ARVUSÜSTEEMID

Kõik olulised arvusüsteemid on *positsioonilised* ehk arvu numbrid asuvad neile ettenähtud kindlatel asukohtadel — *arvujärkudes* $\mathbf{a}_i$:

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

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

Rooma numbrite puudus : ei ole võimalik paberil käsitsi summeerida arve :

  127.9
+ 36.8

arvusüsteemi **alus** ; järgu**kaal**

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

Igal järgul $\mathbf{a}_i$ on **kaal** $\mathbf{p}_i$, mille saame arvusüsteemi alust **p** arvujärgu $\mathbf{a}_i$ indeksiga i astendades: $\mathbf{p}_i = \mathbf{p}^i$

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

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

$$\ldots\ 10^3\ \ 10^2\ \ 10^1\ \ 10^0\ \ 10^{-1}\ \ 10^{-2}\ \ 10^{-3}\ \ldots$$

$$\ldots\ 100\ \ 10\ \ 1\ \bullet\ 0.1\ \ 0.01\ \ldots$$

täisosa *murdosa*

← →

kõrgemad järgud **madalamad järgud**
vanemad järgud *nooremad järgud*

Koma näitab, kus lähevad täisarvulised järgukaalud üle murdarvulisteks (ehk kus lõppeb *täisosa* ja algab *murdosa*).
Kuigi nimetame täisosa ja murdosa eraldajat traditsiooniliselt '*komaks*', on levinum tähemärk 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 $\mathbf{p}^0 = \mathbf{1}$

numbrite ARV konkreetses arvusüsteemis

Igas järgus $\mathbf{a}_i$ saab olla **p** erinevat *numbrimärki* ehk järguväärtust.

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

Igal 10ndnumbril on tema traditsiooniline *väärtus* 0 ..... 9.
Järgu väärtus on selles arvujärgus asuva *numbri* väärtus.
arv koosneb *numbritest.*
näide: arv **1024** koosneb neljast *numbrist:* '**1**' '**0**' '**2**' '**4**'

arvu VÄÄRTUS

$$\ldots\ a_5\ \ a_4\ \ a_3\ \ a_2\ \ a_1\ \ a_0\ \ a_{-1}\ \ a_{-2}\ \ a_{-3}\ \ a_{-4}\ \ldots\ a_i\ \ldots$$
$$\ldots\ p^5\ \ p^4\ \ p^3\ \ p^2\ \ p^1\ \ p^0\ \ p^{-1}\ \ p^{-2}\ \ p^{-3}\ \ p^{-4}\ \ldots\ p^i\ \ldots$$

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

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

.... s.t. iga **järguväärtus** on korrutatud oma **järgukaaluga** + kokkuliidetud

näide:

10ndsüsteemne arv $\mathbf{123}_{10}$ (*indeks näitab siin arvusüsteemi*) on väärtusega "*sada kakskümmend kolm*" ainult sellepärast, et järgnev tehe annab sellise tulemuse:

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

mõiste "*arvu väärtus*" on ainult **10nd**süsteemne

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 $\mathbf{a}_i$ on täidetud 0-dega:

$\mathbf{123.45}_{10}$ = ... 00000**123.45**0000000 . . . . $_{10}$

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

$$\mathbf{N} = \ldots + \mathbf{a}_3 \cdot \mathbf{p}^3 + \mathbf{a}_2 \cdot \mathbf{p}^2 + \mathbf{a}_1 \cdot \mathbf{p}^1 + \mathbf{a}_0 \cdot \mathbf{p}^0 + \mathbf{a}_{-1} \cdot \mathbf{p}^{-1} + \mathbf{a}_{-2} \cdot \mathbf{p}^{-2} + \ldots$$

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

. . . mida tähendab **i n k r e m e n t** / **inkrementeerimine** ?

Tüvenumbrid

Arvu *tüvenumbrid* on arvu numbrid alates <u>kõrgeimast mittenullisest</u> numbrist kuni <u>madalaima mittenullise</u> numbrini.

Kuigi madalaim ja kõrgeim tüvenumber pole kumbki 0 , võivad nende "vahel" olla tüvenumbriteks ka '0'-d.

näide: arvus **0.0000120003000** on tüvenumbriteks **120003** .

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

✈ *nüüd lahkume* ***10nd****süsteemist ja siseneme muudesse arvusüsteemidesse*

? kuidas saame KAHENDSÜSTEEMI ?

Asendades eelvaadatud "arvusüsteemide universaalses üldformaadis" harjumuspärase arvusüsteemi aluse **p = 10** uue alusega: **p = 2** koos kõigi sellega kaasnevate tagajärgedega, saame **kahendsüsteemi** :

KAHENDSÜSTEEM

Kahendsüsteem on lihtsaim võimalik positsiooniline arvusüsteem:

$$\mathbf{p} = \mathbf{2} \qquad \mathbf{a}_i \in \{0, 1\}$$

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

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

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

2ndsüsteemi **järgukaalud**:	. . .	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	. . .
		32	**16**	**8**	**4**	**2**	**1**	**0.5**	**0.25**	**0.125**	

. . . siin õppeaines me ei vaja **murdosa** järke . . .

(. . . *järjestikuste arvude genereerimise / loendamise / inkrementeerimise näide* ***2nd****süsteemis . . . .*)

Järgnevalt on loetletud / "genereeritud" kõik kuni **6**-järgulised kahendarvud (ehk **2nd**arvud väärtusega **0** kuni **63** — **2nd**arvude *inkrement*-jada) :

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

. . . kuidas saab leida olemasoleva **2nd**arvu **väärtuse** ehk **10nd**kuju?

Ka **2nd**arvu väärtus arvutub sellesama eelnäidatud **korrutiste summana** :

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

kuid

☺ ☻ arvu **väärtuse N** leidmine osutub **2nd**arvude jaoks eriti lihtsaks:

Kuna kahendarvudes ei leidu suuremaid järguväärtusi kui **1**, siis kahendarvude korral arvu väärtust arvutav avaldis **N = ...** (ehk teisendus **10nd**süsteemi) **lihtsustub** :
korrutamisest saab loobuda ja **summeerida tuleb ainult need järgukaalud, kus asub järguväärtus 1**

näide:

vaatleme mingit juhuslikku 4-järgulist 2ndarvu : $\mathbf{1\,1\,1\,0}_2$
. . . ja soovime teada, kui suur arv see on ? (väärtus ?)

4-järgulise täisarvu korral asub arv <u>nendes</u> *järkudes* $\mathbf{a}_i$:

$a_3 \quad a_2 \quad a_1 \quad a_0$ . . . . ja nende järkude **k a a l u d** on :
$p^3 \quad p^2 \quad p^1 \quad p^0$

2ndnumbrid on ainult **0** ja **1** — mis juhtuvad mõlemad olema aritmeetikas väga "mugavad" korrutajad **!** :

$$\mathbf{1\,1\,1\,0}_2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 =$$
$$= \mathbf{8} + \mathbf{4} + \mathbf{2} + \mathbf{0} = \mathbf{14}_{10}$$

. . . seega leidsime / kontrollisime järgu**kaalude** ja järgu**väärtuste** kaudu, et :

$$\mathbf{1\,1\,1\,0}_2 = \mathbf{14}_{10}$$

. . . ka eelnev (järjestikuste) **2nd**arvude loetelu näitas, et 2ndarv **1110** "tekkis" *nullist* startides i n k r e m e n t e e r i m i s e l *viieteistkümnendana* — ehk tekkis arvuks väärtusega **14** (kuna jadas "esimesena" "genereerub" arv *null* **0**) :

$0000_2 = 0_{10}$			
. . . .			
$0111_2 = 7_{10}$	$10111_2 = 23_{10}$	$100111_2 = 39_{10}$	$110111_2 = 55_{10}$
$1000_2 = 8_{10}$	$11000_2 = 24_{10}$	$101000_2 = 40_{10}$	$111000_2 = 56_{10}$
$1001_2 = 9_{10}$	$11001_2 = 25_{10}$	$101001_2 = 41_{10}$	$111001_2 = 57_{10}$
$1010_2 = 10_{10}$	$11010_2 = 26_{10}$	$101010_2 = 42_{10}$	$111010_2 = 58_{10}$
$1011_2 = 11_{10}$	$11011_2 = 27_{10}$	$101011_2 = 43_{10}$	$111011_2 = 59_{10}$
$1100_2 = 12_{10}$	$11100_2 = 28_{10}$	$101100_2 = 44_{10}$	$111100_2 = 60_{10}$

$1101_2 = 13_{10}$	$11101_2 = 29_{10}$	$101101_2 = 45_{10}$	$111101_2 = 61_{10}$
$1110_2 = 14_{10}$	$11110_2 = 30_{10}$	$101110_2 = 46_{10}$	$111110_2 = 62_{10}$
$1111_2 = 15_{10}$	$11111_2 = 31_{10}$	$101111_2 = 47_{10}$	$111111_2 = 63_{10}$

? 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 ***2nd****arvu eelnevast tabelist ja leiame ta* ***väärtuse*** :

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

$$\mathbf{1\,0\,1\,0\,1\,1}_2 = ?_{10}$$

$$\mathbf{1\,0\,1\,0\,1\,1}_2 = 1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 =$$

$$= \mathbf{32} + \mathbf{8} + \mathbf{2} + \mathbf{1} = \mathbf{43}_{10}$$

ülesanne:

Leida järgnevate positiivsete 2ndarvude **väärtus**

(ehk teisendada 2ndarvud 10ndsüsteemi)

$101_2 = \ldots_{10}$

$0110_2 = \ldots_{10}$

$0001101_2 = \ldots_{10}$

$000010011_2 = \ldots_{10}$

$010001011_2 = \ldots_{10}$

$000000101100_2 = \ldots_{10}$

$101_2 = \mathbf{5}_{10}$

$0110_2 = \mathbf{6}_{10}$

$0001101_2 = \mathbf{13}_{10}$

$000010011_2 = \mathbf{19}_{10}$

$010001011_2 = \mathbf{139}_{10}$

$000000101100_2 = \mathbf{44}_{10}$

? . . . aga vastupidine teisendus ? Kuidas viime **10nd**arvu **2nd**kujule ?

Teisendus 10ndsüsteemist 2ndsüsteemi

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

$$\mathbf{7} : \mathbf{2} = \mathbf{3} \quad (\text{jääk } \mathbf{1})$$

jagatav : jagaja = jagatis (jääk)

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

näide:

Teisendame 10ndtäisarvud **37**$_{10}$ **56**$_{10}$ **109**$_{10}$ **2nd**kujule:

: 2		
37	**1**	↑ *madalaim järk*
18	**0**	
9	**1**	
4	**0**	
2	**0**	
1	**1**	*kõrgeim järk*
0		

$$37_{10} = \overset{32}{1}\,0\,0\,\overset{4}{1}\,0\,\overset{1}{1}_{\,2}$$

$$37 = 32 + 4 + 1$$

	: 2 ← *jagaja*		
	56	**0**	
	28	**0**	
	14	**0**	
jagatav →	7	**1**	← *jääk*
jagatis →	3	**1**	
	1	**1**	
	0		

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

: 2	
109	**1**
54	**0**
27	**1**
13	**1**
6	**0**
3	**1**
1	**1**
0	

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

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

? . . . kuidas tunneme **2nd**arvu vaadates ära, kas ta on **paaris**arv või **paaritu** arv ?

näeme, et kõik järgukaalud on **paaris**arvud — ainus **paaritu** järgukaal on **1** :

. . . . 4096 2048 1024 512 256 128 64 32 16 8 4 2 1

. . . . arvu *murdosa* teisendusmeetod erineb oluliselt *täisarvu* teisendusest.
Kuna me murdarvudega järgnevas ei tegele, siis me arvu murdosa teisendusmeetodit ei vaatle.

☺☻ väikeste **2ndarvude kiirkoostamine 1**de **"sobitamise" teel õigetesse järkudesse**:

Vajaliku arvu kahendkuju saab koostada ka järguväärtuste **1** paigutamise teel vajalikesse 2ndjärkudesse. Selleks tuleb esmalt kirjutada välja 2ndsüsteemi järgukaalud piisava suuruseni:

. . . . . . **64 32 16 8 4 2 1**

37$_{10}$ **=**

.... ja peast arvutades täidame (**kõrgematest järkudest alates!**) vajalikud järgud "*ühtedega*" nii, et **1**-ga täidetud järkude **kaalude summa** võrduks soovitud 10ndarvuga.

37$_{10}$ **=**

. . . . . .	64	32	16	8	4	2	1	
	a_6	a_5	a_4	a_3	a_2	a_1	a_0	
	0	**1**						
	0	**1**	**0**					
	0	**1**	**0**	**0**				
	0	**1**	**0**	**0**	**1**			
	0	**1**	**0**	**0**	**1**	**0**		
	0	**1**	**0**	**0**	**1**	**0**	**1**	2
. . . . . .	64	32	16	8	4	2	1	

Eelpool toodud seitsme 2ndjärgu abil õnnestub esitada arve kuni väärtuseni 127:

$64 + 32 + 16 + 8 + 4 + 2 + 1 = \mathbf{127}_{10} = \mathbf{1\,1\,1\,1\,1\,1\,1}_2$

ülesanne:

Teisendada järgnevad 10ndarvud **2ndkujule** :

2 0 $_{10}$ = ... $_2$

8 7 $_{10}$ = ... $_2$

1 3 1 $_{10}$ = ... $_2$

210_{10} = ... $_2$

32_{10} = ... $_2$

$$20_{10} = \mathbf{10100}_2$$
$$87_{10} = \mathbf{1010111}_2$$
$$131_{10} = \mathbf{10000011}_2$$
$$210_{10} = \mathbf{11010010}_2$$
$$32_{10} = \mathbf{100000}_2$$

Lisaks alustele **p = 10** ja **p = 2** on olulisemateks arvusüsteemide alusteks veel **8** ja **16**, kuna nad on mõlemad arvu 2 astmed:

$$\mathbf{p} = \mathbf{8} = 2^3$$
$$\mathbf{p} = \mathbf{16} = 2^4$$

"*ühendsüsteem*" (arvusüsteem alusega **p** = **1**) ei ole võimalik. (*miks*?)

KAHEKSANDSÜSTEEM

8ndsüsteemi alus on 8 ja seega peab seal olema 8 võimalikku järguväärtust, milleks on kaheksa esimest araabia numbrit **0** . . . **7** :

$\mathbf{p = 8}$ $\qquad a_i \in \{ 0 \; 1 \; 2 \; 3 \; 4 \; 5 \; 6 \; 7 \}$

8ndsüsteemi järgukaalud: . . . 8^4 8^3 8^2 8^1 8^0 8^{-1} 8^{-2} 8^{-3} . . .

	8^4	8^3	8^2	8^1	8^0	8^{-1}	8^{-2}	8^{-3}
	4096	512	64	8	1	0.125		

8ndarve nimetatakse ka *oktaalarvudeks*.

ülesanne:

Leia nende kaheksandarvude **väärtus** :

25_8 = . . . $_{10}$

74_8 = . . . $_{10}$

123_8 = . . . $_{10}$

$$\mathbf{25}_8 = 2\times8^1 + 5\times8^0 = 16 + 5 = \mathbf{21}_{10}$$
$$\mathbf{74}_8 = 7\times8^1 + 4\times8^0 = 56 + 4 = \mathbf{60}_{10}$$
$$\mathbf{123}_8 = 1\times8^2 + 2\times8^1 + 3\times8^0 = \mathbf{83}_{10}$$

Teisendus 10ndsüsteemist 8ndsüsteemi

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

ülesanne:

Teisenda 10ndtäisarvud $\mathbf{83}_{10}$ $\mathbf{236}_{10}$ $\mathbf{109}_{10}$ 8ndkujule:

: 8		
236	**4**	↑ *madalaim järk*
29	**5**	
3	**3**	*kõrgeim järk*
0		

: 8	
109	**5**
13	**5**
1	**1**
0	

$$\mathbf{236}_{10} = \overset{64}{\mathbf{3}}\;\overset{8}{\mathbf{5}}\;\overset{1}{\mathbf{4}}\,{}_8 \qquad \mathbf{109}_{10} = \overset{64}{\mathbf{1}}\;\overset{8}{\mathbf{5}}\;\overset{1}{\mathbf{5}}\,{}_8$$

← *järgukaalud*

(kontrollimisvõimalus)

KUUETEISTKÜMNENDSÜSTEEM

hexadecimal (*hex*)

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

probleem : araabia numbreid (**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 $\quad a_i \in \{\, 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ A\ B\ C\ D\ E\ F \,\}$

16ndnumbrid **A . . . F** omavad väärtusi:

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

16ndsüsteemi järgukaalud:	. . .	16^4	16^3	16^2	16^1	16^0	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

$\ldots\ p^5\ \ p^4\ \ p^3\ \ p^2\ \ p^1\ \ p^0\ \ p^{-1}\ \ p^{-2}\ \ p^{-3}\ \ p^{-4}\ \ldots\ p^i\ \ldots$

$p^0 = 1$ suvalise **aluse p** korral

ülesanne:

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

$1\,2_{16} = \ldots_{10}$ $\qquad$ $F\,F_{16} = \ldots_{10}$

$4\,D_{16} = \ldots_{10}$ $\qquad$ $1\,0\,0_{16} = \ldots_{10}$

$A\,6_{16} = \ldots_{10}$ $\qquad$ $1\,C\,D_{16} = \ldots_{10}$

meenutuseks **16nd**süsteemi **järgukaalud**:	. . .	65536	4096	256	16	1
		16^4	16^3	16^2	16^1	16^0

$1\,2_{16} = 1\times16^1 + 2\times16^0 = \mathbf{18}_{10}$

$4\,D_{16} = 4\times16^1 + 13\times16^0 = \mathbf{77}_{10}$

$A\,6_{16} = 10\times16^1 + 6\times16^0 = \mathbf{166}_{10}$

$F\,F_{16} = \mathbf{255}_{10}$

$1\,0\,0_{16} = \mathbf{256}_{10}$

$1\,C\,D_{16} = \mathbf{461}_{10}$

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 **16nd**süsteemi toimub **16**-ga jagamise teel, kusjuures igal jagamissammul saadakse jäägina arvu järgmine 16ndnumber **0 . . . . F**.

ülesanne:

Teisendada järgnevad 10ndarvud **16ndkujule** :

$77_{10} = \ldots_{16}$

$32_{10} = \ldots_{16}$

$256_{10} = \ldots_{16}$

$77_{10} = \mathbf{4D}_{16}$

oli: $4D_{16} = 4 \times 16^1 + 13 \times 16^0 = \mathbf{77}_{10}$

$32_{10} = \mathbf{20}_{16}$

$256_{10} = \mathbf{100}_{16}$

16ndsüsteem on vajalik meie kodutöö alguses **!** (novembris - detsembris)

pane tähele:

16ndarve esitatakse tekstides / ASCII-tekstifailides / programmeerimiskeeltes eesliitega **0x**...... või tagaliitega ......**h** ......**H**

näide ($7E_{16}$ *kohta*) : **0x7E** **7Eh** **7EH**

Edaspidi vajame nendest arvusüsteemidest kõige rohkem **kahendsüsteemi**

ARVUSÜSTEEMID

kokkuvõttev loetelu

kuni arvutite ilmumiseni polnud vaja muid arvusüsteeme peale **10nd**süsteemi

$$2 \le p \le 16$$

2ndsüsteem: **p = 2** $\mathbf{a}_i \in \{0\ 1\}$

3ndsüsteem: p = 3 $\mathbf{a}_i \in \{0\ 1\ 2\}$

4ndsüsteem: p = 4 $\mathbf{a}_i \in \{0\ 1\ 2\ 3\}$

5ndsüsteem: p = 5 $\mathbf{a}_i \in \{0\ 1\ 2\ 3\ 4\}$

6ndsüsteem: p = 6 $\mathbf{a}_i \in \{0\ 1\ 2\ 3\ 4\ 5\}$

7ndsüsteem: p = 7 $\mathbf{a}_i \in \{0\ 1\ 2\ 3\ 4\ 5\ 6\}$

8ndsüsteem: **p = 8** $\mathbf{a}_i \in \{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\}$

9ndsüsteem: p = 9 $\mathbf{a}_i \in \{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\}$

10ndsüsteem: **p = 10** $\mathbf{a}_i \in \{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\}$

11ndsüsteem: p = 11 $\mathbf{a}_i \in \{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ A\}$

12ndsüsteem: p = 12 $\mathbf{a}_i \in \{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ A\ B\}$

13ndsüsteem: p = 13 $\mathbf{a}_i \in \{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ A\ B\ C\}$

14ndsüsteem: p = 14 $\mathbf{a}_i \in \{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ A\ B\ C\ D\}$

15ndsüsteem: p = 15 $\mathbf{a}_i \in \{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ A\ B\ C\ D\ E\}$

16ndsüsteem: **p = 16** $\mathbf{a}_i \in \{\mathbf{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ A\ B\ C\ D\ E\ F}\}$

olulised arvusüsteemid: $\mathbf{p} = \mathbf{2} = 2^1$ (kus $\mathbf{p} = 2^n$)

$\mathbf{p} = \mathbf{8} = 2^3$

$\mathbf{p} = \mathbf{10}$

$\mathbf{p} = \mathbf{16} = 2^4$

. . . 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.

iseseisev vabatahtlik kodune lahendamine:

rohelise õpiku **lk. 85** ülesanded :

- Teisendada 10ndarvud $\mathbf{75}_{10}$ $\mathbf{182}_{10}$ $\mathbf{273}_{10}$ $\mathbf{456}_{10}$ **2nd**kujule

- Teisendada 10ndarvud 95_{10} 182_{10} 273_{10} 456_{10} **8nd**kujule
- Teisendada 10ndarvud 79_{10} 182_{10} 273_{10} 456_{10} **16nd**kujule

(jagamisel sobib kasutada kalkulaatorit)

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, **8nd**süsteemi ja **16nd**süsteemi 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 **2nd**arvu viia **8nd**kujule ?

võimalik oleks teisendada **2nd** → **10nd** → **8nd** kuid see oleks asjatu töö

2ndarvu on võimalik "otse" teisendada (ümber kirjutada) tema **8nd**kujule, asendades (alates **2nd**arvu **madalamatest** järkudest)

iga tema järkude<u>kolmiku</u> **000 .... 111** vastava **8nd**numbriga **0 .... 7** nagu näitas eelnev 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	

ülesanne:

Viia 2ndarv: **1 0 1 1 0 1 0 1 0 0 1 1** $_2$ **8nd**kujule

viimine / ümberkirjutamine **8nd**kujule :

parim teisendusviis: Grupeerime **2nd**arvu järgud <u>3-järgulistesse</u> gruppidesse alates <u>madalamatest</u> järkudest :

1 0 1 1 0 1 0 1 0 0 1 1 1 $_2$

. . . . . 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 **1 0 1 1 0 1 0 1 0 0 1 1 1** $_2$ = **1 3 2 4 7** $_8$

! *viga:*

alates **kõrgematest** järkudest ei tohi grupeerida :

101| 101| 010 | 011 | 1

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

... kuna **täis**arvu <u>lõppu</u> ei tohi lisada **0-**lle — see "rikuks" arvu väärtuse !

ülesanne:

Viia sama 2ndarv: **1 0 1 1 0 1 0 1 0 0 1 1 1** $_2$ **16nd**kujule

Selleks grupeerime **2nd**arvu järgud **4 j ä r g u** kaupa alates madalamatest järkudest ja
asendame iga 2ndjärkude neliku temaga väärtuselt võrdse **16nd**numbriga **0 .... F** :
juhindudes vastavustabelist **2nd**süsteemi ja **16nd**süsteemi 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	

$1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1_2$

1	6	A	7
0001	0110	1010	0111

seega $1011010100111_2 = 16A7_{16}$

ülesanne:

Kontrollida eelnevalt leitud arvude võrdsust, leides eelmise näite **2nd**arvu, **8nd**arvu ja **16nd**arvu **väärtused** :

$$1011010100111_2 = \ ...\ _{10}$$
$$13247_8 = \ ...\ _{10}$$
$$16A7_{16} = \ ...\ _{10}$$

arvestades järgukaalusid nendes arvusüsteemides :

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

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

$13247_8 = 1\times8^4 + 3\times8^3 + 2\times8^2 + 4\times8^1 + 7\times8^0 = \ ....\ _{10}$

$13247_8 = 1\times8^4 + 3\times8^3 + 2\times8^2 + 4\times8^1 + 7\times8^0 = 5799_{10}$

$16A7_{16} = 1\times16^3 + 6\times16^2 + 10\times16^1 + 7\times16^0 = \ ....\ _{10}$

$16A7_{16} = 1\times16^3 + 6\times16^2 + 10\times16^1 + 7\times16^0 = 5799_{10}$

kontrollitud: kõik need 3 (erinevate arvusüsteemide) arvu on **võrdsed** :

$1011010100111_2 = 13247_8 = 16A7_{16} = 5799_{10}$

iseseisev vabatahtlik kodune lahendamine:

- kontrolli kalkulaatoriga, kas täna leitud **8nd**arv ja **16nd**arv omavad väärtust 5799 **?**

$13247_8 = 1\times8^4 + 3\times8^3 + 2\times8^2 + 4\times8^1 + 7\times8^0 = 5799_{10}$?

$16A7_{16} = 1\times16^3 + 6\times16^2 + 10\times16^1 + 7\times16^0 = 5799_{10}$?

sellise asendamisega saab teisendada ka "vastupidises suunas" :

numbrite **vastupidise** asendamisega kahendjärkude kolmikuteks või nelikuteks saab arvu **8nd**kujult või **16nd**kujult kergesti üle minna tema **2nd**kujule: **8nd** ——→ **2nd** **16nd** ——→ **2nd**

?. . . aga **4nd**süsteem ? see on ju kah alusega $p = 2^n = 2^2 = 4$?

4ndsüsteem (ebaoluline)

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

4ndsüsteemi ja **2nd**süsteemi siduv vastavustabel oleks:

2nd	4nd
00_2	0_4
01_2	1_4
10_2	2_4
11_2	3_4

$0\,1\,0\,0\,1\,0\,1\,1_2$

seega **4nd**süsteemi

ümberkirjutamiseks

vaja grupeerida :

0 1 | 0 0 | 1 0 | 1 1

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

?. . . aga kui grupeerida 2ndarvu järgud **ühe**kaupa ?

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

. . . siis saame mõttetu "teisenduse" 2ndsüsteemist . . . **2nd**süsteemi **!**

Programmeerimiskeeled võimaldavad arvude ("*arvkonstantide*") esitamist

10ndsüsteemis

2ndsüsteemis

8ndsüsteemis

16ndsüsteemis

. . . . . kuid mitte 4ndsüsteemis.

?. . . kas on võimalik **8nd**arv ümberkirjutada **16nd**kujule ka "**otse**" ehk ilma **2nd**süsteemi "läbimata" **?**

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

13247_8 ←——→ $16A7_{16}$

OTSE ei saa

saab nii : **8nd** ←——→ **2nd** ←——→ **16nd**

16ndsüsteemi tähtsus

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

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

00000000_2

00000001_2

00000010_2

00000011_2

.

.

01111001_2

01111010_2

01111011_2

01111100_2

.

.

11111100_2

11111101_2

11111110_2

11111111_2

16ndsüsteem võimaldab esitada / näidata baitide sisu (ja üldse igasuguseid kahendkoode) palju kompaktsemalt võrreldes nende "vahetu" esitamisega kahendkujul.

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

00000000_2

00000001_2

00000010_2

00000011_2

.

.

01111001_2

01111010_2

01111011_2

.

.

11111101_2

11111110_2

11111111_2

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

$00000000_2 = 00_{16}$

$00000001_2 = 01_{16}$

$00000010_2 = 02_{16}$

$00000011_2 = 03_{16}$

.

.

$01111001_2 =$? ? $_{16}$

$01111010_2 =$? ? $_{16}$

$01111011_2 = 7B_{16}$

.

$01111111_2 =$? ? $_{16}$

.

$10001001_2 = 89_{16}$

$10001010_2 = 8A_{16}$

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

$11111110_2 = FE_{16}$

$11111111_2 = FF_{16}$

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

$10110111_2 = B7_{16}$ $\quad$ $00000101_2 = 05_{16}$

$11100100_2 = E4_{16}$ $\quad$ $01101010_2 = 6A_{16}$

$11111110_2 = FE_{16}$ $\quad$ $11111111_2 = FF_{16}$

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

16ndsüsteemi kasutatakse **2nd**arvude kompaktsemaks esitamiseks

(**16nd**süsteem on vajalik ka **kodutöös**)

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

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

$1\,0\,1\,0\,1\,1_2 = \,. . ._8$

$1\,0\,1\,|\,0\,1\,1_2 = 5\,3_8$

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 **k a h e n d v e k t o r i t** :

00101101 $\quad$ **010** $\quad$ **11011** $\quad$ **10** $\quad$ **1** $\quad$ **000101**

Kahendvektoril pole seost füüsikast tuntud vektori mõistega.
Erinevalt *kahendarvust* ei tohi kahend**vektoris** ära jätta algusnulle:

$0\,0\,0\,1\,0\,1 \neq 1\,0\,1$

Kahendvektori järkudel pole *järgukaalu*. Tema sarnasuse tõttu 2ndarvudega osutub mõnes rakenduses siiski kasulikuks ja vajalikuks vaadelda teda