló sorozat. Írjuk ki azoknak a számoknak a pozícióját, amelyeknek az előjele megegyezik az utolsó szám előjelével!
9. Olvassunk be egy természetes számokból álló sorozatot! Számítsuk ki az első beolvasott számnál kisebb elemek összegét!
10. Számítsuk ki egy egész számokat tartalmazó vektor azon elemeinek a szorzatát, amelyeknek a paritása (előjele) ellentétes az utolsó elem paritásával!
11. Írjuk ki egy tömb azon elemeit, amelyek kisebbek, mint az első elem abszolút értéke!
12. Adott egy n ($1 \leq n \leq 30$) valós számot tartalmazó tömb. Számítsuk ki azon nullától különböző elemek ellentettjeinek összegét, amelyeknek az értéke kisebb az első elem értékénél!
13. Egy legtöbb 20 egész számot tartalmazó tömb elemeit másoljuk át egy másik tömbbe úgy, hogy a párosokat a tömb elejétől, a páratlanokat pedig a tömb végétől kezdve töltsük be az új tömbbe!
14. Állapítsuk meg, hogy egy n ($1 \leq n \leq 30$) elemű tömb számtani sorozatot alkot, vagy nem!
15. Hozzunk létre egy olyan kétdimenziós tömböt, amelynek minden eleme egyenlő saját két indexének összegével, majd írjuk ki a tömböt a képernyőre!
16. Írjunk olyan programot, amely előbb sorfolytonosan, majd oszlopfolytonosan listáz ki egy n soros és n oszlopos mátrixot, végül pedig csak a főátlón levő elemeket írja ki!
17. Írjunk programot, amely egy 4×4 -es mátrix főátlójába 1-et, a főátló feletti elemekbe 2-öt, a főátló alatti elemekbe 3-at ír! Irassuk ki a mátrix elemeit!
18. Írjunk programot, amely feltölti az egész típusú 3×3 -as mátrixot adatokkal, majd kiszámítja a főátlóbeli elemek szorzatát, a főátló feletti elemek összegét és a főátló alatti elemek szorzatát!
19. Növeljük egy adott a valós szám értékével egy n soros és m oszlopos ($1 \leq n, m \leq 50$), valós elemeket tartalmazó mátrix minden elemét! Írjuk ki a képernyőre az új mátrixot!
20. Adott egy n soros és m oszlopos ($1 \leq n, m \leq 50$), egész számokat tartalmazó kétdimenziós tömb. Töröljük a tömb k -edik sorát!
21. Határozzuk meg egy mátrix minden sorának a maximumát, valamint a teljes tömb maximumát!
22. Olvassunk be egy n soros és m oszlopos ($1 \leq n, m \leq 30$), egész számokat tartalmazó mátrixot! Határozzuk meg azoknak a soroknak a számát, amelyeknek az elemei szigorúan csökkenő sorrendet alkotnak!
23. Számítsuk ki egy valós számokat tartalmazó négyzetes mátrix főátlóján levő számok átlagát!
24. Számoljuk meg egy természetes számokat tartalmazó kétdimenziós tömb azon sorait, amelyek azonos elemeket tartalmaznak!
25. Beolvasunk két n soros és m oszlopos ($1 \leq n, m \leq 100$), valós számokat tartalmazó mátrixot. Állapítsuk meg, hogy a két mátrix azonos-e vagy sem! Írjunk ki megfelelő üzenetet!
26. Számítsuk ki egy egész számokat tartalmazó mátrix azon páros elemeinek az összegét, amelyeknek mindkét indexe páros!
27. Írjunk programot, amely ellenőrzi, hogy egy $n \times m$ -es, valós számokat tartalmazó kétdimenziós tömb valamennyi eleme hozzátartozik-e a $[0, 1]$ intervallumhoz és azt, hogy minden sor összege egyenlő-e 1-gyel! Írjunk megfelelő üzenetet a képernyőre!
28. Állapítsuk meg egy beolvasott karaktersorozatról, hogy megegyezik-e a tükörképével!

29. Határozzuk meg egy beolvasott szó előfordulásainak a számát egy karakterláncban!
30. Olvassunk be egy tetszőleges karakterláncot! Alakítsunk át minden kisbetűt nagybetűvé és minden nagybetűt kisbetűvé!
31. Adott egy maximum 255 számjegyű szám! Írjuk ki a számot alkotó különböző számjegyeket!
32. Hasonlítsunk össze két karakterláncot, és döntsük el, hogy azonos betűkből állnak-e!
33. Állapítsuk meg egy legtöbb 255 számjegyű számról, hogy megegyezik-e a tükörképével!
34. Olvassunk be több karakterláncot * végjelig! Írjuk ki a leghosszabbat!
35. Írjunk programot, amely egy adott szövegből kivesz egy adott jelt vagy jelsorozatot, és az ezt követő első betűt nagybetűre írja át!
36. Írjunk programot, amely egy karakterláncban minden szó első betűjét nagybetűvé írja át!
37. Olvassunk be több szót a billentyűzetről! A szavak felhasználásával hozzunk létre egy karakterláncot úgy, hogy a szavakat szóközzel válasszuk el egymástól! Határozzuk meg a szavak számát!

III. Függvények

1. Javítsuk ki a következő függvényben lévő hibákat! Mit csinál a következő függvény?

```
void mi_ez (int* a, int na, int* b, int nb, int* c, int nc){
    int *enda = a + na, *endb = b + nb;
    while (a < enda && b < endb){
        if (*a < *b)
            *c++ = *a++;
        else
            *c++ = *b++;
    }
    while (a < enda)
        *c++ = *a++;
    while (b < endb)
        *c++ = *b++;
    nc = na + nb;
}
```

2. Készítsen függvényt, amely meghatározza, hogy egy hónap, nap alakban megadott dátum az év hányadik napjára esik, ha pedig rossz valamelyik paraméter, akkor -1 legyen a visszaadott érték!
3. Készítsünk függvényt, amely megadja két egész szám legkisebb közös többszörösét!
4. Szimuláljuk az strlen függvényt!
5. Írjunk függvényt, amely megfordít egy szót!
6. Írjunk függvényt, amely meghatározza egy tömb legkisebb elemét!
7. Írjunk függvényt, amely leegyszerűsíti az $\frac{m}{n}$ törtet (m és n természetes számok)!
8. Adott egy szám p számrendszerben ($p < 10$). Írjuk ki a számot a 10-es számrendszerben!
9. Olvassunk be egy természetes számot! Ha nem Fibonacci-szám, bontsuk fel Fibonacci-számok összegére!
10. Számítsunk át órára, percre és másodpercre egy másodpercben megadott időt!
11. Fejezzük ki méter, centiméter és milliméter segítségével egy milliméterben megadott távolságot!
12. Írjuk ki 1-től 1000-ig a római számokat! Használjunk függvényeket!
13. Olvassunk be egy természetes számot! Írjuk ki a benne szereplő páros számjegyek számát egy alprogram segítségével!

14. Adottak egy tetszőleges pont koordinátái. Írjunk programot, amely megmondja, melyik síknegyedben van a pont!
15. Döntsük el egy természetes számról, hogy Armstrong-féle szám-e! Egy számról azt mondjuk, hogy *Armstrong-féle*, ha egyenlő a számjegyei köbének összegével.
16. Beolvasunk egy egész számot és egy számjegyet. Számoljuk ki hányszor fordul elő az adott számjegy a számban!
17. Számítsuk ki n darab beolvasott szám számtani és mértani középátlását!
18. Számítsuk ki n darab beolvasott szám közül az $[a,b)$ intervallumba eső negatív számok összegét és szorzatát!

IV. Összetett adatszerkezetek

1. Egy szállításról a következő adatokat tartjuk nyilván: árukód, irányítószám, cím, mennyiség. Az adatokat tömbben tároljuk. Ha az árukód beolvasásakor **Enter**-t ütünk, befejeződik a beolvasás. Ellenőrizzük az esetleges hibákat! A bevitel után írjuk ki a szállított áruk kódját és mennyiségét, valamint azt, hogy összesen hányféle áru szerepel a feldolgozott szállításban!
2. Adjunk kezdőértéket az alábbi példában szereplő `tanulo` nevű struktúrának!

```
typedef struct {
    char* csnev, knev;
    double atlag;
} tanulo;
tanulo t;
```

3. Használjuk az alábbi deklarációt:

```
typedef double tomb2[3][3];
tomb2 a;
```

Olvassuk be az adatokat és számítsuk ki a főátlóban lévő adatok összegét és a mátrix elemeinek átlagát!

4. Írjunk programot, amely nyilvántartja és követi egy verseny adatait! Minden egyes versenyző szereplése után írja ki a pillanatnyi első három helyezett nevét és pontszámát!
5. Készítsünk programot egy autókereskedésben található használt autók nyilvántartásához! Az autók márkáját, típusát, megtett kilométerét, gyártási évét és árát tároljuk. Keressük ki a legolcsóbb és legdrágább autót!
6. Írjunk programot, amely különböző autók fogyasztását és megtett útjainak hosszát tartja nyilván autónként és naponként! Számítsuk ki a 7 nap alatt megtett utat és a teljes fogyasztást! Az eredményt és az autók sorszámát írjuk ki a képernyőre!
7. Írjunk programot, amely egy vállalkozás alkalmazottairól a képesítésük függvényében háromféle listát készít! Ha az illető szakmunkás, akkor a szakot vesszük figyelembe, ha mérnök, az általa elvégzett egyetem nevét, valamint tanulmányainak záróévét írjuk a rekordba. A könyvelő adatai közé az eddig lekönyvelt évek száma, valamint gyermekeinek a száma kerül. Ezen kívül, minden alkalmazottnak ismernünk kell a nevét és a fizetését.

V. Állományok

1. Adott egy szöveget tartalmazó `szoveg.in` nevű állomány. Másoljuk át a `szoveg.out` állományba az adott állomány nem üres sorait!
2. Olvassunk be két szöveges állományt, majd illesszük össze őket (másoljuk a második tartalmát az első állomány végére)!

3. Töltsünk fel egy `in.txt` állományt úgy, hogy minden sora három, szóközzel elválasztott valós számot tartalmazzon! Készítsünk egy `out.txt` állományt, amelynek minden sorába írjuk be az `in.txt`-ben levő három szám által alkotott háromszög típusát!
4. Írjuk ki a `prim.txt` állományba az első n ($2 \leq n \leq 100$) prímszámot úgy, hogy soronként csak 15 szám szerepeljen!
5. A `szamok.txt` egész számokat tartalmaz. Számítsuk ki a számok összegét és átlagát!
6. Számoljuk meg egy adott állományba írt páros és páratlan számok számát!
7. Adott egy szöveges állomány. „Töröljük” az állomány minden második sorát!
8. Adott a `szam.in` állomány, amely egész számokat tartalmaz, szóközzel elválasztva. Építsük fel a `szam.out` állományt úgy, hogy minden sorában az előbbi állomány megfelelő sorában szereplő legkisebb szám szerepeljen!
9. Egy autószervíz számítógépe minden megbízásról egy-egy sort ír a `szerviz.txt` állományba. Egy sor a következő adatokat tartalmazza:

```
RENDSZÁM           : 6 karakter
TULAJDONOS NEVE    : 40 karakter
FORG. ENG. SZÁMA    : 8 karakter
DÁTUM              : 11 karakter, (2005.01.03.)
ÖSSZEG             : 6 karakter (egész szám)
```

Írjon C programot, amely a `szerviz.dat` állomány alapján névsorban kiírja azoknak az autótulajdonosoknak a nevét, akik több különböző autót is javítottak a szervízben 2005 januárjában és februárjában! Az állományt csak egyszer olvashatja be! Használjon dinamikus adatszerkezetet!

10. Írjunk programot, amely indítási paraméterként egy szöveges állomány nevét és 3 valós számot (x_0, y_0, R) kap. Az állomány pontok koordinátáit (x, y) tartalmazza. Az összetartozó koordinátapárok az állományban egymás után következnek, de nem tudjuk, hogy milyen sorrendben. Azt azonban feltételezheti, hogy az állomány „helyes”, vagyis minden x -hez tartozik y koordináta. A program az (x_0, y_0) koordinátájú ponttól mért távolságok növekvő sorrendjében írja ki azokat a koordinátákat, amelyek az (x_0, y_0) középpontú, R sugarú körön belül helyezkednek el. A koordinátapárok számát nem tudjuk előre, ezért használjon dinamikus adatszerkezetet!
11. Írjunk C programot, amely a `szoveg.txt` szöveges állományból kiírja azokat a szavakat amelyek magánhangzóval kezdődnek! Szónak tekintünk minden kis- és nagybetűkből álló karaktersorozatot. Szót határol minden nem betű karakter. Az állományt csak egyszer olvashatja be és nem használhat munkaállományt!
12. Írjunk C programot, amely a `szoveg.txt` szöveges állományban megszámlolja a csupa kisbetűvel írt szavakat! Szónak tekintünk minden kis- és nagybetűkből álló karaktersorozatot. Szót határol minden nem betű karakter.