

Dr. Kiss Attila  
[kiss@inf.elte.hu](mailto:kiss@inf.elte.hu)

# Bevezetés az informatikába

# Mi az informatika?

- Az **informatika** (*information technology, IT* vagy *computer science*) önálló tudományág, amely az **adatok** rögzítésével, kezelésével, rendszerezésével, továbbításával foglalkozik.
- Ezt a tevékenységét főként számítógépeken végzi.



# Az informatika típusai

- **elméleti** : módszereket, modelleket, formalizmusokat dolgoz ki a számítógépek készítéséhez és működtetéséhez;
- **mérnöki** : számítógépeket készít, illetve azokhoz elektronikai eszközöket alkot ([hardver](#));
- **rendszertervezői, programozói** : a számítógépek működtető eszközeit hozza létre, illetve azokat működteti ([szoftver](#));
- **alkalmazói** : a számítógépet különböző feladatok elvégzésére használja, például: orvosi alkalmazások, kereskedelmi rendszerek, [CAD](#), nyilvántartások stb.

# Informatika a mindennapokban

- Telefon, Mobil
- TV, háztartási gépek, kütyük
- Bankautomaták
- Internet, keresők, webszolgáltatások
- Közösségi oldalak, Facebook, Twitter, LinkedIn
- Irodai alkalmazások, szövegszerkesztők
- Email
- ...



# Mit csinál egy informatikus?

- Az [informatikus](#) az, aki e fenti területek valamelyikében szerzett képesítésével számítógépeket, vagy számítógépeken alkot, fejleszt, kutat, vagy azok eredményeit használja.



# Milyen informatikusra van szükség?

## ■ Álláshirdetések (Jobinfo.hu, 2015.09.22.)

The screenshot shows the Jobinfo.hu website interface. At the top, there is a navigation bar with icons for home, information, email, and user login/registration. The main navigation menu includes categories like 'Mérnök állások', 'Pénzügyi állások', 'IT állások', and 'Értékesítő állások'. Below this, there are tabs for 'Állások városok szerint', 'Állás kategóriák', and 'Állások nyelv szerint'. The 'Állás kategóriák' tab is active, displaying a list of job categories with checkboxes and the number of available positions. A modal window is open, prompting the user to select a sub-category within 'Informatika / telekommunikáció'. The modal lists various sub-categories such as 'Fejlesztés / programozás', 'Mérnök informatikus', 'Grafikus / webdesigner / webszerkesztő', 'Rendszergazda', 'Informatikai értékesítő / IT sales', 'Telekommunikáció / távközlés', 'IT / telekommunikáció projektvezető', 'Tesztelés', and 'IT Support / helpdesk'. A note in the modal states: '\*Kérjük maximum 5 elemet válasszon egyszerre!'. On the right side of the modal, there is a vertical button labeled 'HÍRLEVÉL FELIRATKOZÁS'.

Adminisztráció / Közigazgatás / Egyéb állások  
Bank / Biztosítás / Közigazgatás / Egyéb állások  
Beszerzés / Logisztika / Számvitel / Számtervezés / Egyéb állások  
Egészségügy / Gyógyászat / Szakmai tevékenység / Egyéb állások  
Egyéb szolgáltatások / Egyéb állások  
Értékesítés / Kereskedelem / Egyéb állások  
Gazdaság / Pénzügy (1619 állás) >>  
HR / Munkaügy (706 állás) >>  
Informatika / telekommunikáció (3476 állás) >>  
Ipari és agrár termelés / technikus (2118 állás) >>

Válasszon a Informatika / telekommunikáció kategórián belüli szakterületek közül!

- Fejlesztés / programozás (1791 állás)
- Grafikus / webdesigner / webszerkesztő (149 állás)
- Informatikai értékesítő / IT sales (20 állás)
- IT / telekommunikáció projektvezető (225 állás)
- IT Support / helpdesk (677 állás)
- Mérnök informatikus (312 állás)
- Rendszergazda (574 állás)
- Telekommunikáció / távközlés (176 állás)
- Tesztelés (141 állás)
- Vezetői pozíciók - Informatika / telekommunikáció (39 állás)

\*Kérjük maximum 5 elemet válasszon egyszerre!

Értékesítés / Kereskedelem / Egyéb állások >>  
Iparmunka / Alkalmi munka / Diákmunka (661 állás) >>  
Vendéglátás / Idegenforgalom (902 állás) >>  
Vezetői állások (2297 állás) >>

# Mihez értsen egy informatikus?

## Szoftverfejlesztő (PL/SQL)

Pénzügyi szektorban működő partnerünk IT üzemeltetési csapatába keresünk új kollégát Szoftverfejlesztő pozíció betöltésére.

### Főbb feladatok:

- Oracle alapú fejlesztések kivitelezése PL/SQL nyelven.
- Adatbázis alapú üzleti logika fejlesztése a vállalati web alapú keretrendszer működése mögött.
- A csapat tagjaként segíteni a többi fejlesztő munkáját.

### Elvárások:

- PL/SQL, Oracle 10g/11g programozási tapasztalat. (tárolt eljárások, packagek, triggerek, nézetek fejlesztése)
- Rendszertervezésben szerzett tapasztalat, adatbázis normalizálás.
- Folyamat- és rendszerszemléletű gondolkodásmód.
- Minimum 1 év hasonló munkakörben szerzett tapasztalat.
- Szakmai angol szöveg megértése, magabiztos angol nyelvtudás főleg írásban.
- Minőségorientált hozzáállás, precíz, hatékony munkavégzés.
- Problémamegoldó és jó kommunikációs, kapcsolatteremtő készség, kreativitás.
- A feladattal, illetve a munkával kapcsolatban érzett szenvedély és igényesség.
- Új technológiák magtanulására való nyitottság és hajlandóság.

### Előny:

- Web-fejlesztésben szerzett tapasztalat. (PHP/Javascript keretrendszerek használata során szerzett tapasztalat)
- Verziókezelő rendszer adminisztrációs tapasztalat. (SVN)
- Oracle DBA tapasztalat.
- SQL optimalizálásban szerzett tapasztalat.
- BASH programozási ismeret.
- Linux rendszergazdai tapasztalat.

## 3D Programozó

munkatársakat keresünk főállásba

### ELVÁRÁSOK:

- Kimagasló C++ programozói tudás
- Angol nyelvismeret (minimum középfokon)
- Felsőfokú informatikai / mérnöki végzettség vagy hallgatói jogviszony
- Kiváló matematikai készség

### ELŐNY:

- 3D programozási tapasztalat (OpenGL/DirectX)
- iPhone/MacOSX vagy Android fejlesztői tapasztalat
- OpenCL/GPGPU tapasztalat

### PHP programozó

munkatársat keres szoftverfejlesztő csapatába

### Feladatok

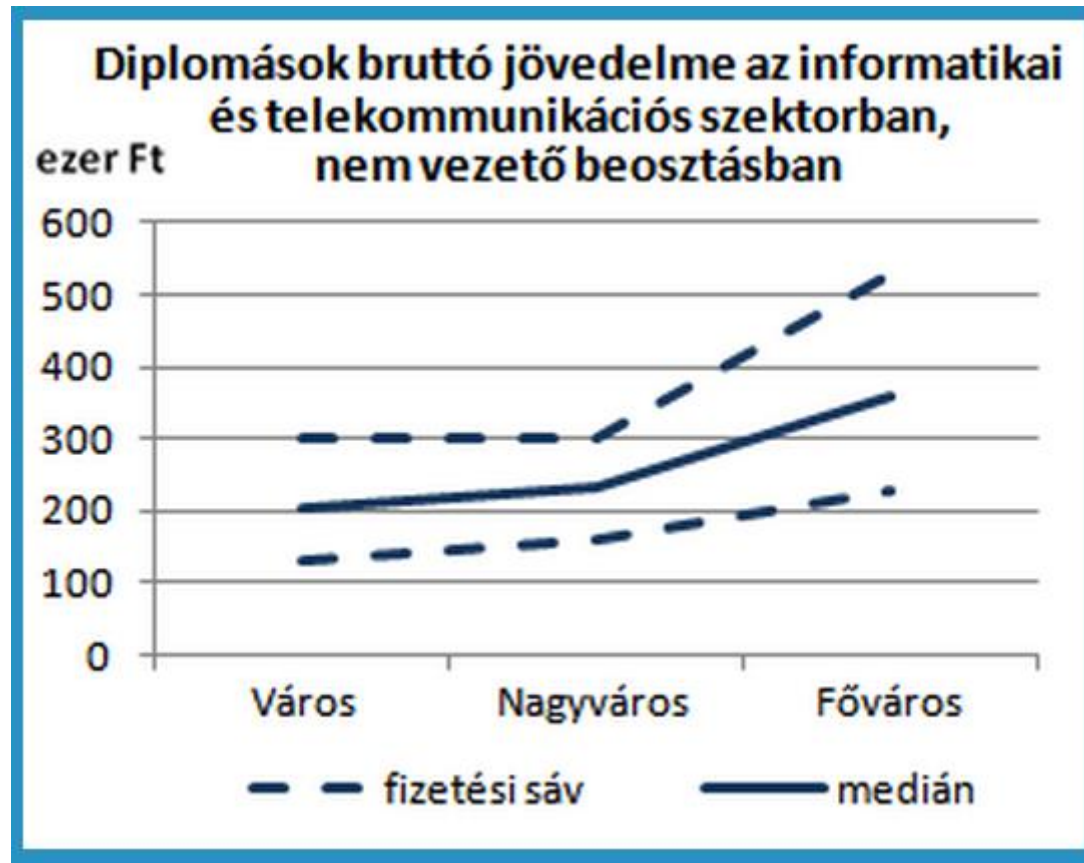
#### A cégcsoport szoftverfejlesztő csoport tagjaként feladata

- A cégcsoport által fejlesztett szoftver fejlesztése, karbantartása;
- Web alapú alkalmazások fejlesztése, karbantartása;
- Az elvégzett munka tesztelése, dokumentálása;
- A szoftverrel kapcsolatos support tevékenység.

### Elvárások

- PHP nyelv alapos ismerete;
- Relációs adatbázis alapos ismerete;
- Precizitás mind a határidők, mind a prioritások kezelésében;
- Proaktív, felelősségteljes munkavégzés csapatban és egyénileg egyaránt;

# Mennyit keres egy informatikus?



Magyarország, 2015.09.22., 1 Euro = 310 Hungarian forint



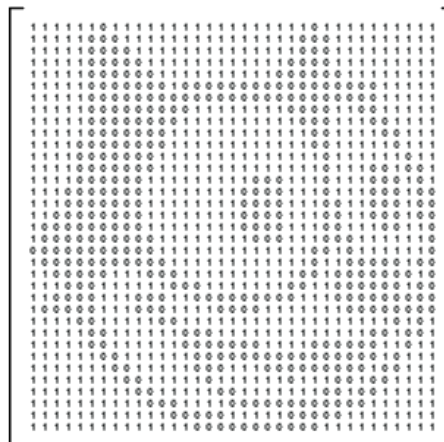
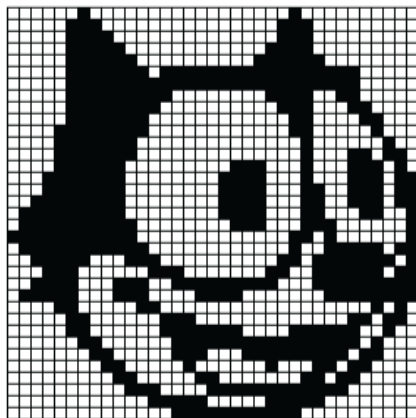
# IT keresetek összehasonlítása Németországban és Ausztriában

Position	Germany		Austria	
	Experience			
	3-5 years	6-9 years	3-5 years	More than 5 years
CEO	86.000 – 106.000	103.000 – 142.000	85.000 – 106.000	103.000 – 142.000
Head of infrastructure	62.000 – 75.000	73.000 – 87.000	62.000 – 75.000	73.000 – 87.000
IT Project Manager	61.500 – 76.000	74.000 – 92.000	61.500 – 76.000	74.000 – 92.000
Head of Development	66.500 – 83.000	78.500 – 99.500	66.500 – 83.000	78.500 – 99.500
IT Manager	72.000 – 92.000	83.000 – 116.000	72.000 – 92.000	83.000 – 116.000
IT Project Manager	59.000 – 69.000	67.000 – 82.000	62.000 – 72.000	> 72.000
Database Developer	47.000 – 58.000	60.000 – 74.500	45.000 – 56.000	56.000 – 63.000
Database Administrator	45.500 – 54.500	54.000 – 70.000	45.500 – 56.000	56.000 – 63.000
Web Developer	44.000 – 52.500	51.500 – 62.500	44.000 – 53.000	> 53.000
Mobile App Developer	46.000 – 55.500	53.500 – 64.500	46.000 – 55.500	53.500 – 64.500
VoiP Specialist	40.000 – 50.500	48.500 – 62.500	40.000 – 50.500	48.500 – 62.500
Network & Security Administrator	42.000 – 52.500	50.500 – 60.500	42.000 – 49.000	> 50.000

Source: Robert Half

# Az informatika jellemzői: a tudás reprezentálása

- A tudás reprezentálása adatok formájában
  - 101010111111010101010101111111
  - 2015.09.24.
  - 3.1415
  - Selye János



35x35

# Az informatika jellemzői: algoritmikus problémamegoldás

- Algoritmikus gondolkodás, algoritmikus problémamegoldás



A paprikás krumpli készítésének algoritmus

Ha nincs meg otthon minden hozzávaló, **akkor** menjünk el vásárolni,  
**egyébként** készítsük elő a hozzávalókat!

Tisztítsuk meg és vágjuk fel a hagymát és a krumplit!

Egy elég nagy edénybe tegyünk olajat és tegyük bele a felvágott  
hagymát!

Gyújtsuk meg a gázt!

Tegyük fel az edényt!

**Ismételjük, amíg nem pirul meg a hagyma**

kavarjuk meg és várjunk fél percet!

Tegyük bele a piros paprikát!

Kavarjuk össze!

Tegyük bele a krumplit és a fűszereket!

Öntsük fel vízzel!

**Ismételjük, amíg nem főtt meg a krumpli**

Kavarjuk meg! Várjunk 2 percet!

Tegyük bele a kolbászt (virslit)!

**Ismételjük, amíg nem főtt meg a kolbász (virslit)**

Kavarjuk meg! Várjunk egy percet!

Zárjuk el a gázt!

Tálaljunk

# Algoritmusok

- **Az algoritmus fogalma:**

Az algoritmus egy bizonyos feladattípus megoldására szolgáló lépések (utasítások, előírások) véges sorozata, amely alapján a feladat véges lépésben megoldható.
- **Az algoritmusokkal szemben támasztott követelmények:**
  1. **Végesség:** a feladat megoldására szolgáló lépések számának és minden egyes lépésnek is végesnek kell lennie
  2. **Meghatározottság:** Az algoritmus minden lépésének pontosan definiálnak, egyértelműnek, félreérthetetlennek kell lennie
  3. **Elvégezhetőség:** Az algoritmus minden lépésének elvégezhetőnek kell lennie
- **Néhány algoritmus leíró módszer:**

Mondatszerű, folyamatábra, struktogram, struktúra diagram

# Mondatszerű leírás

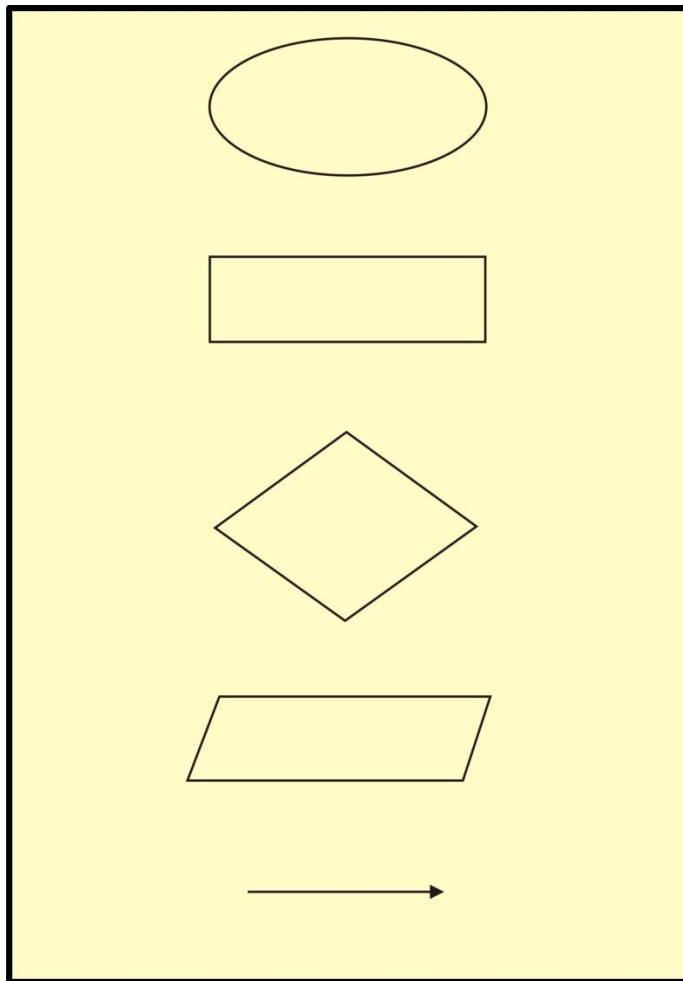
- Az algoritmusok megfogalmazásának első és legtermészetesebb módja a természetes emberi beszéd.
- Ennél a módszernél a lépések (utasítások, előírások) megfogalmazására betűket, szavakat, mondatokat használunk.
- Nagy hátránya, hogy nem egyértelmű. Ezért szükség van ennél pontosabb, formális leírásra.

Például:

1. Elmegyek moziba
2. Veszek pattogatott kukoricát
3. Ha marad még pénzem, veszek üdítőt
5. Megnézem a filmet
6. Hazamegyek a moziból



# A folyamatábra jelei



A tevékenységeknek síkidomokat feleltetünk meg

- *Ellipszis*: a folyamatábra indulási és befejezési pontja
- *Téglalap*: elemi tevékenységek
- *Rombusz*: elágazás, választás
- *Paralelogramma*: input és output tevékenységek
- *Nyilak*: jelzik a haladás irányát

# Az algoritmusok építőelemei

- Minden algoritmus 3 alapvető szerkezeti elemből építhető fel:
  1. *Szekvencia*: egymás után végrehajtandó tevékenységek sorozata
  2. *Szelekció (választás, elágazás)*: lépések, tevékenységek közötti választás
  3. *Iteráció (ismétlés, ciklus)*: valamely tevékenység sorozat ismételt végrehajtása

# Egy napunk

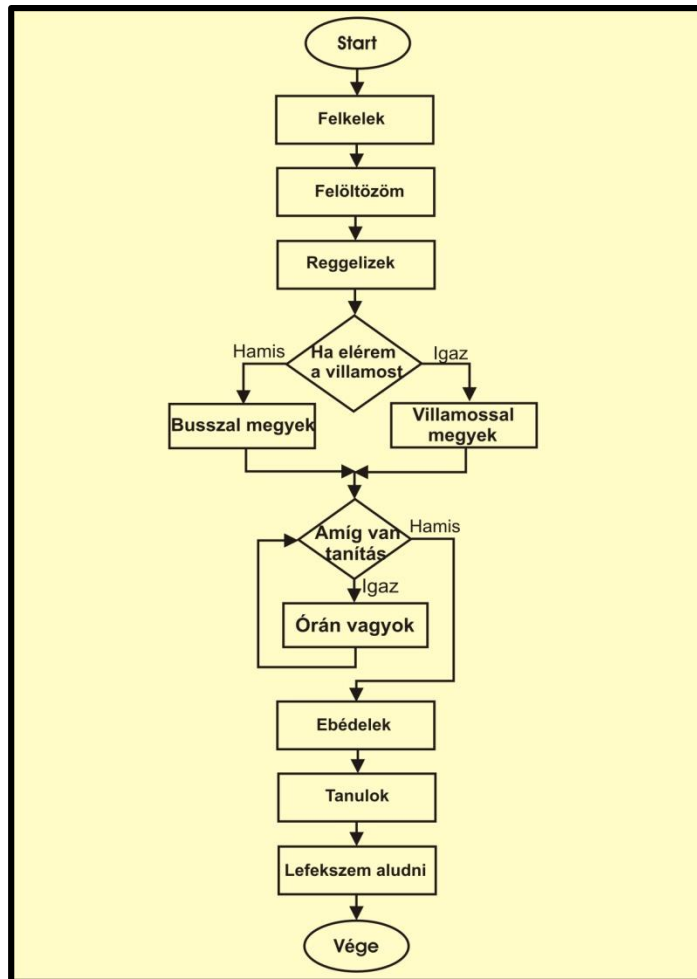
1. Felkelek
2. Felöltözöm
3. Reggelizek
4. Ha elérem a villamost  
Villamossal megyek  
Ha nem érem el  
Busszal megyek
5. Amíg van tanítás  
Órán vagyok
6. Ebédelek
7. Tanulok
8. Lefekszem aludni

## Egy napunk algoritmus mondatszerűen leírva

- Szekvencia
- Szelekció
- Iteráció
- Szekvencia



# Egy napunk



Egy napunk algoritmus  
folyamatábrával leírva

- Szekvencia
- Szelekció
- Iteráció
- Szekvencia

# Mi az információ?

- Az **információ** latin eredetű szó, amely értesülést, hírt, üzenetet, tájékoztatást jelent.
- Általánosságban információnak azt az [adatot](#), [hírt](#) tekintjük amely számunkra releváns és ismerethiányt, bizonytalanságot csökkent.



# Mit kezdünk az információval?

- Gyűjtjük
- Reprezentáljuk (kódoljuk)
- Tömörítjük, transzformáljuk, titkosítjuk
- Tároljuk
- Betöltjük
- Visszakeressük
- Feldolgozzuk, összesítjük, elemezzük
- Megjelenítjük
- Továbbítjuk
- Elmentjük, archiváljuk

# Honnan lehet információt gyűjteni?

- Nyomtatványok, kérdőívek (manuális adatrögzítés, szkennelés, karakterfelismerés)
- Internet, weboldalak, közösségi oldalak (automatikus gyűjtés programokkal)

**1453-02** RENDELKEZŐ NYILATKOZATOK AZ ADÓRÓL II.

Adózó neve: **Egység Endre**  
adóazonosító jele: [ ]  
adószáma: [ ]

**I. RENDELKEZÉS AZ ÖNKÉNTES KÖLCSÓNÓS BIZTOSÍTÓ PÉNZTÁRHOZ UTALANDÓ ÖSSZEGRŐL**

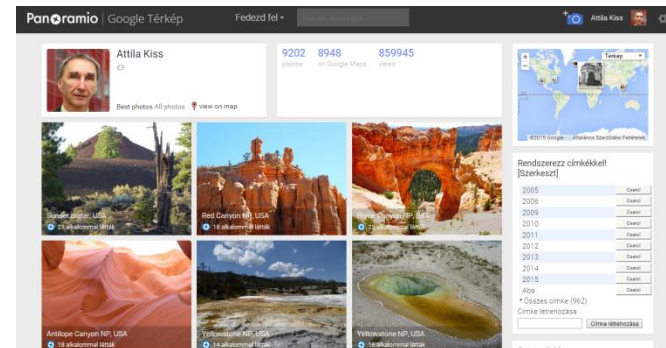
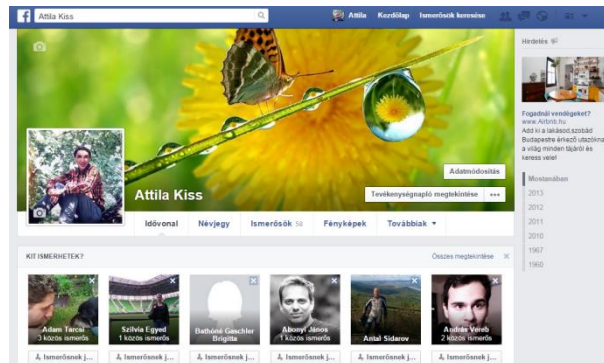
a	b
131. Az önkéntes kölcsönös nyugdíjpénztárba befizetett összeg 20%-a	
132. Az önkéntes kölcsönös egészségpénztárba és oncssegélyező pénztárba befizetett összeg 20%-a	
133. Önkéntes kölcsönös egészségpénztár, oncssegélyező pénztárban lekötött összeg 10%-a	
134. Önkéntes kölcsönös egészségpénztár, oncssegélyező pénztár prevenció szogaltatásának 10%-a	
135. Önkéntes kölcsönös biztosítópénztári kedvezmények együttes összege, legfeljebb 150 000 Ft	
136. Az összevont adóalap utáni adó terhére utalandó összeg, legfeljebb 150 000 Ft	

(A) Töltse ki a pénztár adószámát. A nevét és a számlaszámát a hivatalos értesítéstől a program írja be.


**NYILATKOZAT OTP Országos Egységpénztár**

137. A 136. sor "b" oszlopában lévő összeget kérem a(z) **OTP Országos Egységpénztár** elnevezésű: [ 1 8 1 0 5 5 6 4 ] - [ 1 ] - [ 4 ] adószámú önkéntes pénztár [ 1 1 7 0 3 0 0 6 ] - [ 2 0 4 1 1 4 4 0 ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] bankszámlaszámára utalni.

138. Jelölje X-szel, ha Ön a nyilatkozat megtetelekor nyugdíjszolgáltatás miatt nem tagja az önkéntes kölcsönös biztosító pénztárnak!

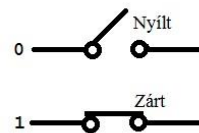
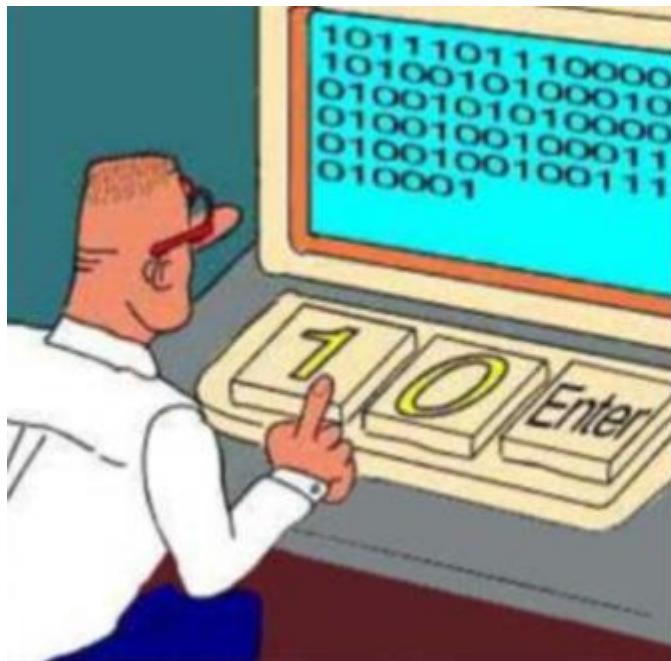


# Hogyan reprezentáljuk az információt?

- A számítástechnikában adatnak (angol nyelven *data*) nevezzük a számokkal leírható dolgokat, melyek számítástechnikai eszközökkel rögzíthetők, feldolgozhatóak, és megjeleníthetők.
- 
- Az adatoknak önmagukban nincs jelentésük. (Például annak az adatnak, hogy 2015 nincs önmagában jelentése.) Az adatok az értelmezéstől, azok feldolgozásának módjától, alkalmazásuktól nyernek értelmet, és válhatnak információvá, *hasznos* adatokká.

# Bináris (kettes) számrendszer

- Két állapot – fizikai eszközzel könnyen megvalósítható



decimális	bináris	hexadecimális
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

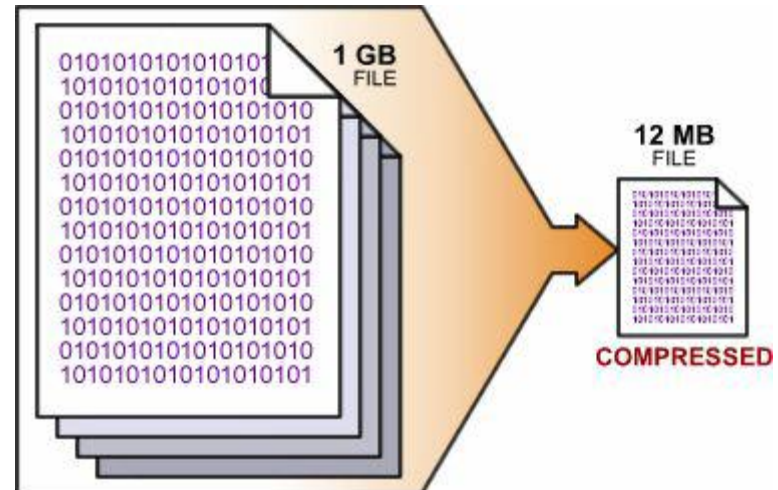
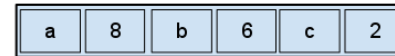
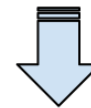
Az első 16 természetes szám írása a különböző számrendszerekben

# Tömörítés

- Tömörítés  
(például az egyforma jelek helyett: egy jel és egy szám)

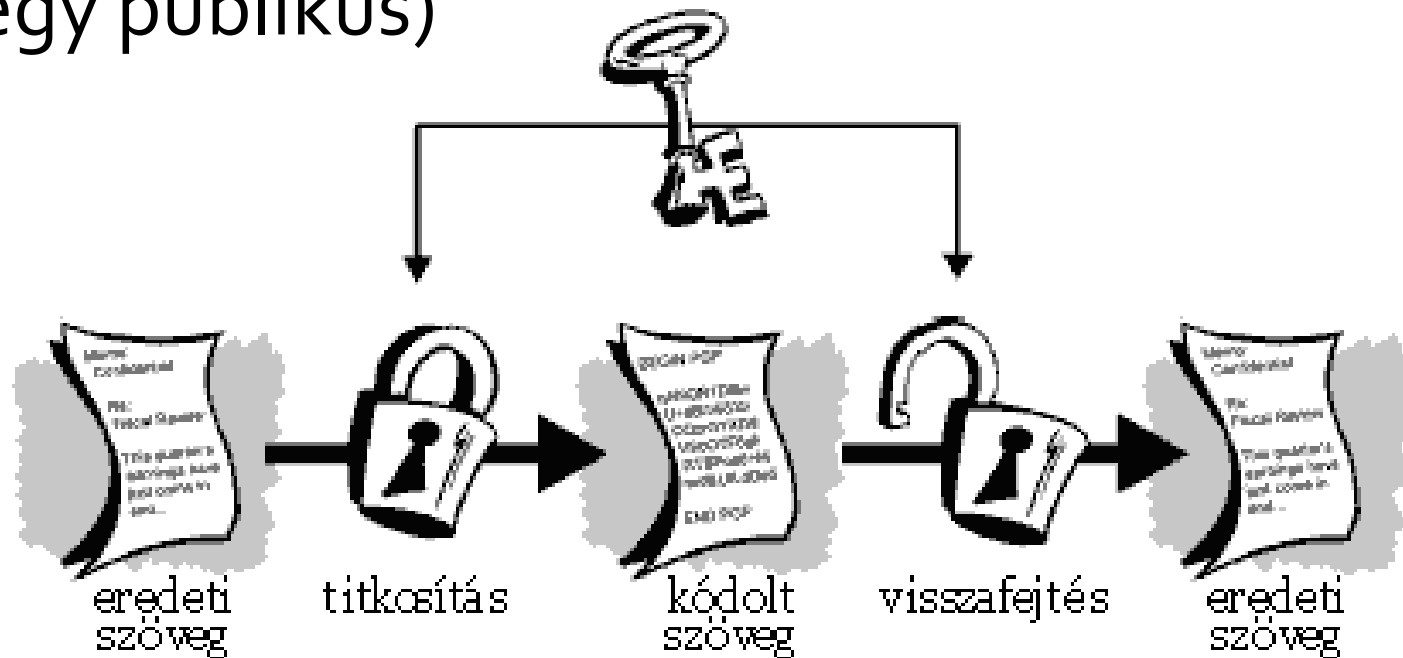


run-length encoding



# Titkosítás

- DES (egy kulcs), RSA (két kulcs, egy privát és egy publikus)





# Adatok tárolása

- Az adatokat adatbázisokban, adattárházakban, adatközpontokban tároljuk
- Az adatműveleteket speciális rendszerek, adatbázis-kezelők végzik. (Oracle, MS SQL, MySQL, MS Access, PostgreSQL, ...)



# Adatok betöltése egy helyre

- Az adattárházakba különböző helyeken tárolt adatokat is be lehet tölteni, ha egy helyen akarjuk feldolgozni az információt.



# Az információ visszakeresése

- Mennyi Euro van ma a számlámon?
- **Select** egyetlen **from** számla **where** datum='2015.09.25'
- Az adatbázis-kezelőkben lekérdező nyelvek segítségével fogalmazhatjuk meg a kérdéseinket. Például SQL-ben.



```
SQLQuery2.sql - DA...PACIFIC\dalvi (56)* x SQLQuery1.sql - DA...PACIFIC\dalvi (52)*  
1 SELECT create_date AS 'SQL Server Installed Date',  
2     expiry_date AS 'SQL Server Expiry Date',  
3     DATEDIFF(dd,create_date, GETDATE()) 'No_of_Days_Used',  
4     ABS(DATEDIFF(dd,expiry_date,GETDATE())) 'No_of_Days_Left'  
5 FROM  
6 (SELECT sp.create_date,  
7     DATEADD(dd, 180, sp.create_date) AS Expiry_date  
8 FROM sys.server_principals sp  
9 WHERE sp.name = 'NT AUTHORITY\SYSTEM') as exp_date_tbs
```

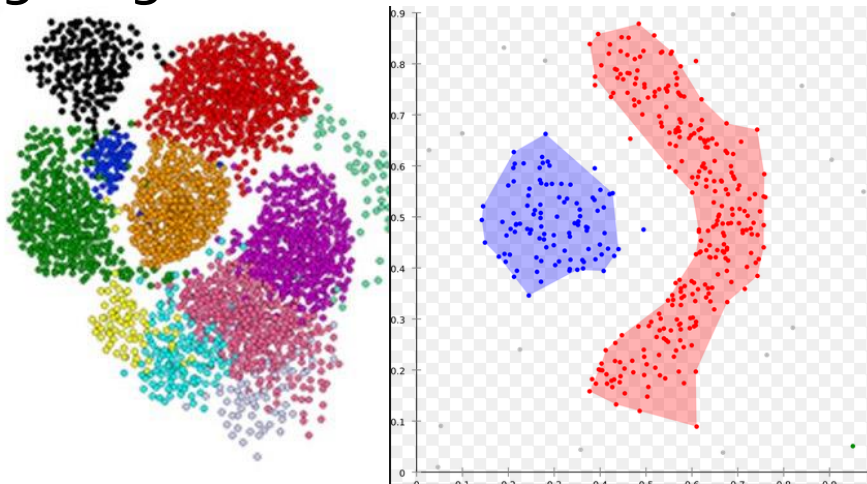
100 %

Results Messages

	SQL Server Installed Date	SQL Server Expiry Date	No_of_Days_Used	No_of_Days_Left
1	2012-10-04 00:09:42.820	2013-04-02 00:09:42.820	141	39

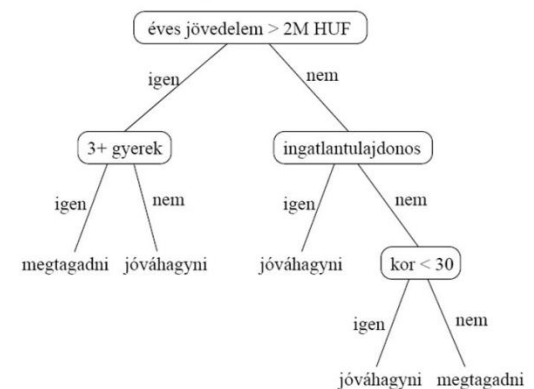
# Az információ feldolgozása, elemzése

- Az **adatbányászat** a nagy mennyiségű adatokban rejlő információk, újszerű, érvényes, nem triviális és vélhetően hasznos és magyarázható összefüggések félautomatikus feltárása különféle algoritmusok alkalmazásával.
- Speciális adatbányászati rendszerek (RapidMiner, R, SAS, ...) segítségével lehet ezeket a feladatokat elvégezni.



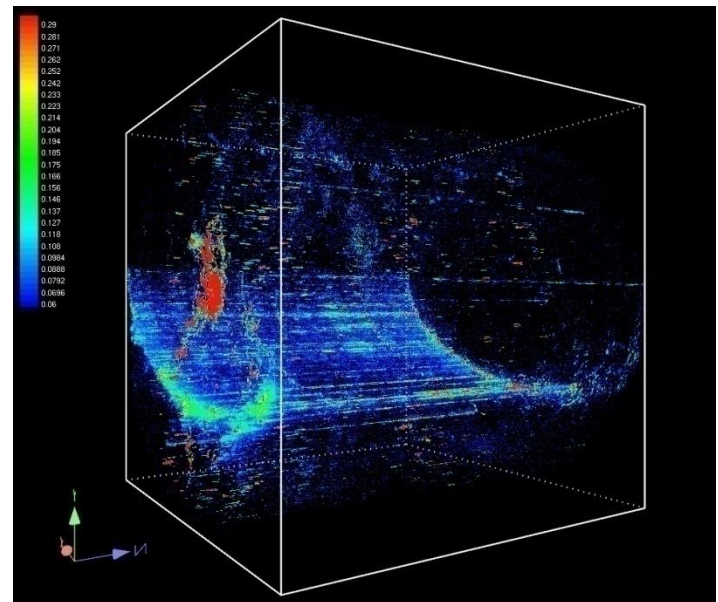
## Döntési fa példa hitelbírálatra

(forrás: <http://www.cs.bme.hu/~bodoni/magyar/adatbanyaszat/tanulmany/adatbanyaszat.pdf>)



# Az információ megjelenítése, vizualizáció

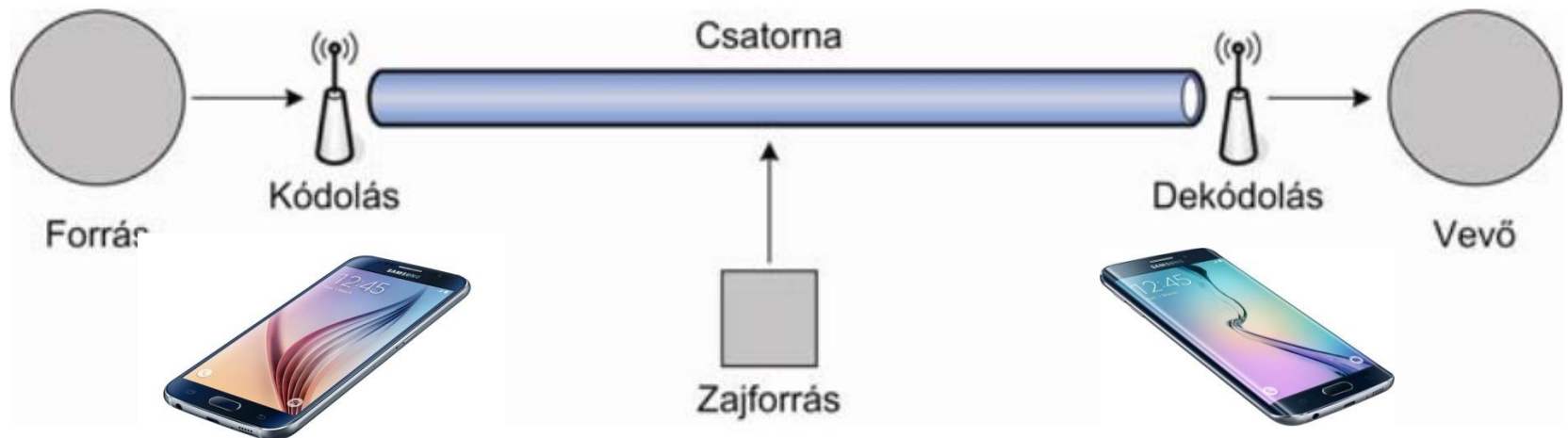
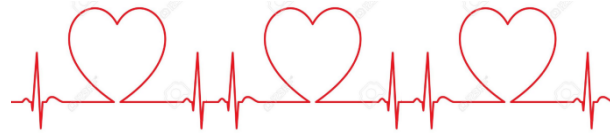
- Grafikus eszközökkel, 2D-ben, vagy 3D-ben, nagyítható, forgatható grafikonokkal ábrázoljuk az adatokat.





# Az információ továbbítása

- Shannon modell:



# Az információ mentése

- Meghibásodás után helyre kell tudnunk állítani az adatokat. Másolatot készítünk, archiváljuk az adatbázist.
- Ha közben futnak programok, akkor naplózásra (melyik program milyen adatokat módosított) is szükségünk lesz.

**BACKUP**



# Az informatika legfontosabb eszköze a számítógép





# Milyen elven működjön a számítógép?

## Neumann-elvek

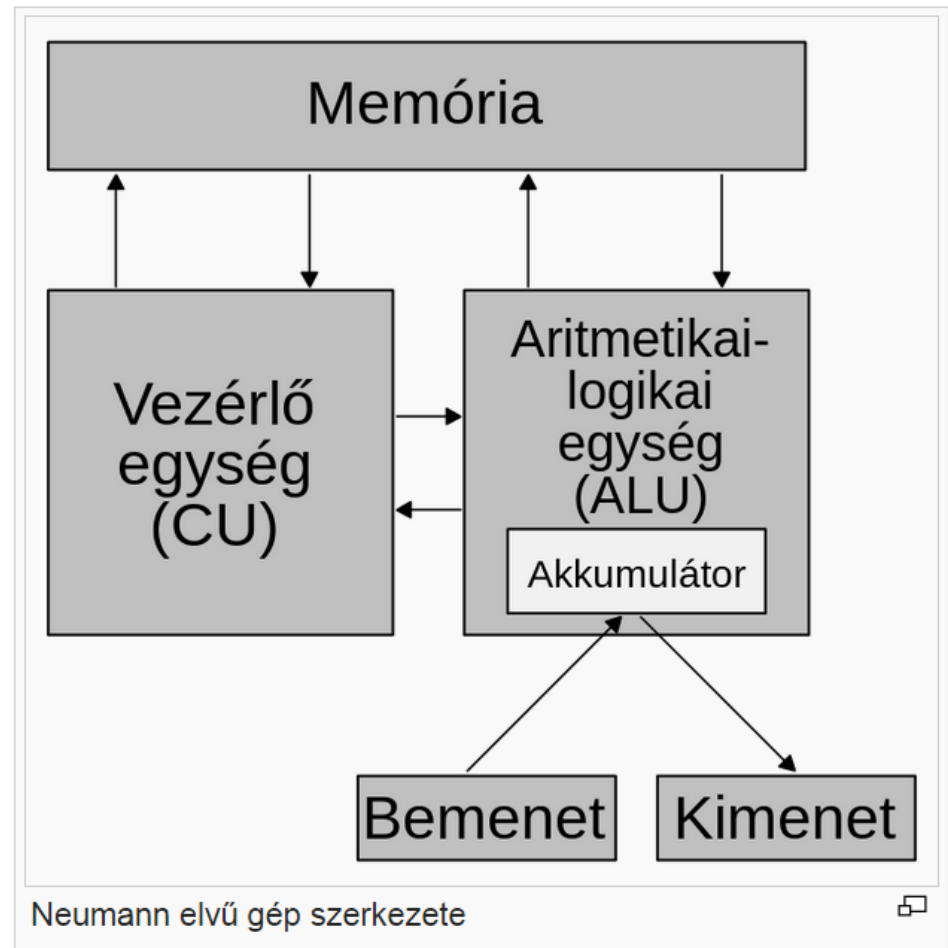
---

1. Teljesen elektronikus működés (– ez Neumann idejében elektroncsöves felépítést jelentett, amit később a tranzisztoros, majd az integrált áramkörös felépítés követett)
2. **Kettes számrendszer** használata (– az összes művelet, pl. összeadás, szorzás, kettes számrendszerbeli logikai műveletekre redukálható)
3. Belső memória használata
4. Tárolt program elve. A számításokhoz szükséges adatokat és **programutasításokat** a gép azonos módon, egyaránt a belső **memóriában** (operatív tár) tárolja.<sup>[2]</sup>
5. Soros utasításvégrehajtás (az utasítások végrehajtása időben egymás után történjen; ennek egy alternatívája a párhuzamos utasításvégrehajtás, amikor több utasítás egyidejűleg is végrehajtható: ezt a lehetőséget Neumann elvetette)
6. Univerzális felhasználhatóság (programozhatóság; a különböző feladatok programokkal legyenek megoldva, nem pedig erre a célra épített hardverrel)
7. Szerkezet: öt funkcionális egység (**aritmetikai egység**, központi vezérlőegység, memóriák, bemeneti és kimeneti egységek)

# A Neumann-gép szerkezete

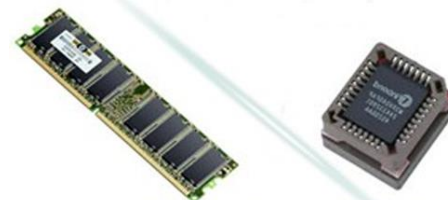
## A Neumann-elvű számítógépek elméleti felépítése

- központi egység
  - központi feldolgozó egység
    - központi vezérlő egység
    - aritmetikai-logikai egység
    - regiszterblokk
    - gyorsítómemória
    - matematikai társprocesszor
  - operatív tár (memória)
- háttértárak
- perifériák
  - input perifériák
  - output perifériák



# Perifériák Neumann elvei alapján

- központi egység (CPU, Central Processing Unit, Központi Feldolgozó Egység, [processzor](#))
- memória
  - ROM (Read-Only Memory) csak olvasható
  - RAM olvasható, írható és bővíthető
- háttértárak
  - merevlemez
  - flash
  - CD / DVD
  - SSD
  - stb.
- bemenet
  - billentyűzet
  - egér
  - stb.
- kimenet
  - Monitor
  - Nyomtató

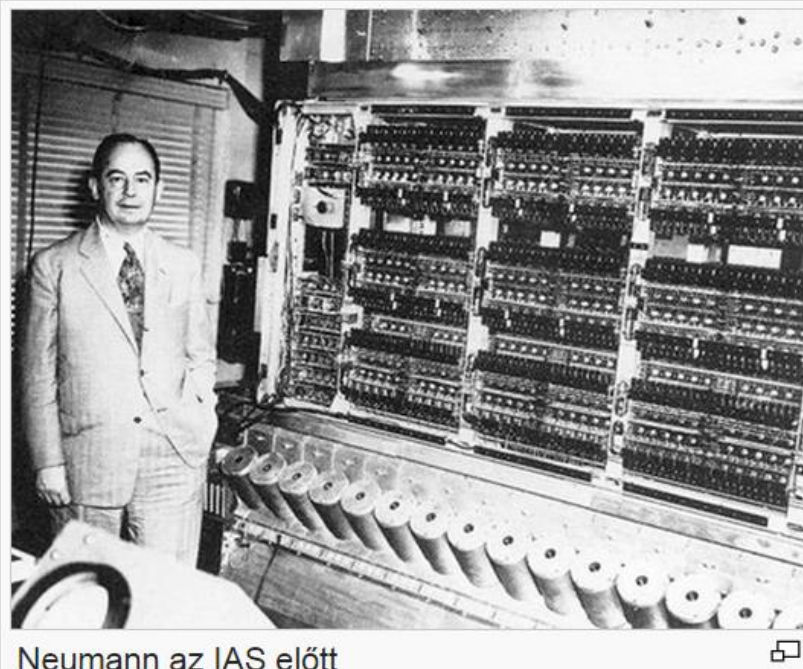


**RAM - ROM**



# Neumann János

- [Budapest, 1903-Washington, 1957](#)
- magyar matematikus,
- az első számítógép megalkotója



Neumann az IAS előtt



Neumann az 1940-es években

# Köszönöm a figyelmet!



KÖSZÖNÖM

# Gyakorlati feladatok

- CAT – computational and algorithmic thinking (számítási és algoritmikus gondolkodás) legfontosabb elemei
  - Absztrakció: a problémát számokkal, betűkkel, rajzokkal, stb. reprezentáljuk
  - Dekompozíció: a problémát visszavezetjük egyszerűbb részfeladatokra
  - Megoldó algoritmus tervezése
  - Minták, összefüggések keresése, általánosítás



# A CAT fogalmai és megközelítései



# Problémák reprezentálása

- 3-szor 3-as amőba játékban egy állást, egy lépést, hogyan reprezentálhatunk, hogy lehet eldönteni, hogy vége a játéknak, ki nyert ?

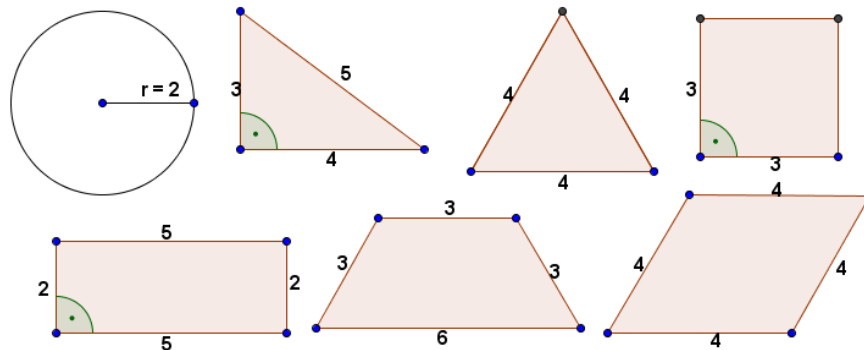






# Problémák reprezentálása

- Hogy lehet egyenest, illetve az alábbi síkidomokat reprezentálni, ha a bal sarkuk, illetve a kör középpontja az origóban van, és az egyenes is átmegy az origón?
- Ha tetszőleges helyen lehetnek, meg lehet-e állapítani, hogy metszi egymást két síkidom?



# Problémák reprezentálása

- Hogy lehet Komárom térképét, utcákat, kereszteződéseket, utcák hosszát, egy irányú utcákat reprezentálni?
- A reprezentálás alapján hogy lehetne reprezentálni, hogy jöttünk ma be otthonról az egyetemre, mennyi az út hossza?
- Hogy lehetne megállapítani, hogy el lehet-e jutni egyik helyről egy másikra?



# Problémák reprezentálása

- 12 érme között egy hamis van, de nem tudjuk, hogy könnyebb vagy nehezebb. Kétkarú mérleggel végezhetünk méréseket, hogy megállapítsuk melyik a hamis.
- Hogy lehet reprezentálni a feladatot, egy mérési sorozatot? (Mennyi a legkevesebb mérés, ami szükséges?)



# Problémák reprezentálása

- 10 ládában 5 grammos érmék vannak, kivéve az egyiket, amelyikben mindegyik érme csak 4 grammos.
- Hogy lehet minél kevesebb méréssel megállapítani, melyik ládában vannak a selejtes érmék?
- Reprezentáljuk a problémát, és egy nem feltétlen optimális mérési sorozatot.



# Problémák reprezentálása

- **Hölgyek társasága**
- Egy társaságban van három hölgy. Szeretnék kideríteni, hogy közülük ki a legidősebb és a legfiatalabb, de úgy, hogy ezen kívül semmi más információt ne szerezhessen egyikük sem. Ehhez persze annyit sugdolózhatnak, amennyit akarnak. Hogyan csinálják?



# Algoritmus

- Adjunk algoritmust, eljárást arra, hogy lehet megszámolni, hányan vannak a szemináriumon, és a végén mindenki tudja ezt a számot.
- Megoldások:
  - vezetőt választanak, egyesével megszámolja, bemondja
  - mindenkinek van egy száma, kezdetben 1, felállnak, párok alakulnak, a párok összeadják a számukat, az egyik leül, amíg egy marad, bemondja a számát.
- Melyik gyorsabb, miért? (A párok párhuzamosan dolgozhatnak,  $\log(n)$  idő.)



# Algoritmus

- Álljanak ki a hallgatók, és álljanak nagyság szerinti sorrendbe. Írjuk le, algoritmussal, mit csináltak.
- Hajtsák végre a buborék rendezést, párokból mindig balra áll a magasabb, amíg van változás. Írjuk le a módszert.



# Információáramlás

- 10 hajótöröttet meg akarnak enni a bennszülöttek. Mindegyik kap egy piros vagy kék sapkát és libasorba állítják őket, bármelyik az összes előtte levő sapkáját látja. Miután megbeszélték a stratégiájukat, hátulról kezdve, mindegyik tippelhet, hogy milyen sapka van a fején, ha eltalálja életben marad, különben megeszik.
- Milyen módszerrel marad életben a legtöbb hajótörött?
- Megoldás: Az utolsó pirosat mond, ha páros sok piros sapkát lát, az előtte lévő ebből meg tudja határozni, hogy milyen színű van a fején, és így tovább, mindegyik a mögötte elhangzottakból ki tudja számolni, hogy milyen színű sapka van a fején. Így 9,5 menthető meg átlagban.
- (Ha párok alakulnak, és bemondja az előtte levő színét, akkor 5 menthető meg, általában ez szokott a diákoknak eszükbe jutni.)



# Házi feladat

- Reprezentáljuk két különböző szakos hallgató órarendjét. Hogy lehetne a két reprezentáció alapján megkapni, hogy milyen kurzusokra járnak mindketten?
- Öt barát versenyt futott. Dénes keserűen állapította meg, hogy nem ő lett az első. György a harmadik helyen végzett, Béla pedig jobb helyezést ért el, mint György. Péter azt vette észre, hogy ő közvetlen Dénes után ért célba, és Béla nem a második lett. András nem lett a versenyben sem első, sem utolsó. Ki hányadik lett, ha a versenyen nem fordult elő azonos helyezés?



# Házi feladat

- Egy kör alakú szállóban egymásba nyílnak a szobák. A szobák teljesen egyformák, minden szobában egy lámpa és egy kapcsoló van, amivel fel vagy le lehet kapcsolni a lámpát. Nem tudjuk, milyen állásban hagyták a lámpákat.

A szobákban a kör mentén bármilyen irányba mehetünk, többször is, le- vagy felkapcsolhatjuk a lámpákat.

Hogy tudnánk megállapítani, hogy hány szoba van?

Útmutatás: Kapcsoljuk fel a lámpát, ahol vagyunk, és próbáljuk körbehaladva lekapcsolni.



# Kódolás, kettes számrendszer

- Mit látunk a képen, hogyan kódolnánk?



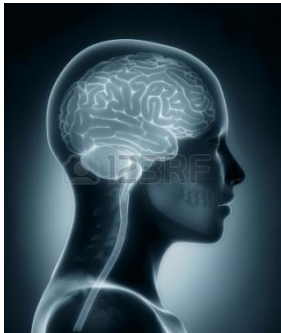
Isten zöld tenyerén,  
teremtett templom áll,  
falában századok,  
bent színes félhomály.

Kint maradsz-e rabként,  
hol tágas még a tér,  
vagy belépsz szabadon,  
s fényelked célba ér.

- Szentháromság-templom (Velemér), 13. század

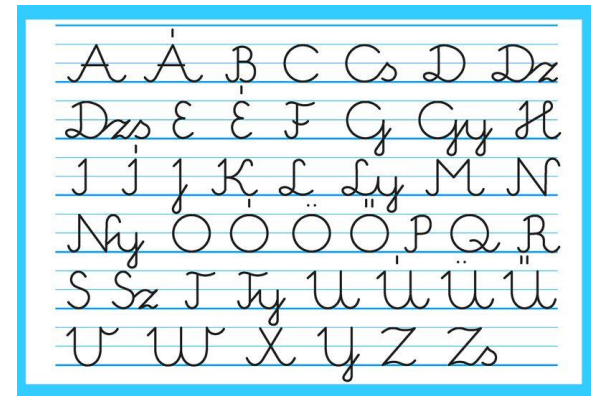
# Miért van szükség kódolásra?

- **Tárolás:** Tömörebben tároljuk az információt.
- **Feldolgozás:** Számítógép számára feldolgozhatóvá tesszük az információt.
- **Továbbítás:** Üzenet formájában elküldjük másnak az információt, akire hatni akarunk ennek tartalmával, és aki dekódolja, visszaalakítja azt saját számára.



# Kódolás természetes nyelven

- Betűket, szavakat, mondatokat használunk egy természetes (magyar, szlovák, angol, stb.) nyelven.
  - Ábécé (sok jel, szimbólum)
  - Szavak (betűsorozatok), kb. 100 ezer szó szerepel egy nyelvben
  - Mondatok (szósorozatok)
  - Az értelmes sorozatokat, szavakat, mondatokat a szókincs (szótár) és a nyelvtan határozza meg.
- Szentháromság-templom (Velemér), 13. század
  - Ember számára könnyen érthető, dekódolható, de a gép nehezen tudná értelmezni.
  - Hogy lehetne érzelmeket kódolni, például, amit egy vers vált ki az olvasóban?



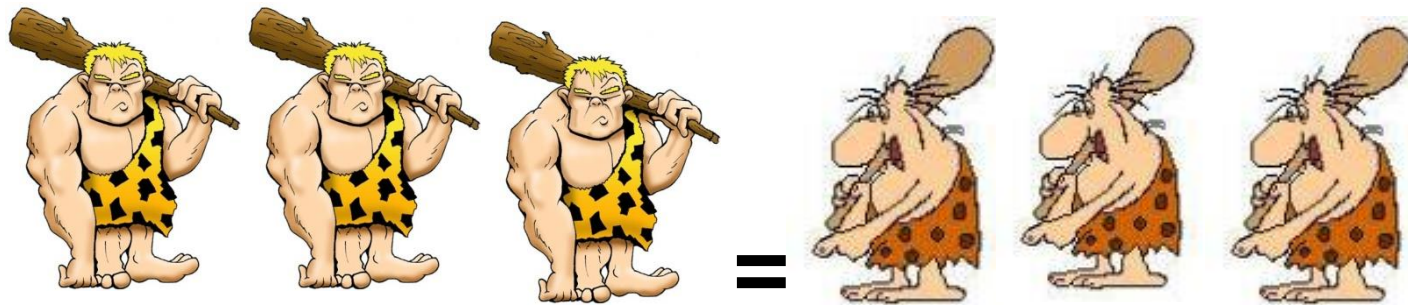
# A kódolás számok segítségével

- **Szintaktika** (forma):  
milyen számjegyeket  
és hogyan használunk
- **Szemantika**  
(jelentés): mi, mennyi  
a leírt szám jelentése
- **1011** = tizenegy vagy  
ezertizenegy vagy  
mennyi?

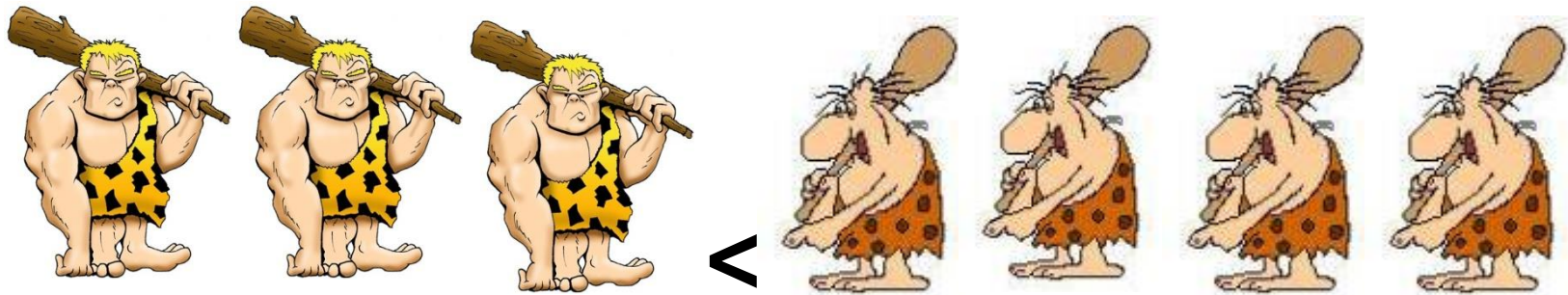


# A számok, számolás megjelenése

- Számfogalom:



kössünk békét

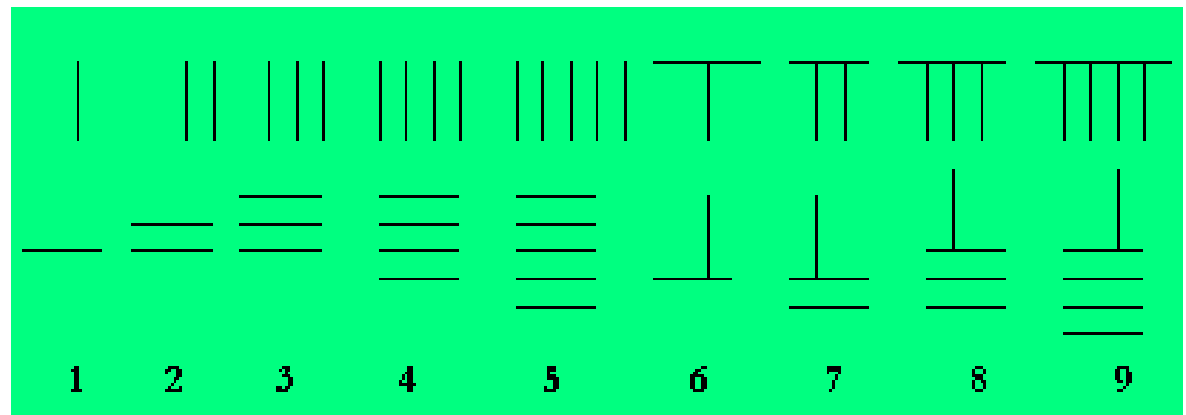


add oda a sárga tigrisbundádat



# Digit = ujj

- Kezdetben ujjakat, pálcikákat használtak számolásra



# Számrendszerek

$$318 = 3 \cdot 100 + 1 \cdot 10 + 8 \cdot 1$$

**Szám helyiértéke**

$$318 = 3 \cdot 10^2 + 1 \cdot 10^1 + 8 \cdot 10^0$$

**Szám alakí értéke**

**Számjegyek: 0,1,2,3,4,5,6,7,8,9**

**Számrendszer alapja: 10**

$$318 = 300 + 10 + 8$$

**Szám valódi értéke**

**Decimális számrendszer**

# q-alapú számrendszer

$$318_{(10)} = 3 \cdot 100 + 1 \cdot 10 + 8 \cdot 1$$

**10-es alapú**

$$318_{(q)} = 3 \cdot q^2 + 1 \cdot q^1 + 8 \cdot q^0$$

**q alapú**

**x** szám **q**-alapú számrendszerbeli alakja:  $a_n \dots a_1 a_0$ ,  
ha:  $0 \leq a_i < q, \quad i = 0, 1, \dots, n$

$$x = a_n \cdot q^n + \dots + a_1 \cdot q^1 + a_0 \cdot q^0$$

**Számjegyek: 0, 1, ..., (q-1)**

# Bináris számrendszer

$$11010110_{(2)} =$$

$$1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

**Számjegyek: 0,1**

**A számítástechnika a bináris számrendszerre épül**

# Hexadecimális számrendszer

$$14FB = 1 \cdot 16^3 + 4 \cdot 16^2 + F \cdot 16^1 + B \cdot 16^0$$

$$= 1 \cdot 4096 + 4 \cdot 256 + 15 \cdot 16 + 11 \cdot 1$$

$$= 5371_{(10)}$$

**16-os alapú**

**Számjegyek: 0, 1, ..., 9, A, B, C, D, E, F**

# Számrendszerek: összefoglaló

	Decimális	Bináris	Hexa-decimális	q-alapú
<b>Alap</b>	10	2	16	q
<b>Számjegyek</b>	0, 1, 2, ..., 9	0, 1	0, 1, ..., 9, A, B, C, D, E, F	0, ..., q-1
<b>Helyiértékek</b>	..., 100, 10, 1	..., 8, 4, 2, 1	..., 256, 16, 1	..., $q^2$ , $q^1$ , $q^0$
<b>512 valódi értéke</b>	$5 \cdot 100 + 1 \cdot 10 + 2 \cdot 1$		$5 \cdot 256 + 1 \cdot 16 + 2 \cdot 1$	
<b>110 valódi értéke</b>	$1 \cdot 100 + 1 \cdot 10 + 0 \cdot 1$	$1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1$	$1 \cdot 256 + 1 \cdot 16 + 0 \cdot 1$	

# 2 hatványai 10-ed rendig

$2^1$	$2^2$	$2^3$	$2^4$	$2^5$	$2^6$	$2^7$	$2^8$	$2^9$	$2^{10}$
2	4	8	16	32	64	128	256	512	1024

# Átváltás a számrendszerek között

2, 8, 10, 16




# Átváltás 10-es számrendszerből 2-esbe

$$131_{10} = 10000011_2$$

## Átváltás menete:

1. Készítsünk egy 2-oszlopos táblázatot
2. Írjuk fel a számot a bal felső sarokba
3. Osszuk el a számot 2-vel
  - a) Az osztás eredményét írjuk a szám alá
  - b) Az osztás maradékát írjuk a szám mellé
4. Az osztást ismétlegessük, amíg a bal oldalon 0-t nem kapunk
5. A jobb oldali oszlop számjegyeit olvassuk össze letről felfelé

:2 ↻	131	1
	65	1
	32	0
	16	0
	8	0
	4	0
	2	0
	1	1
0		



Átváltás 10-es számrendszerből 2-esbe

# Átváltás 10-es számrendszerből 8-asba

$$131_{10} = 203_8$$

## Átváltás menete:

1. Készítsünk egy 2-oszlopos táblázatot
2. Írjuk fel a számot a bal felső sarokba
3. Osszuk el a számot 8-cal
  - a) Az osztás eredményét írjuk a szám alá
  - b) Az osztás maradékát írjuk a szám mellé
4. Az osztást ismétlegessük, amíg a bal oldalon 0-t nem kapunk
5. A jobb oldali oszlop számjegyeit olvassuk össze letről felfelé

:8	↻	131	3
		16	0
		2	2
		0	

↑

Átváltás 10-es számrendszerből 8-asba


# Átváltás 10-es számrendszerből 16-osba

$$131_{10} = 83_{16}$$

## Átváltás menete:

1. Készítsünk egy 2-oszlopos táblázatot
2. Írjuk fel a számot a bal felső sarokba
3. Osszuk el a számot 16-tal
  - a) Az osztás eredményét írjuk a szám alá
  - b) Az osztás maradékát írjuk a szám mellé
4. Az osztást ismétlegessük, amíg a bal oldalon 0-t nem kapunk
5. A jobb oldali oszlop számjegyeit olvassuk össze letről felfelé

:16 ↻	131	3
	8	8
	0	



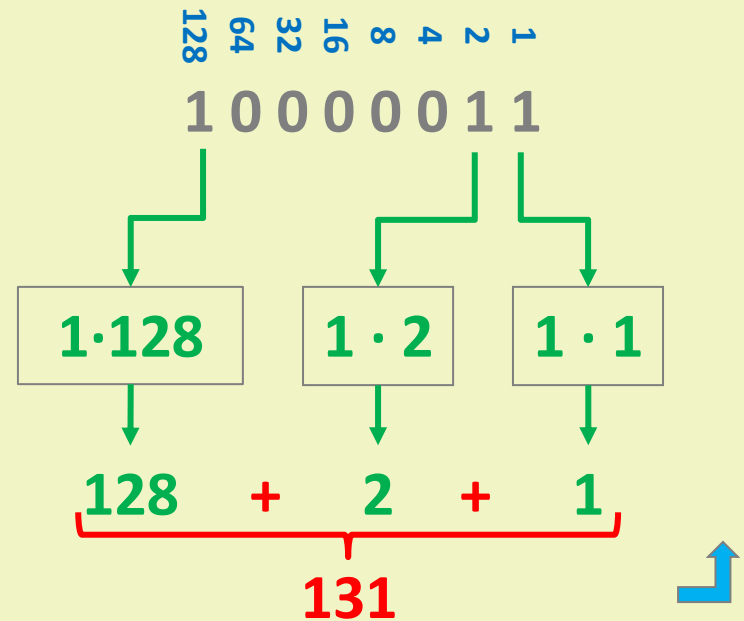
# Átváltás 10-es számrendszerből 16-osba

# Átváltás 2-es számrendszerből 10-esbe

$$10000011_2 = 131_{10}$$

## Átváltás menete:

1. Írjuk fel az átváltandó számot
2. Írjuk a számjegyek fölé 2 hatványait
3. Szorozzuk össze a számjegyeket a fölöttük lévő hatványokkal
4. Adjuk össze a szorzatokat
5. Az összeg lesz a végeredmény



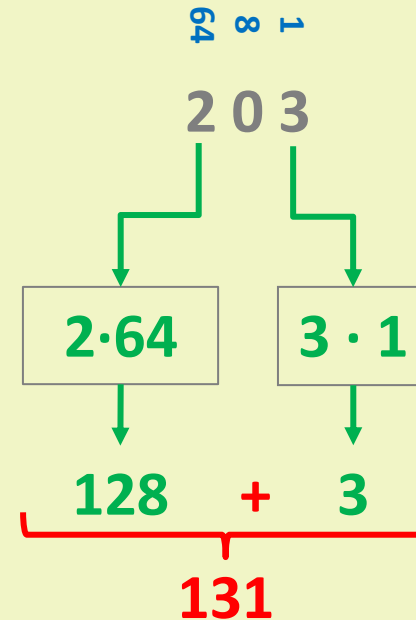
Átváltás 2-es számrendszerből 10-esbe

# Átváltás 8-as számrendszerből 10-esbe

$$203_{\underline{8}} = \mathbf{131}_{\underline{10}}$$

## Átváltás menete:

1. Írjuk fel az átváltandó számot
2. Írjuk a számjegyek fölé 8 hatványait
3. Szorozzuk össze a számjegyeket a fölöttük lévő hatványokkal
4. Adjuk össze a szorzatokat
5. Az összeg lesz a végeredmény



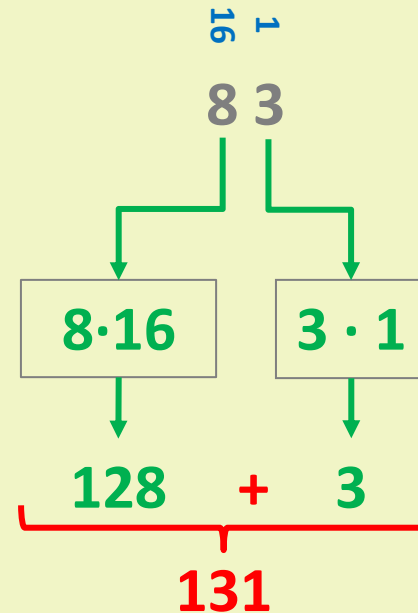
Átváltás 8-as számrendszerből 10-esbe

# Átváltás 16-os számrendszerből 10-esbe

$$83_{16} = 131_{10}$$

## Átváltás menete:

1. Írjuk fel az átváltandó számot
2. Írjuk a számjegyek fölé 16 hatványait
3. Szorozzuk össze a számjegyeket a fölöttük lévő hatványokkal
4. Adjuk össze a szorzatokat
5. Az összeg lesz a végeredmény



Átváltás 16-os számrendszerből 10-esbe

# Különbség az átváltásoknál

## 10-esből $X$ -esbe

Átváltás menete:

1. Készítsünk egy 2-oszlopos táblázatot
2. Írjuk fel a számot a bal felső sarokba
3. Osszuk el a számot  $X$ -szel
  - a) Az osztás eredményét írjuk a szám alá
  - b) Az osztás maradékát írjuk a szám mellé
4. Az osztást ismételtessük, amíg a bal oldalon 0-t nem kapunk
5. A jobb oldali oszlop számjegyeit olvassuk össze letről felfelé

## $X$ -esből 10-esbe

Átváltás menete:

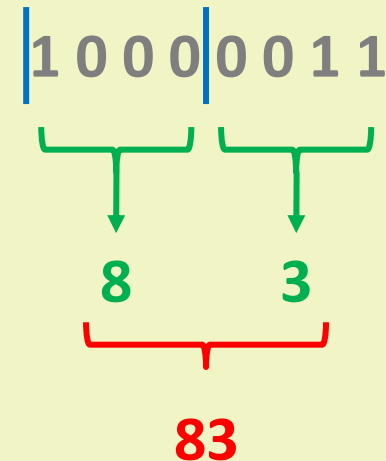
1. Írjuk fel az átváltandó számot
2. Írjuk a számjegyek fölé  $X$  hatványait
3. Szorozzuk össze a számjegyeket a fölöttük lévő hatványokkal
4. Adjuk össze a szorzatokat
5. Az összeg lesz a végeredmény

# Átváltás 2-es számrendszerből 16-osba

$$10000011_{\underline{2}} = \mathbf{83}_{\underline{16}}$$

## Átváltás menete:

1. Írjuk fel az átváltandó számot
2. Hátról indulva osszuk fel a számot 4 bites csoportokra (digitekre), ha kell, írjunk 0-kat a szám elé
3. A 4 bites csoportokat egyenként alakítsuk át (segédtábla segítségével)
4. Az átváltások eredményét balról jobbra kell összeolvasni
5. A lesz a végeredmény



# Átváltás 2-es számrendszerből 16-osba

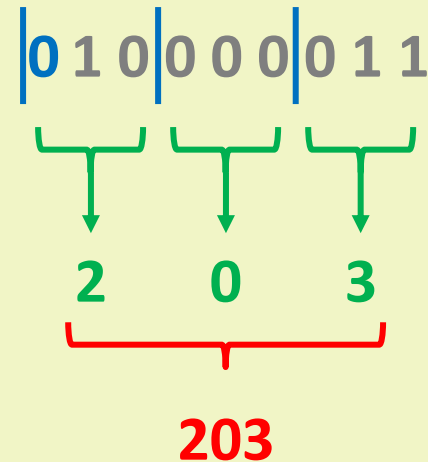


# Átváltás 2-es számrendszerből 8-asba

$$10000011_2 = 203_8$$

## Átváltás menete:

1. Írjuk fel az átváltandó számot
2. Hátról indulva osszuk fel a számot 3 bites csoportokra, ha kell, írjunk 0-kat a szám elé
3. A 3 bites csoportokat egyenként alakítsuk át (segédtábla segítségével)
4. Az átváltások eredményét balról jobbra kell összeolvasni
5. A kapott szám lesz a végeredmény



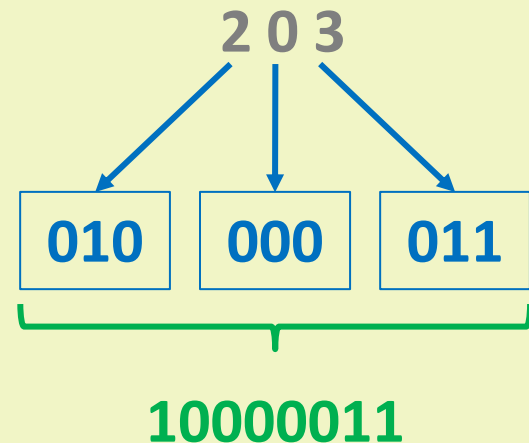
# Átváltás 2-es számrendszerből 8-asba

# Átváltás 8-as számrendszerből 2-esbe

$$203_{\underline{8}} = 10000011_{\underline{2}}$$

## Átváltás menete:

1. Írjuk fel az átváltandó számot
2. Minden számjegyet írjunk át 3 bites bináris számra (segédtablával)
3. A 3 bites csoportokat balról jobbra olvassuk össze (elején lévő 0-kat nem)
4. A kapott szám lesz a végeredmény



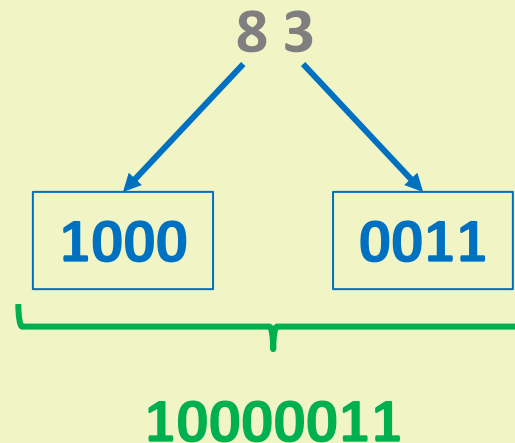
Átváltás 8-as számrendszerből 2-esbe

# Átváltás 16-os számrendszerből 2-esbe

$$83_{\underline{16}} = 1000?0011_{\underline{2}}$$

## Átváltás menete:

1. Írjuk fel az átváltandó számot
2. Minden számjegyet írjunk át 4 bites bináris számra (segédtablával)
3. A 4 bites csoportokat balról jobbra olvassuk össze (elején lévő 0-kat nem)
4. A kapott szám lesz a végeredmény



Átváltás 16-os számrendszerből 2-esbe

# Különbség az átváltásoknál

## 2-esből 8-asba vagy 16-osba

### Átváltás menete:

1. Írjuk fel az átváltandó számot
2. Hátról indulva osszuk fel a számot 3 vagy 4 bites csoportokra, ha kell, írjunk 0-kat a szám elé
3. A 3-4 bites csoportokat egyenként alakítsuk át (segéd tábla segítségével)
4. Az átváltások eredményét balról jobbra kell összeolvasni
5. A kapott szám lesz a végeredmény

## 8-asból vagy 16-osból 2-esbe

### Átváltás menete:

1. Írjuk fel az átváltandó számot
2. Minden számjegyet írjunk át 3 vagy 4 bites bináris számra (segéd táblával)
3. A 3-4 bites csoportokat balról jobbra olvassuk össze (elején lévő 0-kat nem)
4. A kapott szám lesz a végeredmény

# Definíciók

- **Bit** – egyetlen bináris jegy
- **Bájt (byte)** – egy 8-bites egység (8 jegyű bináris szám)
- **1 KB (kilobájt)** =  $2^{10}$  bájt = 1024 bájt
- **1 MB (megabájt)** =  $2^{20}$  bájt =  $1024^2$  bájt =  
= 1 048 576 bájt
- **1 GB (gigabájt)** =  $2^{30}$  bájt =  $1024^3$  bájt =  
= 1 073 741 824 bájt
- **1 TB (terabájt)** =  $2^{40}$  bájt =  $1024^4$  bájt =  
= 1 086 511 627 776 bájt

# Információ

## Számítógép:

- információk tárolására és feldolgozására szolgáló eszköz.

## Információ:

- A címzettje számára új, vagy általa nem ismert adat, hír, közlés vagy tájékoztatás.
- A releváns adat, amely valamely bizonytalanság megszüntetéséhez elegendő. Nem minden adat információ! Csak az értékes (fontos) adat.

- alapegysége: **bit**
- mérése: **bájt**-okban
- 1 szó: 2 bájt



# Információ

## Adat:

- Valakinek vagy valaminek a megismerését, jellemzését segítő tény, részlet.
- Valamilyen formában rögzített ismeret = **potenciális információ**.
- Jellege szerint kvantitatív vagy kvalitatív.

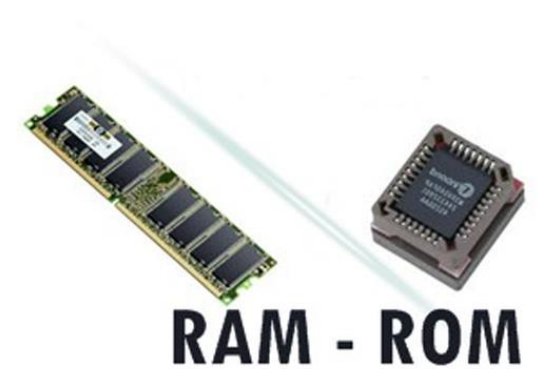
## Információ sokfélesége

(numerikus adatsor, szöveg, zene, egyetlen elektronikus jel, kép, videofelvétel,...)

# Információ tárolása

## ■ Memória

- ideiglenes tárolás
- kikapcsoláskor tartalma elvész
- kisebb kapacitás
- gyors elérés



## ■ Háttértárak

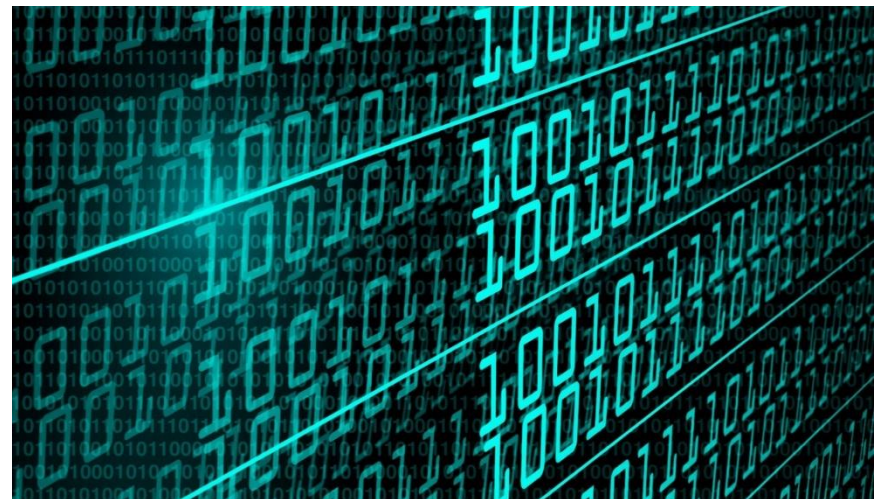
- hosszú távú, biztonságos tárolás
- lassabb elérés
- általában nagyobb tárhelykapacitás





# Információ kódolása

- **Kódolás:** közölnivalónknak a szokásos-tól eltérő ábrázolása, kifejezési formája
- A számítógépen a tárolandó, feldolgozandó információt bináris kódkészlettel fejezzük ki
- matematikai modell



# Matematikai modellezés

- $S = \{s_1, s_2, \dots, s_p\}$  - elsődleges szimbólumok, üzenetek halmaza (amit ábrázolni akarunk)
- $A = \{a_1, a_2, \dots, a_q\}$  - kódok ábécéje, elemei betűk (segítségükkel kódolunk)
- $n$  hosszúságú szavak halmaza:

$$A^n = \{w = a_{i_1} a_{i_2} a_{i_3} \dots a_{i_n} \mid a_{ij} \in A, j = 1, n\}$$

# Matematikai modellezés

- összes szavak halmaza:

$$A^+ = A \cup A^2 \cup A^3 \cup \dots \cup A^n$$

- **kód:**  $C : S \rightarrow A^+$  leképezés

- **egyenletes** (pl. számjegyek bináris ábrázolása 8 biten) **01001101**

- **nem egyenletes** (pl. Morse ábécé)

A	•-	J	•---	S	•••
B	-•••	K	-•-	T	-
C	-•-	L	••••	U	••-
D	-••	M	-	V	•••-
E	•	N	-•	W	•---
F	••-	O	---	X	-••-
G	-••	P	••••	Y	-•---
H	••••	Q	-•-	Z	-•••
I	••	R	••-		

- **dekódolás:**  $C^{-1} : C(S) \rightarrow S$

- egyértelműen dekódolható függvény

# Szövegek kódolása

- **Karakterek** – betűk, számjegyek, írásjelek, speciális jelek összefoglaló neve;
- gépi reprezentálása: binárisan
- Lehetne-e minden mondatot (100 betűből állót) számmal kódolni? Kb.  $44^{100}$  mondat. Nagy lenne a kódtáblázat, nehezen lehetne megtalálni a kódhoz tartozó mondatot.

- Atomok száma a Földön:  $9.01 \times 10^{49}$



Ehelyett:

- 1 karakter  $\leftrightarrow$  1 bináris, hexa vagy decimális számkód
- **Hogyan?**  
Kódolási szabványok

# Szöveg konvertálása bináris számokká

Isten zöld tenyerén,  
teremtett templom áll,  
falában századok,  
bent színes félhomály.

Kint maradsz-e rabként,  
hol tágas még a tér,  
vagy belépsz szabadon,  
s fénylelked célba ér.



```
01001001 01110011 01110100 01100101 01101110 00100000 01111010
11000011 10110110 01101100 01100100 00100000 01110100 01100101
01101110 01111001 01100101 01110010 11000011 10101001 01101110
00101100 00100000 00001101 00001010 01110100 01100101 01110010
01100101 01101101 01110100 01100101 01110100 01110100 00100000
01110100 01100101 01101101 01110000 01101100 01101111 01101101
00100000 11000011 10100001 01101100 01101100 00101100 00001101
00001010 01100110 01100001 01101100 11000011 10100001 01100010
01100001 01101110 00100000 01110011 01111010 11000011 10100001
01111010 01100001 01100100 01101111 01101011 00101100 00100000
00001101 00001010 01100010 01100101 01101110 01110100 00100000
01110011 01111010 11000011 10101101 01101110 01100101 01110011
00100000 01100110 11000011 10101001 01101100 01101000 01101111
01101101 11000011 10100001 01101100 01111001 00101110 00001101
00001010 00001101 00001010 01001011 01101001 01101110 01110100
00100000 01101101 01100001 01110010 01100001 01100100 01110011
01111010 00101101 01100101 00100000 01110010 01100001 01100010
01101011 11000011 10101001 01101110 01110100 00101100 00100000
00001101 00001010 01101000 01101111 01101100 00100000 01110100
11000011 10100001 01100111 01100001 01110011 00100000 01101101
11000011 10101001 01100111 00100000 01100001 00100000 01110100
11000011 10101001 01110010 00101100 00001101 00001010 01110110
01100001 01100111 01111001 00100000 01100010 01100101 01101100
11000011 10101001 01110000 01110011 01111010 00100000 01110011
01111010 01100001 01100010 01100001 01100100 01101111 01101110
00101100 00001101 00001010 01110011 00100000 01100110 11000011
10101001 01101110 01111001 01101100 01100101 01101100 01101011
01100101 01100100 00100000 01100011 11000011 10101001 01101100
01100010 01100001 00100000 11000011 10101001 01110010 00101110
00100000
```

# Szöveg konvertálása decimális számokká

Isten zöld tenyerén,  
teremtett templom áll,  
falában századok,  
bent színes félhomály.

Kint maradsz-e rabként,  
hol tágas még a tér,  
vagy belépsz szabadon,  
s fénylelked célba ér.



```
73 115 116 101 110 32 122 195 182 108
100 32 116 101 110 121 101 114 195 169
110 44 32 13 10 116 101 114 101 109
116 101 116 116 32 116 101 109 112 108
111 109 32 195 161 108 108 44 13 10
102 97 108 195 161 98 97 110 32 115
122 195 161 122 97 100 111 107 44 32
13 10 98 101 110 116 32 115 122 195
173 110 101 115 32 102 195 169 108 104
111 109 195 161 108 121 46 13 10 13 10
75 105 110 116 32 109 97 114 97 100
115 122 45 101 32 114 97 98 107 195
169 110 116 44 32 13 10 104 111 108 32
116 195 161 103 97 115 32 109 195 169
103 32 97 32 116 195 169 114 44 13 10
118 97 103 121 32 98 101 108 195 169
112 115 122 32 115 122 97 98 97 100
111 110 44 13 10 115 32 102 195 169
110 121 108 101 108 107 101 100 32 99
195 169 108 98 97 32 195 169 114 46 32
```

# Szöveg konvertálása hexadecimális számokká

Isten zöld tenyerén,  
teremtett templom áll,  
falában századok,  
bent színes félhomály.

Kint maradsz-e rabként,  
hol tágas még a tér,  
vagy belépsz szabadon,  
s fénylelked célba ér.



49	73	74	65	6e	20	7a	c3	b6	6c	64	20	74
65	6e	79	65	72	c3	a9	6e	2c	20	0d	0a	74
65	72	65	6d	74	65	74	74	20	74	65	6d	70
6c	6f	6d	20	c3	a1	6c	6c	2c	0d	0a	66	61
6c	c3	a1	62	61	6e	20	73	7a	c3	a1	7a	61
64	6f	6b	2c	20	0d	0a	62	65	6e	74	20	73
7a	c3	ad	6e	65	73	20	66	c3	a9	6c	68	6f
6d	c3	a1	6c	79	2e	0d	0a	0d	0a	4b	69	6e
74	20	6d	61	72	61	64	73	7a	2d	65	20	72
61	62	6b	c3	a9	6e	74	2c	20	0d	0a	68	6f
6c	20	74	c3	a1	67	61	73	20	6d	c3	a9	67
20	61	20	74	c3	a9	72	2c	0d	0a	76	61	67
79	20	62	65	6c	c3	a9	70	73	7a	20	73	7a
61	62	61	64	6f	6e	2c	0d	0a	73	20	66	c3
a9	6e	79	6c	65	6c	6b	65	64	20	63	c3	a9
6c	62	61	20	c3	a9	72	2e	20				

# ASCII kódolás

- American Standard Code for Information Interchange
- 1 karakter  $\leftrightarrow$  1 byte
- 256-féle kód; kódtáblában rögzítve
- Alap karakterkészlet (rögzített): 0 - 127
- Kiegészítő karakterkészlet-grafikus jelek vagy nemzeti jelek (cserélhető, adott országra jellemző kódlapok): 128 - 255

**Példák:** 852-es Közép-Kelet-Európa, ISO-8859-2 (latin-2)

Ez utóbbi tartalmazza a latin betűs szláv nyelvek (horvát, szlovén, szlovák, cseh, lengyel), és a magyar, román, német nyelv ékezetes betűit.





# Az ASCII kódtáblázat

0		30	▲	60	<	90	Z	120	x	150	ı	180	ı	210	Ď	240	
1	☺	31	▼	61	=	91	[	121	y	151	Š	181	Á	211	Ě	241	"
2	☻	32		62	>	92	\	122	z	152	š	182	Â	212	ď	242	„
3	♥	33	!	63	?	93	]	123	{	153	Ö	183	Ë	213	Ň	243	ˇ
4	♦	34	"	64	@	94	^	124		154	Ü	184	Ş	214	í	244	˘
5	♣	35	#	65	A	95	_	125	}	155	Ť	185	ı	215	î	245	§
6	♠	36	\$	66	B	96	`	126	~	156	ť	186		216	ě	246	+
7	•	37	%	67	C	97	a	127	△	157	ł	187	⌈	217	ǰ	247	˙
8	◻	38	&	68	D	98	b	128	Ç	158	×	188	⌋	218	ǂ	248	◦
9	○	39		69	E	99	c	129	ü	159	č	189	Ž	219	■	249	”
10	◼	40	(	70	F	100	d	130	é	160	á	190	ž	220	■	250	˙
11	♂	41	)	71	G	101	e	131	â	161	í	191	⌌	221	Ť	251	ü
12	♀	42	*	72	H	102	f	132	ä	162	ó	192	⌍	222	Ů	252	Ř
13	♪	43	+	73	I	103	g	133	û	163	ú	193	⌎	223	■	253	ř
14	♫	44	,	74	J	104	h	134	ć	164	Ą	194	⌏	224	Ó	254	■
15	☀	45	-	75	K	105	i	135	ç	165	ą	195	⌐	225	β	255	
16	▶	46	.	76	L	106	j	136	ł	166	Ż	196	⌑	226	Ö		
17	◀	47	alt	77	M	107	k	137	ě	167	ž	197	⌒	227	Ň		
18	↕	48	0	78	N	108	l	138	Ő	168	Ę	198	⌓	228	ń		
19		49	1	79	O	109	m	139	ő	169	ę	199	⌔	229	ň		
20	¶	50	2	80	P	110	n	140	î	170	ı	200	⌕	230	Š		
21	§	51	3	81	Q	111	o	141	Ž	171	ž	201	⌖	231	š		
22	—	52	4	82	R	112	p	142	Ä	172	Č	202	⌗	232	Ř		
23	↕	53	5	83	S	113	q	143	Ć	173	š	203	⌘	233	Ú		
24	↑	54	6	84	T	114	r	144	É	174	«	204	⌙	234	ř		
25	↓	55	7	85	U	115	s	145	Ĺ	175	»	205	⌚	235	Ů		
26	→	56	8	86	V	116	t	146	Í	176	⌛	206	⌜	236	ý		
27	←	57	9	87	W	117	u	147	ô	177	⌝	207	⌞	237	Ý		
28	L	58	:	88	X	118	v	148	ö	178	⌟	208	⌞	238	ţ		
29	↔	59	;	89	Y	119	w	149	Ł	179		209	Đ	239	˘		

# ASCII karaktercsoportok

- Számjegyek: 0, 1, ..., 9
- Betűk: angol abc kis-, nagybetű
- Írásjelek: pl. szóköz, (,), /,!, ...
- Ékezetes betűk
- Grafikus karakterek (nem használjuk)
- Vezérlő karakterek:
  - nyomtató,
  - szöveg megjelenése a képernyőn
  - CR: sor elejére pozicionálás
  - LF : soremelés (CR+LF : sorvége jel)
  - FF : lapváltás

# Karakterek

- **Numerikus karakterek:**

0, 1, ..., 9

- **Alfabetikus karakterek:**

a, b, ..., z, A, B, ..., Z

- **Alfanumerikus karakterek:**

0, 1, ..., 9, a, b, ..., z, A, B, ..., Z

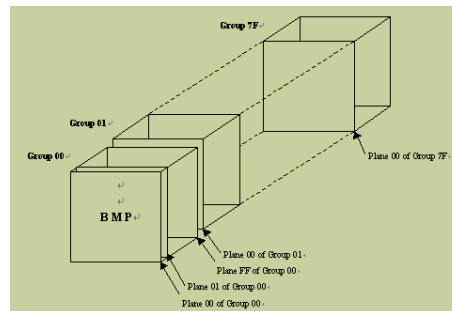
# Az Unicode szükségessége

- Az internet előretörésével egyre erősebb lett arra az igény, hogy a sok-sok karakterkódolás helyett egyetlen univerzális karakterkódolást alkalmazzanak, amit minden szoftver ugyanúgy képes kezelni.
- Például egy magyar „**árvíztűrő tükörfúrógép**” szót egy Kínában használt levelezőszoftver vagy egyéb szoftver képes legyen ugyanígy megjeleníteni, illetve a nálunk használt szoftverek is a kínai karaktereket.



# Unicode (ISO-10646) szabvány

- minden nyelv karaktere (pl. görög, kínai, cirill, héber, japán, koreai) egyetlen karakterkészlettel ábrázolható: platform-, program- és nyelvfüggetlen
- 1 karakter ábrázolására 2 bájtot használ
- 65536 elemű kódtábla (fix)
- első 128 elem: ASCII kódtábla első fele
- többi elem: minden más egyidejűleg
- 256 bájtos blokkokra van felosztva a különböző nyelvek részére.
- Unicode formátumok: UTF-8, UTF-16, UTF-32



# UTF-8

- változó hosszúságú formátum
- 1-4 bájtot használ (8, 16, 24, ill. 32 bit) a karakterektől függően (pl. az ASCII karaktereket 1 bájton ábrázolja, a magyar nyelvben használatos idézőjeleket 3 bájton)
- Ezt a formátumot használja az IE, a Mozilla, mivel kompatibilis az ASCII kóddal
- a leghatékonyabban használható; sok adatbázis-rendszer, a UNIX rendszer, a Microsoft Windows, és majdnem minden XHTML ezt használja

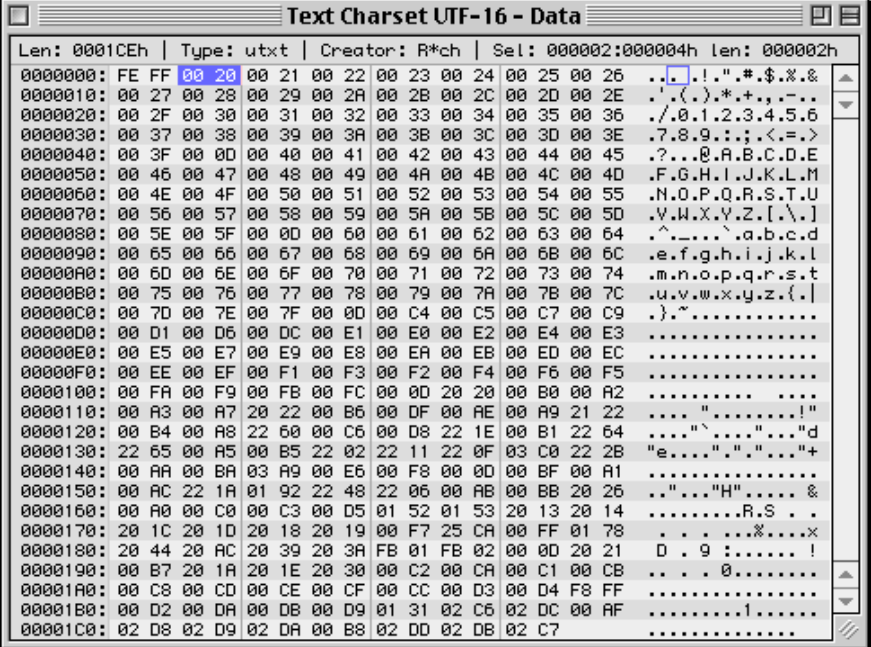


**Linux**



# UTF-16

- változó hosszúságú formátum
- a legtöbb karakter ábrázolása 16 biten történik
- a többi karaktert rendezett párként fejezi ki 16 bit-egységen (32 bit)
- főként karakterfüzérék kezelésének implementációjában használatos



Len: 0001CEh	Type: utxt	Creator: R*ch	Sel: 000002:000004h	Len: 000002h
0000000:	FE FF	00 20	00 21 00 22 00 23 00 24 00 25 00 26	..!."#\$%&
0000010:	00 27 00 28	00 29 00 2A 00 2B 00 2C 00 2D 00 2E	.'(.)*,+,-..	
0000020:	00 2F 00 30	00 31 00 32 00 33 00 34 00 35 00 36	./0.1.2.3.4.5.6	
0000030:	00 37 00 38	00 39 00 3A 00 3B 00 3C 00 3D 00 3E	.7.8.9.:;<.=>	
0000040:	00 3F 00 40	00 41 00 42 00 43 00 44 00 45	.?..@.A.B.C.D.E	
0000050:	00 46 00 47	00 48 00 49 00 4A 00 4B 00 4C 00 4D	.F.G.H.I.J.K.L.M	
0000060:	00 4E 00 4F	00 50 00 51 00 52 00 53 00 54 00 55	.N.O.P.Q.R.S.T.U	
0000070:	00 56 00 57	00 58 00 59 00 5A 00 5B 00 5C 00 5D	.V.W.X.Y.Z.[\.]	
0000080:	00 5E 00 5F	00 60 00 61 00 62 00 63 00 64	^_`...`a.b.c.d	
0000090:	00 65 00 66	00 67 00 68 00 69 00 6A 00 6B 00 6C	.e.f.g.h.i.j.k.l	
00000A0:	00 6D 00 6E	00 6F 00 70 00 71 00 72 00 73 00 74	.m.n.o.p.q.r.s.t	
00000B0:	00 75 00 76	00 77 00 78 00 79 00 7A 00 7B 00 7C	.u.v.w.x.y.z.{.}	
00000C0:	00 7D 00 7E	00 7F 00 80 00 C4 00 C5 00 C7 00 C9	.}~"'"	
00000D0:	00 D1 00 D6	00 DC 00 E1 00 E0 00 E2 00 E4 00 E3	.....	
00000E0:	00 E5 00 E7	00 E9 00 E8 00 EA 00 EB 00 ED 00 EC	.....	
00000F0:	00 EE 00 EF	00 F1 00 F3 00 F2 00 F4 00 F6 00 F5	.....	
0000100:	00 FA 00 F9	00 FB 00 FC 00 0D 20 20 00 B0 00 A2	.....	
0000110:	00 A3 00 A7	20 22 00 B6 00 DF 00 AE 00 A9 21 22	.... "....!"	
0000120:	00 B4 00 A8	22 60 00 C6 00 D8 22 1E 00 B1 22 64	.... "...."...."d	
0000130:	22 65 00 A5	00 B5 22 02 22 11 22 0F 03 C0 22 2B	"e...."...."+	
0000140:	00 AA 00 BA	03 A9 00 E6 00 F8 00 0D 00 BF 00 A1	.....	
0000150:	00 AC 22 1A	01 92 22 48 22 06 00 AB 00 BB 20 26	.... "...."...." &	
0000160:	00 A0 00 C0	00 C3 00 D5 01 52 01 53 20 13 20 14	.....R.S .	
0000170:	20 1C 20 1D	20 18 20 19 00 F7 25 CA 00 FF 01 78	....%....x	
0000180:	20 44 20 AC	20 39 20 3A FB 01 FB 02 00 D0 20 21	D . 9 :..... !	
0000190:	00 B7 20 1A	20 1E 20 30 00 C2 00 CA 00 C1 00 CB	.... 0	
00001A0:	00 C8 00 CD	00 CE 00 CF 00 CC 00 D3 00 D4 F8 FF	.....	
00001B0:	00 D2 00 DA	00 DB 00 D9 01 31 02 C6 02 C0 AF	.....1.....	
00001C0:	02 D8 02 D9	02 DA 00 B8 02 DD 02 DB 02 C7	.....	

# UTF-32

- rögzített hosszúságú formátum, minden karaktert 32 biten ábrázol
- egyelőre ritkán használatos
- ritka karakterek, teljes scriptek kódolására alkalmazzák
- fix hosszúsága miatt könnyen kezelhetők vele a tömbök
- elképzelések szerint az összes élő, halott és mesterséges kultúra írásjeleinek ábrázolására elegendő kb. kétmillió szám ( $2^{21}$ )





Köszönöm a figyelmet  
Szép napot!

