$$i^i = e^{-\frac{\pi}{2}} \text{ vagy } i^{-i} = \sqrt{e^{\pi}} .$$

A számítástechnika fejlődésével az *e*-nek egyre több számjegyét sikerült kiszámolni. Versenyt is hirdettek ezzel a témával. 1999-ig az *e* $10^9$ nagyságrendű tizedes jegyet sikerült megállapítani.

***Példa.*** Egy egyszerű meghatározása az *e*-nek a UNIX alatti **bc** program segítségével történik. A **bc** program egy olyan nyelvet kínál, amelyen könnyen megfogalmazhatjuk a kívánt pontosságú számábrázolás mellett végezett matematikai műveleteket. A standard matematikai könyvtárat a **-l** parancssori opció megadásával tölthetjük be. A **scale** nevű változó értéke szabja meg, hogy hány tizedes pontossággal történjen a műveletek végzése.

Az *e* értékére az *e = exp(1)* összefüggést használhatjuk fel. A program a következő:

- elindítjuk a **bc** programot: **bc -l**
- beállítjuk a pontosságot: **scale=1000**
- kiadjuk a számítási utasítást: **e(1)**
- 5-6 másodperc után 1000 tizedesnyi pontossággal megkapjuk az *e* értékét:

2.71828182845904523536028747135266249775724709369995957496696762772407663035354759457138217852516642742746639193200305992181741359662904
357290033429526059563073813232862794349076323382988075319525101901157383418793070215408914993488416750924476146066808226480016847741185374234544243710753907774499206955170276183860626133138458300075204493
382656029760673711320070932870912744374704723069697720931014169283681902551510865746377211125238978442505695369677078544996996794686445490598793163688923009879312773617821542499922957635148220826989519366
803318252886939849646510582093923982948879332036250944311730123819706841614039701983767932068328237646480429531180232878250981945581530175671736133206981125099618188159304169035159888851934580727386673858
942287922849989208680582574927961048419844436346324496848756023362482704197862320900216099023530436994184914631409343173814364054625315209618369088870701676839642437814059271456354906130310720851038375051
011574770417189861068739696552126715468895703503544

**Kovács Lehel István**

# feladat megoldók rovata

## Kémia

**K. 413.** Hányszor nehezebb egy jód molekula, mint egy fluor molekula?

**K. 414.** Mekkora mólarányban tartalmaz etánt és butánt az a gázminta, amelyben mennyiségi elemzéskor 81,36 tömeg % szenet találtak?

**K. 415.** Milyen tömegarányban kevertek össze konyhasót mosószódával, ha a kapott elegy 22,64 tömeg % oxigént tartalmazott, s az összekevert anyagokat vegytisztának tekinthetjük ?

**K. 416.** Ammóniagyártáskor a kontaktkamrába nitrogén-hidrogén elegyet vezetnek 1:3 térfogatarányban. Mekkora a gázelegy sűrűsége standard körülmények között a reakciótérben?

**K. 417.** Az A vegyület gőzeinek oxigénre vonatkoztatott sűrűsége 2,375. Mennyiségi elemzésekor 15,79% szenet és 84,21% (m/m) ként találtak benne. Az adatok felhasználásával állapítsd meg a vegyület molekulaképletét!

**K. 418.** Barnaszénből 3,23g tömegű mintát széntartalmának meghatározásáért fölös mennyiségű oxigénben égették. Az égési gázokat először tömény kénsavoldaton, majd nátrium-hidroxid oldatba vezették, aminek a tömege 7,7g-al növekedett. Határozd meg a minta tömegszázalékos széntartalmát!

**K. 419.** A víz fagyáshője –6 kJ/mol. $1m^3$ víz tömege megváltozik-e fagyás közben, s ha meg, mennyivel?

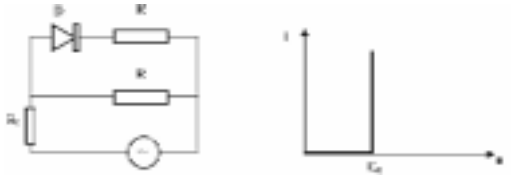
**K. 420.** Feloldanak 250mL vízben 30g kristályos szódát ($Na_2CO_3 \cdot 10H_2O$). Számítsd ki a szóda-oldat tömeg %-os töménységét!

# Fizika

**F. 293.** Az M tömegű, l hosszúságú kis kocsi sima vízszintes felületen áll. A kis kocsin két $m_1$ és $m_2$ tömegű ember áll. Mekkora a kis kocsi elmozdulása, ha a két ember helyet cserél? (A két ember a kocsi két végén helyezkedik el!)

**F. 294.** A lapos drótkeret $\vec{B}$ indukciójú mágneses térben található. A mágneses tér merőleges a keret síkjára. A keret *a* oldalú négyzet. Ezután a keretet: a) 1:2 oldal arányú téglalappá hajlítjuk; b) kihúzzuk egy egyenessé; c) két 1/H területarányú négyzetté formáljuk. Adjátok meg a kereten az egyes alakváltoztatáskor áthaladó töltést. A keret ellenállása *R*.


**F. 295.** Adjuk meg az $R_0$ ellenálláson áthaladó áram erősségét az ábrán látható áramkörben. A dióda feszültség-áramerősség karakterisztikája az ábrán látható. (ideális dióda).



$u=U_m \cos\omega t$

az $U_m$=300V, $U_0$=40V, $R_0$=10kΩ, R=2kΩ .

a feladatokat a *Kvant* folyóirat nyomán közli

**Gaál László**,
tanár, Csíkszereda

# Informatika

**A Nemes Tihamér Számítástechnika Verseny**
**II. fordulójának feladatai (2003)**

III. kategória: 11-13. osztályosok

**1. feladat:** Magánhangzók távolsága *(10 pont)*

Egy magyar szóban lehetnek több karakterrel leírt mássalhangzók is (pl. sz, cs, dzs, …). Feltételezzük, hogy az egymás melletti s+z, … betűket mindig egy hangnak, azaz sz-nek, … értelmezhetjük. A hosszú mássalhangzókat egy hangnak kell venni!

Írj programot (MAGAN.PAS, MAGAN.C,…), amely megadja, hogy egy szóban a magánhangzók hány hangra vannak egymástól!

A MAGAN.BE szöveges állomány egyetlen sorában egy legalább 1 és legfeljebb 255 karakterrel leírt magyar szó van.

A MAGAN.KI szöveges állományba eggyel kevesebb számot kell írni, mint a bemeneti szóban levő magánhangzók száma. Az i-edik szám a szó i-edik és azt követő magánhangzója közötti hangok száma legyen!

*Példa:*

| | MAGAN.BE | MAGAN.KI |
|---|---|---|
| 1. példa: | templomtorony | 3 2 1 |
| 2. példa: | hosszú | 1 |

**2. feladat:** Megrendelés *(12 pont)*

Egy rendezvényt olyan teremben tartanak, ahol ***M*** db ülőhely van. Az ülőhelyek ***1***-től ***M***-ig sorszámozottak. A rendezvény szervezője megrendeléseket fogad. Minden megrendelés egy ***A B*** számpárt tartalmaz, ami azt jelenti, hogy a megrendelő olyan ülőhelyet szeretne kapni, amelynek ***S*** sorszáma ***A*** és ***B*** közé esik ***(A≤S≤B)***.

Írj programot (KIOSZT.PAS, KIOSZT.C,…), amely kiszámítja, hogy a szervező a megrendelések alapján a legjobb esetben hány megrendelést tud kielégíteni és meg is ad egy olyan jegykiosztást, amely kielégíti a megrendeléseket!

A KIOSZT.BE szöveges állomány első sorában két egész szám van, ***M*** és ***N***. ***M*** az ülőhelyek száma ***(1≤M≤1000)***, ***N (1≤N≤1000)*** pedig a megrendelések száma. A következő ***N*** sor mindegyike két egész számot ***A, B (1≤A≤B≤M)*** tartalmaz egy szóközzel elválasztva. Az állomány ***i+1***-edik sorában lévő megrendelés sorszáma ***i***.

A KIOSZT.KI szöveges állomány első sorába a legtöbb kielégíthető megrendelés ***K*** számát kell írni! A további ***K*** sor tartalmazza a jegykiosztást, minden sor két egész számot tartalmazzon egy szóközzel elválasztva. Az első szám egy megrendelés sorszáma, a második pedig azon ülőhely sorszáma, amelyet a megrendelő kap. A kiosztás kiírása tetszőleges sorrendben lehet. Ha több megoldás van, akkor egy tetszőlegeset ki lehet írni.

*Példa:*

| KIOSZT.BE | KIOSZT.KI |
|---|---|
| 10 6 | 4 |
| 3 3 | 5 1 |
| 2 2 | 2 2 |
| 2 3 | 1 3 |
| 1 3 | 6 4 |
| 1 2 | |
| 2 4 | |

**3. feladat:** Lámpák *(16 pont)*

Egy ***NxM***-es téglalap alakú téren ***K*** lámpát helyeztek el. Mindegyiknek ismerjük a helyét. Mindegyik lámpa azt a ***HxH***-s (***H*** páratlan) négyzet alakú területet világítja be, amely átlóinak metszéspontjában áll a lámpa. A világos területek éjszaka is biztonságosak, a sötéteken azonban tanácsosabb nem járni.

Írj programot (LAMPAK.PAS, LAMPAK.C,…), amely megadja, hogy mekkora a téren sötétben maradt terület (a mezők száma), valamint hogy hogyan menjünk át a tér bal felső sarkából a jobb alsó sarkába a legbiztonságosabban úgy, hogy minden pozícióról a 4 oldalszomszédjára léphetünk, átlósan pedig nem léphetünk?

A LAMPAK.BE szöveges állomány első sorában a tér sorai ***N (1≤N≤100)*** és oszlopai ***M*** száma ***(1≤M≤100),*** valamint a lámpák ***K*** száma ***(0≤K≤1000)*** és az általuk bevilágított négyzet oldalhossza (***1≤H≤100***, ***H*** páratlan) van. A következő ***K*** sor mindegyike egy lámpa helyét tartalmazza, egy számpárt egy szóközzel elválasztva: közülük az első egy lámpát tartalmazó mező sorindexe, a második pedig az oszlopindexe. A sorokat felülről-lefelé, az oszlopokat balról-jobbra sorszámozzuk.

A LAMPAK.KI szöveges állomány első sorába a sötétben maradt mezők számát kell írni. A második sorba azon sötét mezők száma kerüljön, ahányon minimálisan át kell menni, ha a tér bal felső sarkából a jobb alsó sarkába szeretnénk eljutni.

*Példa:*

| LAMPAK.BE | LAMPAK.KI |
|---|---|
| 8 10 3 5 | 20 |
| 3 3 | 4 |
| 7 3 | |
| 3 9 | |

**4. feladat:** Képkódolás (18 pont)

Egy ***NxN***-es színes képet (***N*** kettőhatvány) a következőképpen kódolunk:

- Ha a kép egyszínű, akkor a kódja: 0 szín.
- Ha nem egyszínű, akkor bontsuk négy egyforma részre: Ezzel négy kódrészlet áll elő, a kód első jele a fenti 4 számjegy, s ezután a 4 részre alkalmazzuk újra ugyanezt a módszert.

| | |
|---|---|
| 1 | 2 |
| 3 | 4 |

*Példa:*

| | | |
|---|---|---|
| 5666 | kódja: | 1105; 1206; 1306; 1406; 206; 306; |
| 6666 | | 4107; 4207; 4308; 4409 |
| 6677 | | |
| 6689 | | |

Írj programot (KODOL.PAS, KODOL.C,…), amely egy adott képhez kiszámítja a képet megadó kódhalmazt!

A KODOL.BE szöveges állomány első sorában a kép ***N*** mérete (***1≤N≤128***, ***N*** kettőhatvány) van. A következő ***N*** sor mindegyikében pontosan ***N*** jel van, egy-egy képsor képpontjainak a színe. A színt tetszőleges karakter jelöli.

A KODOL.KI állomány első sorába a kép ***N*** méretét (***1≤N≤128***, ***N*** kettőhatvány) és a kódhalmaz ***M*** elemszámát (***1≤M≤1000***) kell írni. A következő ***M*** sor mindegyikébe egy-egy négyzet alakú tartomány kódját kell írni kód szerint lexikografikusan növekvő sorrendben (lásd a példát). A kód nem tartalmazhat semmilyen elválasztójelet.

*Példa:*

| KODOL.BE | KODOL.KI | KODOL.BE | KODOL.KI |
|---|---|---|---|
| 4 | 4 1 | 4 | 4 10 |
| aaaa | 0a | abbb | 110a |
| aaaa | | bbbb | 120b |
| aaaa | | bb77 | 130b |
| aaaa | | bb89 | 140b |
| | | | 20b |
| | | | 30b |
| | | | 4107 |
| | | | 4207 |
| | | | 4308 |
| | | | 4409 |

**5. feladat:** Szavak *(19 pont)*

Adott szóra alkalmazott betű-helyettesítésen azt értjük, hogy a szó minden betűjének helyére egy megadott (legalább egy betűből álló) szót írunk úgy, hogy minden betű minden előfordulását ugyanazon szóval helyettesítjük. Különböző betűk különböző szavakkal helyettesíthetőek. Adott szónak adott betű-helyettesítés melletti képén azt a szót értjük amelyet a helyettesítés elvégzésével kapunk.

Írj programot (SZAVAK.PAS, SZAVAK.C,…), amely kiszámítja, hogy van-e olyan betű-helyettesítés, amely mellett két adott szó képe megegyezik!

A SZAVAK.BE szöveges állomány első két sorában van a két szó, soronként egy-egy. Mindkét szó hossza legfeljebb ***33***. A szavak csak az angol ábécé nagybetűit (A-tól Z-ig ) tartalmazhatják. A két szóban pontosan ugyanazok a betűk fordulnak elő.

A SZAVAK.KI szöveges állomány első sorába egy ***L*** egész számot kell írni! ***L*** értéke ***0*** legyen, ha nincs olyan betű-helyettesítés, amely mellett a két szó egybeesik. Egyébként ***L*** a legrövidebb olyan szó hossza, amire van olyan betű-helyettesítés, hogy a két szó képe megegyezik és hossza ***L***. A további sorokban meg kell adni a betű-helyettesítést, annyi sort kell kiírni, ahány különböző betű szerepelt a két bemeneti szóban. Minden sor első karaktere a helyettesítendő betű legyen, majd egy szóközzel elválasztva álljon az a szó, amelyre a betűt helyettesítjük. A sorokat tetszőleges sorrendben lehet kiírni.

Ha a programod nem a legkisebb ilyen ***L***-et számítja ki, de a helyettesítéssel a két szó egybeesik, akkor fél pontszámot kapsz a megoldásra.

*Példa:*

| SZAVAK.BE | SZAVAK.KI |
|---|---|
| BAACBD | 7 |
| ABDCD | A A |
| | B A |
| | C A |
| | D AA |

# Megoldott feladatok

**Kémia** (Firka 2/2003-2004)
**K. 412.**

**4.** C% = 54,50 H% = 9,09, mivel C% +H% < 100 az **A** vegyület oxigént is tartalmaz, molekulaképlete: $C_xH_yO_z$

$x \cdot 12 / y \cdot 1 = 54,50/9,09$ $x \cdot 12 / z \cdot 16 = 54,50/ (100-54,5-9,09)$

$y = 2x$ $x = 2z$

A vegyülési arányoknak megfelelően a felírt négy képlet közül **A**: $C_2H_4O$

**5.** A feladat adatai szerint $\upsilon_{CH3Cl} = \upsilon_{CHCl3}$

$CH_4 + Cl_2 \rightarrow CH_3Cl + HCl$

$CH_4 + 3Cl_2 \rightarrow CHCl_3 + 3HCl$

$2\upsilon\ CH_4$ ……$4\upsilon Cl_2$ ……$4\upsilon \cdot 71g\ Cl_2$

$1,12m^3/22,4m^3kmol^{-1}$……x kg x = 27,1kg

**6.** $\upsilon_{CH3Cl} : \upsilon_{CH2Cl2} : \upsilon_{CHCl3} : \upsilon_{CH4} = 4 : 2 : 1 : 1$

4mol $CH_3Cl$ keletkezésekor 8mol $CH_4$ –ra van szükség $M_{CH3Cl} = 50,5g/mol$

4.50,5g $CH_3Cl$ ……. 8.22,4L $CH_4$

20,2kg ……. $Vm^3$ $V = 17,92m^3$

**7.** Használjuk a következő jelöléseket az elegy komponenseire:

$CH_3Cl = A$ $C_2H_5Cl = B$

Mivel $\upsilon = m/M$ $50,5\ \upsilon_A + 64,5\ \upsilon_B = 100$

$5,5\ \upsilon_A + 35,5\ \upsilon_B = 59,33$

a két egyenletből $\upsilon_A = 0,57$ mol és $\upsilon_B = 1,12$ mol, ezen értékek alapján az **A** és **B** anyagmennyiségeinek aránya ½.

**8.** $CH_3{-}CH{=}CH_2 + Br_2 \rightarrow CH_3{-}CHBr{-}CH_2Br$

**A** **B**

$\upsilon_A = \upsilon_B$

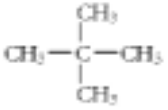
$\upsilon_A = 0,112L/ 22,4L.mol^{-1} = 5.10^{-3}mol$

1000mL **B**old. …. 0,2mol **B**

V …..$5.10^{-3}$mol innen V =25mL

**9. A**: $C_nH_{2n+2}$ $M_A = 72$ akkor $14n +2 = 72$ ahonnan n = 5

A feladat kikötése alapján az **A** az a pentán izomer, amelyben a H atomok egyenértékűek, mert csak ebben az esetben képezhet egyetlen monobrómozott terméket. Ezt a feltételt csak a 2,2-dimetilpropán elégíti ki, amelynek a szerkezete:



**11.** A termokémiai egyenlet értelmében 1mol $CH_3OH$ elégetésekor 674,8kJ hő szabadul fel, akkor 1kmol esetében ennek az ezerszerese. Az a.) válasz a jó.

**12.** A sztöchiometrikus egyenlet együtthatói oxidációsszám változás alapján könnyen kiszámíthatók:

$3CH_3CH_2CH{=}CH_2 + 4K_2Cr_2O_7 + 16H_2SO_4 \rightarrow 6CH_3CH_2COOH +$

**A** **B** $4Cr_2(SO_4)_3 + 4K_2SO_4 + 16H_2$

3mol**A** …..4mol **B** 1L old…..0,1mol
0,1mol……x x= 0,4/3mol V ……. 0,4/3
ahonnan V =1,33L

**13.** $C_nH_{2n+1}COOH + CH_3OH \rightarrow CH_3– O –\underset{\underset{O}{\|}}{C}– C_nH_{2n+1}$

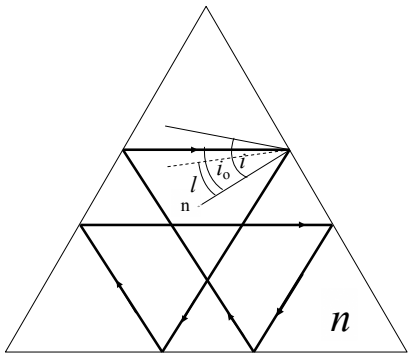
**A** **B**

$M_B = 88$ $M_A = M_B - 14$ , akkor $M_A = 74$
$14n + 46 = 74$ ahonnan $n = 2$ tehát **A**: $CH_3CH_2COOH$ (propán-sav)

**Fizika** (Firka 6/2002-2003)

**F. 290.** A fénysugár akkor nem lép ki, ha az oldallap elérésénél bekövetkezik a teljes visszaverődés, vagyis $i_0 > l$ , ahol $l$ a határszög. Ezért: $\sin i_0 > \sin l$ , de mivel $i_0 = 30^0$ és $\sin l = 1/n \Rightarrow \sin 30^0 > 1/n \Rightarrow n > 1/(1/2) \Rightarrow n > 2$
Tehát ha $n > 2$ , a fénysugár nem tud kiszabadulni a prizma belsejéből!

– Az ábra alapján azonnal belátható, hogy a fény által megtett út $s = a + b + c = 3a$ , ahol $a$ a prizma oldallapjának a hossza.
– Észrevesszük, hogy ha a sugár beesési szögét változtatjuk, például csökkentjük, akkor a következő oldallapon történő visszaverődésnél éppen ennyivel fog növekedni, majd a rákövetkezőn ugyanennyivel csökken, és így tovább. Mivel a fénysugár kilépése csak a beesési szög csökkentésénél történhet meg, csak arra kell ügyeljünk, hogy $i > l$ maradjon, vagyis legyen $|\Delta i| < (i_0 - l)$ .



a

– Ebből $\sin|\Delta i| < \sin(i_0 - l)$ következik. Felbontva: $\sin|\Delta i| < (\sin i_0 \cos l - \cos i_0 \sin l)$ , majd beírva a $\sin i_0 = 1/2$ , $\cos i_0 = \sqrt{3}/2$ valamint a $\sin l = 1/n$ , $\cos l = \sqrt{n^2 - 1}/n$ értékeit: $\sin|\Delta i| < \left(\sqrt{n^2 - 1} - \sqrt{3}\right)/(2n)$ .

* Gyémántprizma esetén a törésmutató értéke $n$=2,42. Behelyettesítve: $\sin|\Delta i| < 0{,}097$ , amiből a $|\Delta i| < 5{,}59^0$ adódik.

**Informatika**

*A Nemes Tihamér Számítástechnika Verseny II. fordulójának feladatai (2003),*
*II. kategória: 9-10. osztályosok*

**1. feladat:** Mássalhangzók
Áfra Attila Tamás megoldása,
Szatmárnémeti, Kölcsey Ferenc Főgimnázium, 10. oszt., 4. helyezett

```
{************************************
Nemes Tihamer - 2003, II. kat. 1. feladat
```

```
MASSAL.PAS
INPUT: MASSAL.BE
OUTPUT: MASSAL.KI
*************************************}
program MASSAL;

type
  TCharSet = set of char;

const
  MySet: TCharSet = ['b', 'c', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'm',
                     'n', 'p', 'q', 'r', 's', 't', 'v', 'w', 'x', 'z'];

var
  s: string;
  lenS, c: byte;
  r: array [1..256] of byte;

{ Read input from file }
procedure Input(FileName: string);
var
  f: text;
begin
  Assign(f, FileName);
  Reset(f);
  Read(f, s);
  Close(f);
  LenS := Length(s);
end;

{ Perform counting }
procedure DoIt;
var
  pos, i: byte;
begin
  c := 0; pos := 1; i := 1;
  while true do
   begin
    while not (s[i] in MySet) do
    begin
      inc(i);
      if i > lenS then Exit;
    end;
    inc(c);
    r[c] := 0;
    while (s[i] in MySet) do
    begin
      inc(r[c]);
      inc(i);
      if i > lenS then Exit;
    end;
   end;
end;

{ Write output to file }
procedure Output(FileName: string);
var
  f: text;
  i: byte;
begin
  Assign(f, FileName);
  Rewrite(f);
  for i := 1 to c do
   begin
    Write(f, r[i]);
    Write(f, ' ');
   end;

  Close(f);
end;
```

```
{ -- Main -- }
begin
  Input('massal.be');
  DoIt;
  Output('massal.ki');
end.
```

**2. feladat:** Képkódolás
Daday Csaba megoldása,
Nagyvárad, Ady Endre Elméleti Líceum, 9. oszt., 5. helyezett

```
{************************************
Nemes Tihamer - 2003, II. kat. 2. feladat
DEKODOL.PAS
INPUT: DEKODOL.BE
OUTPUT: DEKODOL.KI
************************************}
program DEKOD;
var
  tki, tbe: text; eppm: string; a, k: integer;
  kep: array[1..128, 1..128] of char;
  cv: integer;

function kettohatv(kit: integer): longint;
begin
  if kit = 0 then kettohatv := 1 else kettohatv := 2*kettohatv(kit-1)
end;

procedure kiir;
var s, p: byte;
begin
  writeln(tki, a);
   for p := 1 to a do
    begin
     for s:=1 to a do
      write(tki, kep[s, p]);
     writeln(tki)
    end;
  close(tki)
end;

procedure megtolt(betu: char);
var q, w: byte;
begin
  for q := 1 to a do
   for w := 1 to a do
    kep[q, w] := betu;
  kiir
end;

procedure szinez(kod: string);
var
  cv1, cv2, c: byte; str: string; q: 1..4; hiba: integer;
  fs: record
   x, y: byte
  end;
begin
  c := length(kod);
  if length(kod) = 2 then
   begin
    megtolt(kod[2]);
    halt
   end;
  fs.x := 1;
  fs.y := 1;
  str := copy(kod, 1, length(kod)-2);
  for cv1 := 1 to length(str) do
   begin
    val(str[cv1], q, hiba);
```

```
    inc(fs.x, a div kettohatv(cv1)*((q-1) mod 2));
    inc(fs.y, a div kettohatv(cv1)*((q-1) div 2));
   end;
  for cv1 := fs.x to fs.x + a div kettohatv(length(str))-1 do
   for cv2 := fs.y to fs.y + a div kettohatv(length(str))-1 do
    kep[cv1, cv2] := kod[length(kod)];
end;

begin
  assign(tki, 'dekodol.ki');
  rewrite(tki);
  assign(tbe, 'dekodol.be');
  reset(tbe);
  readln(tbe, a, k);
  for cv := 1 to k do
   begin
    readln(tbe, eppm);
    szinez(eppm);
   end;
  kiir
end.
```

**3. feladat:** Harmadolás
Szilágyi Péter megoldása,
Kolozsvár, Báthory István Elméleti Líceum, 10. oszt., 2. helyezett

```
/*************************************
Nemes Tihamer - 2003, II. kat. 3. feladat
HARMAD.CPP
INPUT: HARMAD.BE
OUTPUT: HARMAD.KI
*************************************/
#include <stdio.h>

enum bool {false, true};

struct munka
{
  int kie;
  int osztsz;
  int kinek1, kinek2;
};

int feloszt;
munka m[1000];
int vallalkozo;
int sok, keves, vegez;
int sokt[3], kevest[1000], vegezt[1000];
int max = 0;
int min = 0;

void oszt(int ki)
{
  int i;
  for (i = 0; i < feloszt; i++)
  {
    if (m[i].kinek1 > vallalkozo)
      vallalkozo = m[i].kinek1;
    if (m[i].kinek2 > vallalkozo)
      vallalkozo = m[i].kinek2;
    if (m[i].kie == m[ki].kinek1 || m[i].kie == m[ki].kinek2)
      {
        m[i].osztsz = m[ki].osztsz+1;
        oszt(i);
      }
    if (m[i].osztsz > max)
      max = m[i].osztsz;
  }
}
```

```
void rendez()
{
  int i, j;
  int poz=0;
  int temp;
  for (i = 0; i < feloszt; i++)
    if (m[i].osztsz == max)
    {
      kevest[poz++] = m[i].kie;
      kevest[poz++] = m[i].kinek1;
      kevest[poz++] = m[i].kinek2;
    }
  for (i = 0; i < poz; i++)
    for(j = i+1; j < poz; j++)
      if (kevest[i] > kevest[j])
      {
        temp = kevest[i];
        kevest[i] = kevest[j];
        kevest[j] = temp;
      }
  keves = poz;

  bool talalt1 = false, talalt2 = false;
  for (i = 0; i < feloszt; i++)
    {
      if (m[i].kie == m[0].kinek1)
        talalt1 = true;
      if (m[i].kie == m[0].kinek2)
        talalt2 = true;
    }
  poz = 0;
  sokt[poz++] = 1;
  if (talalt1 == false)
    sokt[poz++] = m[0].kinek1;
  if (talalt2 == false)
    sokt[poz++] = m[0].kinek2;
  sok = poz;
  poz = 0;
  bool van;
  for (i = 1; i <= vallalkozo; i++)
    {
      van = false;
      for (j = 0; j < feloszt; j++)
        if (m[j].kie == i)
          van = true;
        if (van == false)
          vegezt[poz++]=i;
    }
  vegez = poz;
}

void main()
{
  FILE *be, *ki;
  int i;
  be = fopen("Harmad.be", "r+t");
  ki = fopen("Harmad.ki", "w+t");
  fscanf(be, "%d\n", &feloszt);
  max = 0;
  for (i = 0; i < feloszt; i++)
    fscanf(be, "%d %d %d\n", &m[i].kie, &m[i].kinek1, &m[i].kinek2);
  if (feloszt == 0)
  {
    keves = 1;
    kevest[0] = 1;
    sok = 1;
    sokt[0] = 1;
    vegez = 1;
    vegezt[0] = 1;
  }
```

```
  else
  {
    m[0].osztsz = 1;
    oszt(0);
    rendez();
  }
  fprintf(ki, "%d ", keves);
  for (i = 0; i < keves-1; i++)
    fprintf(ki, "%d ", kevest[i]);
  fprintf(ki, "%d\n", kevest[keves-1]);
  fprintf(ki, "%d ", sok);
  for (i = 0; i < sok-1; i++)
    fprintf(ki, "%d ", sokt[i]);
  fprintf(ki, "%d\n", sokt[sok-1]);
  fprintf(ki, "%d ", vegez);
  for (i = 0; i < vegez-1; i++)
    fprintf(ki, "%d ", vegezt[i]);
  fprintf(ki, "%d\n", vegezt[vegez-1]);
  fclose(be);
  fclose(ki);
}
```

**4. feladat:** Konténer rendezés
Korodi-Gál Andor Csaba megoldása,
Marosvásárhely, Bolyai Farkas Elméleti Líceum, 10. oszt., 1. helyezett

```
/*************************************
Nemes Tihamer - 2003, II. kat. 4. feladat
KONTENER.CPP
INPUT: KONTENER.BE
OUTPUT: KONTENER.KI
*************************************/
#include<fstream.h>
a[10001], b[5];

void main()
{
  int n, db = 0;
  ifstream f("kontener.be");
  f >> n;
  for(int i = 1; i <= n; i++)
    {
      f >> a[i];
      b[a[i]]++;
    }
  f.close();
  for(i = 1; i <= n; i++)
    {
      int k = 1;
      for(int j = 1; j < a[i]; j++)
        k += b[j];
      int v = n;
      for(j = a[i]+1; j <= 4; j++)
        v -= b[j];
      if(i < k || i > v) db++;
    }
  ofstream g("kontener.ki");
  if(db) db++;
  g << db;
  g.close();
}
```

**5. feladat:** Verem
Árvay Lóránd megoldása,
Máramarossziget, Dragoş Vodă Líceum, 10. oszt., 3. helyezett

```
{*************************************
Nemes Tihamer - 2003, II. kat. 5. feladat
```

```
VEREM.PAS
INPUT: VEREM.BE
OUTPUT: VEREM.KI
*************************************}
program VEREM;
var
  f, g: text;
  a: array[1..1000] of integer;
  i, n, poz, k: integer;
  cont: boolean;
begin
  assign(f, 'VEREM.BE'); reset(f);
  assign(g, 'VEREM.KI'); rewrite(g);
  readln(f, n);
  for i := 1 to n do
    read(f, a[i]);
  for i := 1 to n do
    if a[i] = 1 then poz := i;
  k := 1;
  cont := true;
  while cont do
   begin
    cont := false;
    if a[poz-1] = k+1 then begin k := k+1; poz := poz-1; cont := true; end;
    for i := poz+1 to n do
     if a[i] = k+1 then begin k := k+1; cont := true; poz := i; break; end;
   end;
  writeln(g, k);
  close(f); close(g);
end.
```


A Magyar Tudományos Akadémia lapjában, a Magyar Tudományban rendszeresen közölnek a tudományos világ újdonságaiból. Ezeket olvasva gyűjtöttem egy pár olyan információt, melyek a természettudományok különböző területe után érdeklődőknek csemegeként szolgálhatnak.

**Gyémánt tranzisztorok**

A tiszta gyémántról már a VII. osztályos tanuló is megtanulja kémiaórán, hogy szigetelő anyag. Nagy keménysége miatt ipari célokra mesterségesen is gyártják. Az így előállított gyémánt egymáshoz képest rendezetlenül elhelyezkedő kristályszemcsékből áll. Sikerült előállítani filmrétegben gyémántot, amiről bebizonyosodott, hogy a szilíciuméhoz hasonló elektromos vezetési tulajdonságai vannak. Bór vagy nitrogén szennyező atomok kismennyiségű jelenlétében félvezetőként viselkedik. Metánból származó szénhez bórvegyületet keverve gőzfázisú epitaxiális rétegleválasztással nyertek olyan félvezető gyémántot, amelyből készített áramkörök sokoldalúbban használhatók, mint a szilícium chipek. Pl. a Si-alapú chipek 150 $^{o}$C hőmérséklet felett felmondják a szolgálatot, a gyémánt chipek több száz fokos hőmérsékleten is működnek.