

ARVUSÜSTEEMID

Kõik olulised arvusüsteemid on *positsioonilised* ehk arvu numbrid asuvad neile ettenähtud kindlatel asukohtadel — *arvujärkudes* a_i :

$$\dots a_5 a_4 a_3 a_2 a_1 a_0 a_{-1} a_{-2} a_{-3} a_{-4} \dots a_i \dots$$

Ainus üldtuntud *mittepositsiooniline* arvusüsteem on *rooma numbrite* süsteem numbrimärkidega **I V X L C D M**

arvusüsteemi alus; **järgukaal**

Igal positsioonilisel arvusüsteemil on olemas täisarvuline **alus** p .

Igal järgul a_i on **kaal** p_i , mille saame arvusüsteemi alust p arvujärgu a_i indeksiga i astendades: $p_i = p^i$

järgukaalud: $\dots p^5 p^4 p^3 p^2 p^1 p^0 p^{-1} p^{-2} p^{-3} p^{-4} \dots p^i \dots$

Kui alus $p = 10$, siis on **kümnendsüsteem**, kus järkude kaaludeks on:

$$\begin{array}{ccccccccccc} \dots & 10^3 & 10^2 & 10^1 & 10^0 & 10^{-1} & 10^{-2} & 10^{-3} & \dots \\ \dots & 100 & 10 & 1 & 0.1 & 0.01 & \dots \\ & & & & \text{täisosa} & \bullet & \text{murdosa} \\ & \longleftarrow & & & & & & & \longrightarrow \\ & \text{kõrgemad järgud} & & & & & & & \text{madalamad järgud} \end{array}$$

Koma näitab, kus lähevad täisarvulised järgukaalud üle murdarvulisteks (ehk kus lõpeb *täisosa* ja algab *murdosa*).

Kuigi nimetame *täisosa* ja *murdosa* eraldajat traditsiooniliselt '*komaks*', on levinum tähe märk tema tähistamiseks *punkt* (ingl. *decimal point*).

kõrgemad ja madalamad järgud

suurema kaaluga järke nimetame *kõrgemateks* järkudeks ja väiksema kaaluga järke *madalamateks* järkudeks.

Kuna me ei vaja siin aines *murdarve*, siis *murdarvulise* kaaluga järke me ei kasuta: kõikide edaspidi vaadeldavate arvude madalaima järgu kaal on $p^0 = 1$

numbrite ARV konkreetsetes arvusüsteemides

Igas järgus a_i saab olla p erinevat *numbrimärki* ehk järguväärtust.

Kui $p = 10$, siis $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Igal 10ndnumbril on tema traditsiooniline *väärtus* $0 \dots 9$.

Järgu väärtus on selles arvujärgus asuva *numbri* väärtus.

Arv koosneb *numbritest*.

näide: arv **1024** koosneb neljast *numbrist*: '1' '0' '2' '4'

arvu VÄÄRTUS

Mistahes positsioonilises arvusüsteemis (ehk iga aluse p korral) avaldub arvu **väärtus** N järgneva **korrutiste summana**:

$$N = \dots + a_3 \cdot p^3 + a_2 \cdot p^2 + a_1 \cdot p^1 + a_0 \cdot p^0 + a_{-1} \cdot p^{-1} + a_{-2} \cdot p^{-2} + \dots$$

$\dots$ s.t. **järgukaalud** ja **järguväärtused** on omavahel korrutatud:

$$\begin{array}{ccccccccccc} \dots & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & a_{-1} & a_{-2} & a_{-3} & a_{-4} & \dots & a_i & \dots \\ \dots & p^5 & p^4 & p^3 & p^2 & p^1 & p^0 & p^{-1} & p^{-2} & p^{-3} & p^{-4} & \dots & p^i & \dots \end{array}$$

näide:

10ndsüsteemne arv **123₁₀** (*indeks näitab siin arvusüsteemi*) on väärtusega "sada kaksikümmend kolm" ainult sellepärast, et järgnev tehe annab sellise tulemuse:

$$\mathbf{1 \ 2 \ 3}_{10} = 1 \cdot 100 + 2 \cdot 10 + 3 \cdot 1 = 123_{10}$$

Mõiste "*arvu väärtus*" on eranditult seotud ainult **10ndsüsteemiga**.

10ndsüsteem on kõigi teiste arvusüsteemidega võrreldes tähtsas eristaatuses, kuna inimesed "tunnetavad" arve just 10ndsüsteemis.

"*väärtuse leidmine*" ja "*10ndsüsteemi teisendamine*" on sünonüümid.

Pole olemas "*kahendsüsteemset väärtust*" ega "*kaheksandsüsteemset väärtust*"; on olemas 2ndsüsteemne **esitus** ja 8ndsüsteemne **esitus**.

"kasutamata" arvujärgud a_i on täidetud 0-dega:

$$123.45_{10} = \dots 00000123.450000000 \dots_{10}$$

Täisosa ees ja murdosa järel asuvad '0'-d ($a_i = 0$) ei mõjuta arvu väärtust N :

$$N = \dots + a_3 \cdot p^3 + a_2 \cdot p^2 + a_1 \cdot p^1 + a_0 \cdot p^0 + a_{-1} \cdot p^{-1} + a_{-2} \cdot p^{-2} + \dots$$

(järjestikuste arvude genereerimise / loendamise / inkrementeerimise näide 10ndsüsteemis)

Üleskirjutatud arvu süsteemikuuluvuse täpsustamiseks lisame talle süsteemi näitava indeksi: 372_8 ei ole mitte "kolmsada seitsekümmend kaks" vaid on 8ndsüsteemne arv "kolm-seitse-kaks"

✈ nüüd lahkume 10ndsüsteemist ja siseneme muudesse arvusüsteemidesse



? kuidas saame KAHENDSÜSTEEMI ?

Asendades harjumuspärase arvusüsteemi aluse $p = 10$ uue alusega: $p = 2$ koos kõigi sellega kaasnevate tagajärgedega, saame kahendsüsteemi :

KAHENDSÜSTEEM

Kahendsüsteem on lihtsaim võimalik positsiooniline arvusüsteem:

$$p = 2 \quad a_i \in \{0, 1\}$$

Kuna positsioonilises arvusüsteemis peab olema tema alusega võrdne arv numbrimärke, siis kahendsüsteemsed arvud koosnevad ainult kahest numbrist: 0 ja 1.

Arvusüsteemi aluse muutmisega kaasneb ka järgukaalude muutus, mis kahendsüsteemis on arvu 10 astmete asemel arvu 2 täisarvastmed:

$$\text{2ndsüsteemi järgukaalud: } \dots 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0 \quad 2^{-1} \quad 2^{-2} \quad 2^{-3} \dots$$

$$32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1 \quad 0.5 \quad 0.25 \quad 0.125$$

(järjestikuste arvude genereerimise / loendamise / inkrementeerimise näide 2ndsüsteemis)

Järgnevalt on loetletud kõik kuni 6-järgulised kahendarvud (ehk 2ndarvud väärtusega 0 kuni 63):

$0_2 = 0_{10}$	$10000_2 = 16_{10}$	$100000_2 = 32_{10}$	$110000_2 = 48_{10}$
$1_2 = 1_{10}$	$10001_2 = 17_{10}$	$100001_2 = 33_{10}$	$110001_2 = 49_{10}$
$10_2 = 2_{10}$	$10010_2 = 18_{10}$	$100010_2 = 34_{10}$	$110010_2 = 50_{10}$
$11_2 = 3_{10}$	$10011_2 = 19_{10}$	$100011_2 = 35_{10}$	$110011_2 = 51_{10}$
$100_2 = 4_{10}$	$10100_2 = 20_{10}$	$100100_2 = 36_{10}$	$110100_2 = 52_{10}$
$101_2 = 5_{10}$	$10101_2 = 21_{10}$	$100101_2 = 37_{10}$	$110101_2 = 53_{10}$
$110_2 = 6_{10}$	$10110_2 = 22_{10}$	$100110_2 = 38_{10}$	$110110_2 = 54_{10}$
$111_2 = 7_{10}$	$10111_2 = 23_{10}$	$100111_2 = 39_{10}$	$110111_2 = 55_{10}$
$1000_2 = 8_{10}$	$11000_2 = 24_{10}$	$101000_2 = 40_{10}$	$111000_2 = 56_{10}$
$1001_2 = 9_{10}$	$11001_2 = 25_{10}$	$101001_2 = 41_{10}$	$111001_2 = 57_{10}$
$1010_2 = 10_{10}$	$11010_2 = 26_{10}$	$101010_2 = 42_{10}$	$111010_2 = 58_{10}$
$1011_2 = 11_{10}$	$11011_2 = 27_{10}$	$101011_2 = 43_{10}$	$111011_2 = 59_{10}$
$1100_2 = 12_{10}$	$11100_2 = 28_{10}$	$101100_2 = 44_{10}$	$111100_2 = 60_{10}$
$1101_2 = 13_{10}$	$11101_2 = 29_{10}$	$101101_2 = 45_{10}$	$111101_2 = 61_{10}$
$1110_2 = 14_{10}$	$11110_2 = 30_{10}$	$101110_2 = 46_{10}$	$111110_2 = 62_{10}$
$1111_2 = 15_{10}$	$11111_2 = 31_{10}$	$101111_2 = 47_{10}$	$111111_2 = 63_{10}$

$$N = \dots + a_3 \cdot p^3 + a_2 \cdot p^2 + a_1 \cdot p^1 + a_0 \cdot p^0 + a_{-1} \cdot p^{-1} + a_{-2} \cdot p^{-2} + \dots$$

😊😊 arvu väärtuse N leidmine osutub 2ndarvude jaoks eriti lihtsaks:

Kuna kahendarvudes ei leidu suuremaid järguväärtusi kui 1, siis kahendarvude korral arvu väärtust arvutav avaldis $N = \dots$

(ehk teisendus 10ndsüsteemi) lihtsustub :

korrumisest saab loobuda ja summeerida tuleb ainult need järgukaalud, kus asub järguväärtus 1 :

näide:

$$1110_2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 =$$

$$= 8 + 4 + 2 + 0 = 14_{10}$$



? kas väiksed kahendarvud peaks pähe õppima ?

kahendarve pole mõtet "pildina" pähe õppida, kuna "väikseid" kahendarve saab hetkega tuletada **järgukaalude kaudu** (täites vajalikud järgud 1-dega)



võtame veel ühe suvalise 2ndarvu eelnevast tabelist ja leiame ta väärtuse:

$101_2 = 5_{10}$	$10101_2 = 21_{10}$	$100101_2 = 37_{10}$	$110101_2 = 53_{10}$
$1010_2 = 10_{10}$	$11010_2 = 26_{10}$	$101010_2 = 42_{10}$	$111010_2 = 58_{10}$
$1011_2 = 11_{10}$	$11011_2 = 27_{10}$	$101011_2 = 43_{10}$	$111011_2 = 59_{10}$
$1100_2 = 12_{10}$	$11100_2 = 28_{10}$	$101100_2 = 44_{10}$	$111100_2 = 60_{10}$
$1101_2 = 13_{10}$	$11101_2 = 29_{10}$	$101101_2 = 45_{10}$	$111101_2 = 61_{10}$
$1110_2 = 14_{10}$	$11110_2 = 30_{10}$	$101110_2 = 46_{10}$	$111110_2 = 62_{10}$
$1111_2 = 15_{10}$	$11111_2 = 31_{10}$	$101111_2 = 47_{10}$	$111111_2 = 63_{10}$

$$101011_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 32 + 8 + 2 + 1 = 43_{10}$$

ülesanne:



Leida järgnevate positiivsete 2ndarvude väärtus (ehk teisendada 10ndsüsteemi)

$$\begin{aligned} 101_2 &= \dots_{10} \\ 0110_2 &= \dots_{10} \\ 0001101_2 &= \dots_{10} \\ 000010011_2 &= \dots_{10} \\ 010001011_2 &= \dots_{10} \\ 000000101100_2 &= \dots_{10} \end{aligned}$$



$$\begin{aligned} 101_2 &= 5_{10} \\ 0110_2 &= 6_{10} \\ 0001101_2 &= 13_{10} \end{aligned}$$

$$\begin{aligned} 000010011_2 &= 19_{10} \\ 010001011_2 &= 139_{10} \\ 000000101100_2 &= 44_{10} \end{aligned}$$



? ... aga vastupidine teisendus? Kuidas viime 10ndarvu 2ndkujule?

Teisendus 10ndsüsteemist 2ndsüsteemi

Teisendus ühest arvusüsteemist teise toimub uue alusega jagamise teel kus jagamine on täisarvuline: murdarvu asemel saame jagatise ja jäägi:

$$7 : 2 = 3 \text{ (jääk 1)}$$

Väärtuse N leidmise suhtes vastupidine teisendus ehk 10ndsüsteemse täisarvu teisendamine 2ndsüsteemi toimub 2-ga jagamise teel, kusjuures (täisarvulise) jagamise jäägid (0 ja 1) on saadava 2ndarvu järguväärtusteks.

näide:

Teisendame 10ndtäisarvud 37_{10} 56_{10} 109_{10} 2ndkujule:

$$\begin{array}{r|l} :2 & \\ 37 & 1 \text{ (madalaim järk)} \\ 18 & 0 \\ 9 & 1 \\ 4 & 0 \\ 2 & 0 \\ 1 & 1 \text{ (kõrgeim järk)} \\ 0 & \end{array}$$

$$\begin{aligned} 37_{10} &= 100101_2 \\ 37 &= 32 + 4 + 1 \end{aligned}$$

$$\begin{array}{r|l} :2 & \leftarrow \text{jagaja} \\ 56 & 0 \\ 28 & 0 \\ 14 & 0 \\ 7 & 1 \text{ (jääk)} \\ 3 & 1 \text{ (jagatav)} \\ 1 & 1 \text{ (jagatis)} \\ 0 & \end{array}$$

$$(3 \times 2) + 1 = 7$$

$$\begin{array}{r|l} :2 & \\ 109 & 1 \\ 54 & 0 \\ 27 & 1 \\ 13 & 1 \\ 6 & 0 \\ 3 & 1 \\ 1 & 1 \\ 0 & \end{array}$$

$$(27 \times 2) + 0 = 54$$

$$37_{10} = 100101_2 \quad 56_{10} = 111000_2 \quad 109_{10} = 1101101_2$$

$$1\ 2\ 3_8 = \dots 10$$



$$\begin{aligned}
 25_8 &= 2 \times 8^1 + 5 \times 8^0 = 16 + 5 = 21_{10} \\
 74_8 &= 7 \times 8^1 + 4 \times 8^0 = 56 + 4 = 60_{10} \\
 123_8 &= 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 = 83_{10}
 \end{aligned}$$

Teisendus 10ndsüsteemist 8ndsüsteemi

10ndtäisarvude teisendus **8ndsüsteemi** toimub **8**-ga jagamise teel, kusjuures igal jagamissammul saadakse jäägina arvu järgmine **8ndnumber** **0...7**.

✓ ülesanne: _____



Teisenda 10ndtäisarvud **83₁₀** **106₁₀** 8ndkujule:



$$\begin{array}{r|l}
 : 8 & \\
 106 & 2 \\
 13 & 5 \\
 1 & 1 \\
 0 &
 \end{array}$$

$$106_{10} = \overset{64}{1} \overset{8}{5} \overset{1}{2}_8 \leftarrow \text{järgukaalud}$$

(kontrollimisvõimalus)

KUUETEISTKÜMNENDSÜSTEEM

hexadecimal (hex)

Kuna **16ndsüsteemis** on arvusüsteemi alus **p = 16**, siis peab seal olema ka 16 võimalikku järguväärtust ja sellest tulenevalt ka **16 numbrimärki** nende esitamiseks.

probleem: araabia numbraid (**0...9**) on olemas ainult **10 tk !?**
6 numbrit jääb puudu ?



lahendus:

Lisaks 10-le araabia numbrile **0...9** on ülejäänud kuueks numbrimärgiks võetud ladina tähestiku algustähed **A...F**:



p = 16 $a_i \in \{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ A\ B\ C\ D\ E\ F\}$

16ndnumbrid A...F omavad väärtusi:

A = 10 **B = 11** **C = 12** **D = 13** **E = 14** **F = 15**

16ndsüsteemi järgukaalud: ... **16⁴** **16³** **16²** **16¹** **16⁰** **16⁻¹** **16⁻²** ...
 65536 4096 256 16 1 0.0625

.... oleme märganud, et järk kaaluga **1** on olemas igas arvusüsteemis:

p = 10:		10000	1000	100	10	1
p = 2:		16	8	4	2	1
p = 8:		4096	512	64	8	1
p = 16:		65536	4096	256	16	1



..... **p⁵** **p⁴** **p³** **p²** **p¹** **p⁰** **p⁻¹** **p⁻²** **p⁻³** **p⁻⁴** **pⁱ**
p⁰ = 1 suvalise **p** korral

✓ ülesanne: _____



Teisenda järgnevad 16ndarvud **10ndkujule** (ehk leia väärtus):

$$12_{16} = \dots 10$$

$$4D_{16} = \dots 10$$

$$A6_{16} = \dots 10$$

$$FF_{16} = \dots 10$$

$$100_{16} = \dots 10$$

$$1CD_{16} = \dots 10$$



$$\begin{aligned}
 12_{16} &= 1 \times 16^1 + 2 \times 16^0 = 18_{10} \\
 4D_{16} &= 4 \times 16^1 + 13 \times 16^0 = 77_{10} \\
 A6_{16} &= 10 \times 16^1 + 6 \times 16^0 = 166_{10} \\
 FF_{16} &= 255_{10} \\
 100_{16} &= 256_{10} \\
 1CD_{16} &= 461_{10}
 \end{aligned}$$

16ndsüsteem on "suurim" praktiliselt kasutatav arvusüsteem.



Võimalik on koostada arvusüsteeme ka suurema alusega kui 16, kuid selliseid suuremaid arvusüsteeme pole vaja.

Teisendus 10ndsüsteemist 16ndsüsteemi

10ndtäisarvude teisendus 16ndsüsteemi toimub 16-ga jagamise teel, kusjuures igal jagamissammul saadakse jäägina arvu järgmine 16ndnumber 0 F.

ülesanne:



Teisendada järgnevad 10ndarvud 16ndkujule :

$$\begin{aligned}
 77_{10} &= \dots_{16} \\
 32_{10} &= \dots_{16} \\
 256_{10} &= \dots_{16}
 \end{aligned}$$



$$\begin{aligned}
 77_{10} &= 4D_{16} \\
 32_{10} &= 20_{16} \\
 256_{10} &= 100_{16}
 \end{aligned}$$

16ndsüsteem on vajalik meie kodutöö alguses ! (detsembris)



Edaspidi vajame nendest arvusüsteemidest kõige rohkem **kahendsüsteemi**

ARVUSÜSTEEMID

kokkuvõttev loetelu

kuni arvutite ilmumiseni polnud vaja muid arvusüsteeme peale 10ndsüsteemi

$$2 \leq p \leq 16$$

- 2ndsüsteem:** $p = 2$ $a_i \in \{0, 1\}$
- 3ndsüsteem:** $p = 3$ $a_i \in \{0, 1, 2\}$
- 4ndsüsteem:** $p = 4$ $a_i \in \{0, 1, 2, 3\}$
- 5ndsüsteem:** $p = 5$ $a_i \in \{0, 1, 2, 3, 4\}$
- 6ndsüsteem:** $p = 6$ $a_i \in \{0, 1, 2, 3, 4, 5\}$
- 7ndsüsteem:** $p = 7$ $a_i \in \{0, 1, 2, 3, 4, 5, 6\}$
- 8ndsüsteem:** $p = 8$ $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$
- 9ndsüsteem:** $p = 9$ $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$
- 10ndsüsteem:** $p = 10$ $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- 11ndsüsteem:** $p = 11$ $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A\}$
- 12ndsüsteem:** $p = 12$ $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B\}$
- 13ndsüsteem:** $p = 13$ $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C\}$
- 14ndsüsteem:** $p = 14$ $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D\}$
- 15ndsüsteem:** $p = 15$ $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E\}$
- 16ndsüsteem:** $p = 16$ $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

olulised arvusüsteemid: $p = 2 = 2^1$ (kus $p = 2^n$)

$$p = 8 = 2^3$$

$$p = 10$$

$$p = 16 = 2^4$$

... muid arvusüsteeme praktikas ei kasutatagi ...

Kuna kahendnumbrid **0** ja **1** on kasutusel ka loogikaväärtuste tähistustena, siis leiab **kahendsüsteem** rakendust ka lausearvutuses, loogikaalgebras ja kõikjal mujal, kus tegeletakse **1**-de ja **0**-de kogumikega ehk kahendkoodidega.

Teisendus 2ndsüsteemist 8ndsüsteemi või 16ndsüsteemi

siin on loetelu arvude **0 ... 15** esitustega neljas erinevas arvusüsteemis :

10nd	16nd	2nd	8nd	2nd
0	0	0000	0	000
1	1	0001	1	001
2	2	0010	2	010
3	3	0011	3	011
4	4	0100	4	100
5	5	0101	5	101
6	6	0110	6	110
7	7	0111	7	111
8	8	1000	10	
9	9	1001	11	
10	A	1010	12	
11	B	1011	13	
12	C	1100	14	
13	D	1101	15	
14	E	1110	16	
15	F	1111	17	

2ndsüsteemi, **8ndsüsteemi** ja **16ndsüsteemi** alused on arvu 2 täisarvastmed: $p = 2^1$ $p = 2^3$ $p = 2^4$.

See annab neile kasuliku lisaomaduse, võimaldades nende süsteemide omavahelisi arvuteisendusi teha ka **numbrimärkide asendamise teel** ehk ilma "uue alusega" jagamata.



?... kuidas saame **2ndarvu** viia **8ndkujule** ?

2ndarvu on võimalik kiirelt teisendada (ümber kirjutada) tema **8ndkujule**, asendades (alates 2ndarvu madalamatest järkudest)

iga tema järkudekolmiku **000 ... 111** vastava **8ndnumbriga** **0 ... 7** nagu näitas eeltoodud vastavustabel :

10nd	16nd	2nd	8nd	2nd
0	0	0000	0	000
1	1	0001	1	001
2	2	0010	2	010
3	3	0011	3	011
4	4	0100	4	100
5	5	0101	5	101
6	6	0110	6	110
7	7	0111	7	111
8	8	1000	10	
9	9	1001	11	
10	A	1010	12	
11	B	1011	13	
12	C	1100	14	
13	D	1101	15	
14	E	1110	16	
15	F	1111	17	

✓ näide:

Võtame suvalise 2ndarvu: **1 0 1 1 0 1 0 1 0 0 1 1 1**₂ eesmärgiga viia see arv **8ndkujule** ja seejärel ka **16ndkujule**.

viimine / ümberkirjutamine **8ndkujule** :

võimalik oleks teisendada **2nd** → **10nd** → **8nd** kuid see oleks asjatu töö
parim teisendusviis: Grupeerime **2nd**arvu järgud 3-järgulistesse gruppidesse
alates madalamatest järkudest :

1 0 1 1 0 1 0 1 0 0 1 1 1₂

..... lisades vajadusel arvu ette 0-lle:

0 0 1 | 0 1 1 | 0 1 0 | 1 0 0 | 1 1 1

.... edasi asendame 2ndjärkude iga grupeeritud kolmiku temaga
väärtuselt võrdse **8nd**numbriga 0....7 :
(eelpoolses vastavustabelis read: **000** kuni **111**; **0** kuni **7**)

1 3 2 4 7
001|011|010|100|111

seega 1011010 100111₂ = 13247₈

! viga:



alates **kõrgematest** järkudest ei tohi grupeerida :

101|101|010|011|1



?... miks ei tohi grupeerida 2ndjärke alates **kõrgematest** järkudest ?

... kuna **täisarvu lõppu** ei tohi lisada **0-lle** — see "rikuks" arvu väärtuse !



sama **2nd**arvu viimine / ümberkirjutamine **16nd**kujule :

Selleks grupeerime **2nd**arvu järgud 4 järgu kaupa alates madalamatest
järkudest ja
asendame iga 2ndjärkude neliku temaga väärtuselt võrdse **16nd**numbriga
0....F :

juhindudes vastavustabelist **2ndsüsteemi** ja **16ndsüsteemi** vahel:

10nd	16nd	2nd	8nd	2nd
------	------	-----	-----	-----

0	0	0000	0	000
1	1	0001	1	001
2	2	0010	2	010
3	3	0011	3	011
4	4	0100	4	100
5	5	0101	5	101
6	6	0110	6	110
7	7	0111	7	111
8	8	1000	10	
9	9	1001	11	
10	A	1010	12	
11	B	1011	13	
12	C	1100	14	
13	D	1101	15	
14	E	1110	16	
15	F	1111	17	

1 0 1 1 0 1 0 1 0 0 1 1 1₂

1 6 A 7
0001|0110|1010|0111

seega 1011010 100111₂ = 16A7₁₆

ülesanne:



Kontrollida eelnevalt leitud arvude võrdsust, leides eelmise näite 2ndarvu, 8ndarvu ja 16ndarvu väärtused :

$$1011010100111_2 = \dots_{10}$$

$$13247_8 = \dots_{10}$$

$$16A7_{16} = \dots_{10}$$



arvestades järgukaalusid nendes arvusüsteemides :

$$1011010100111_2 = 2^{12} + 2^{10} + 2^9 + 2^7 + 2^5 + 2^2 + 2^1 + 2^0 = \dots_{10}$$

$$1011010100111_2 = 2^{12} + 2^{10} + 2^9 + 2^7 + 2^5 + 2^2 + 2^1 + 2^0 = 5799_{10}$$

$$13247_8 = 1 \times 8^4 + 3 \times 8^3 + 2 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 = \dots_{10}$$

$$13247_8 = 1 \times 8^4 + 3 \times 8^3 + 2 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 = 5799_{10}$$

$$16A7_{16} = 1 \times 16^3 + 6 \times 16^2 + 10 \times 16^1 + 7 \times 16^0 = \dots_{10}$$

$$16A7_{16} = 1 \times 16^3 + 6 \times 16^2 + 10 \times 16^1 + 7 \times 16^0 = 5799_{10}$$

(kõik need 3 arvu on võrdsed)

sellise asendamisega saab teisendada ka "vastupidises suunas" :

numbrite vastupidise asendamisega kahendjärkude kolmikuteks või nelikuteks saab arvu 8ndkujult või 16ndkujult kergesti üle minna tema

2ndkujule: 8nd $\longrightarrow$ 2nd 16nd $\longrightarrow$ 2nd



?... aga 4ndsüsteem ? see on ju kah alusega $p = 2^n$?

4ndsüsteem (ebaoluline)

Sarnast arvujärkude asendamist saab rakendada ka 4ndsüsteemiga tegeledes, sest arvusüsteemi alus $p = 4 = 2^2$.

4ndsüsteemi ja 2ndsüsteemi siduv vastavustabel oleks:

2nd	4nd
00 ₂	0 ₄
01 ₂	1 ₄
10 ₂	2 ₄
11 ₂	3 ₄

01001011₂
seega 4ndsüsteemi
ümberkirjutamiseks
vaja grupeerida :
01|00|10|11

4ndsüsteem ei ole oluline arvusüsteem ja praktikas teda ei kasutata.



?... aga kui grupeerida 2ndarvu järgud ühekaupa ?

1|0|1|1|0|1|0|1|0|0|1|1|1



... siis saame mõttetu "teisenduse" 2ndsüsteemist ... 2ndsüsteemi !

Programmeerimiskeeled võimaldavad arvude ("arvkonstantide") esitamist 10ndsüsteemis

2ndsüsteemis

8ndsüsteemis

16ndsüsteemis

..... kuid mitte 4ndsüsteemis.



?... kas on võimalik 8ndarv ümberkirjutada 16ndkujule ka "otse" ehk ilma 2ndsüsteemi "lääbimata" ?

! 8ndsüsteemi ja 16ndsüsteemi vaheline "otseteisendus" (numbrite asenduse teel) pole võimalik: ainult 2ndsüsteemi kaudu saab.



13247₈ $\longleftrightarrow$ 16A7₁₆
OTSE ei saa

saab nii : 8nd $\longleftrightarrow$ 2nd $\longleftrightarrow$ 16nd

16ndsüsteemi tähtsus

Arvutimälus hoitakse andmeid **baitides** (*byte*), mis on **8**-järgulised kahendkoodid.

vaatleme kõikvõimalikke **2nd**koode mis saavad olla *baidis* :

00000000₂

00000001₂

00000010₂

00000011₂

.

.

01111001₂

01111010₂

01111011₂

01111100₂

.

.

11111100₂

11111101₂

11111110₂

11111111₂

16ndsüsteem võimaldab esitada / näidata baitide sisu

(ja üldse igasuguseid kahendkoode) palju kompaktsemalt võrreldes nende "vahetu" esitamisega kahendkujul.

Selleks jaotame *baidi* kõrgemaks ja madalamaks **poolbaidiks** :

00000000₂

00000001₂

00000010₂

00000011₂

.

.

01111001₂

01111010₂

01111011₂

.

.

11111101₂

11111110₂

11111111₂

Mõlema *poolbaidi* saab asendada vastava **16ndnumbriga** 0 F :

00000000₂ = 00₁₆

00000001₂ = 01₁₆

00000010₂ = 02₁₆

00000011₂ = 03₁₆

.

.

01111001₂ = ??₁₆

01111010₂ = ??₁₆

01111011₂ = 7B₁₆

.

01111111₂ = ??₁₆

.

$$10001001_2 = 89_{16}$$

$$10001010_2 = 8A_{16}$$

$$253_{10} = 11111101_2 = FD_{16} = 253_{10}$$

$$11111110_2 = FE_{16}$$

$$11111111_2 = FF_{16}$$

Baidi mistahes võimalikku sisu / koodi saab seega esitada
kahejärgulise 16ndarvuna: (suvalised juhuslikud näitebaidid)

$$10110111_2 = B7_{16}$$

$$00000101_2 = 05_{16}$$

$$11100100_2 = E4_{16}$$

$$01101010_2 = 6A_{16}$$

$$11111110_2 = FE_{16}$$

$$11111111_2 = FF_{16}$$

Kui arvutimälu sisu tuleb kuidagi visuaalselt (inimestele) näidata, siis eelistatakse mälus tegelikult asuvate 1-de ja 0-de näitamise asemel esitada mälubaitides asuvate 2ndarvudega võrdseid 16ndarve.

16ndsüsteemi kasutatakse **2ndarvude** kompaktsemaks esitamiseks



(16ndsüsteem on vajalik ka kodutöös)



?... aga kui bait oleks kehtestatud (näiteks) 6-järgulisena ?

.... siis oleks 8ndsüsteem olnud olulisem kui 16ndsüsteem, kuna poolbait oleks olnud seljuhul 3-järguline :

$$101011_2 = \dots_8$$

$$101|011_2 = 53_8$$

Kahendkoodidega seotud mõisted

♦ (*n*-järguline) **kahendvektor** on kahendnumbritena 0 ja 1 esitatud loogikaväärtuste ühemõõtmeline jada pikkusega *n*.

Vektori **pikkus** on tema 2ndjärkude arv ehk *n*-järgulise 2ndvektori pikkus on *n*.

— näide: —

Järgneval real on esitatud 6 erineva pikkusega **kahendvektoreid** :

00101101	010	11011	10	1	000101
----------	-----	-------	----	---	--------

Kahendvektoril pole mingit seost füüsikast tuntud vektori mõistega.

Erinevalt kahendarvust ei tohi kahendvektoris ära jätta algusnulle:

$$000101 \neq 101$$

Kahendvektori järkudel pole järgukaalu. Tema sarnasuse tõttu 2ndarvudega osutub mõnes rakenduses siiski kasulikuks ja vajalikuks vaadelda teda kahendarvuna ehk 2ndvektori järkudele omistatakse vajadusel 2ndsüsteemi loomulikud järgukaalud :

$$\dots 16 \ 8 \ 4 \ 2 \ 1$$

See võimaldab kahendvektorit kompaktsemalt esitada talle vastava 2ndarvu väärtuse abil.

♦ **lähisvektorid** (lähiskoodid) on võrdse pikkusega kahendvektorid, mis erinevad teineteisest ainult ühes kahendjärgus.

näide: järgnevad 2 vektorit on teineteise lähisvektorid : 1011 1001

näide: ka need kaks 2ndvektorit on teineteise lähisvektorid :

$$1101011010$$

$$1101001010$$

Iga *n*-järguline kahendvektor omab seega *n* tk. lähisvektoreid.

♦ **intervall** on võrdse pikkusega kahendvektorite hulk võimsusega 2^n ($n \in \mathbb{N}$), milles iga hulgaelemendi jaoks leidub samas hulgas täpselt *n* lähisvektorit.

— näide: —