

# Skaičiaus formatai. Sveikieji skaičiai (1)

Dažniausiai programavime naudojami skaičiai – sveikieji ir realieji. Sveikieji skaičiai dažniausiai saugomi *short*, *int* arba *long* duomenų tipuose (žinant, kad skaičius niekada nebus neigiamas ir norint praplėst reikšmių diapazoną, galima naudoti *unsigned*

```
short fantastiskasShortas;
```

```
int klasiskasIntegeris;
```

```
long ispudingasLongas;
```

```
uint bezenklisPasakiskasIntas;
```

Tipas	Baitų kiekis	Galimos dešimtainės reikšmės
Short	2	Nuo -32768 iki 32767
Int	4	Nuo -2147483648 iki 2147483647
Long	8	Nuo -9223372036854775808 iki 9223372036854775807

*Pastaba: baitų kiekis gali skirtis nuo programavimo kalbos ir esamos architektūros.*

# Skaičiaus formatai. Sveikieji skaičiai (2)

Kaip saugoti neigiamus skaičius?

Skaičiaus be ženklo ir skaičiaus su ženklu formatais. Du skirtingi skaičiai (neigiamas ir teigiamas) gali atrodyti identiškai (pasivertus juos į bitus), tačiau **nuo to, kaip interpretuosime, priklausो, koks tai skaičius.**

$1001\ 1100_2 = 156_{10}$  (jei tai skaičius be ženklo)

$1001\ 1100_2 = -100_{10}$  (jei tai skaičius su ženklu)

# Skaičiaus formatai. Realieji skaičiai

- Kaip saugoti realiuosius skaičius?

Mums patogiu trupmeninę dalį atskirti užrašius kablelį.

Užrašę 13,37, mes suprantam, kad tai skaičius  $13\frac{37}{100}$ .

Kompiuteriui „užrašyti“ kablelį nėra taip paprasta (įsivaizduojam skaičiaus bitinę išraišką). Ką daryt?



Išsaugoti realiesiems skaičiams buvo sugalvoti *floating point* (*slankaus kablelio*) formatai. Mūsų architektūroje, 8087 koprocesorius palaiko IEEE-754 standartą.

```
float karaliskasFloatas;
```

```
double patsGeriausiasDablas;
```

# Floating point formatai (1)

*Single-precision floating-point format / Trumpas realus formatas (4 baitai)*



*Double-precision floating-point format / Ilgas realus formatas (8 baitai)*



*Vidinis realus formatas (10 baitų)*



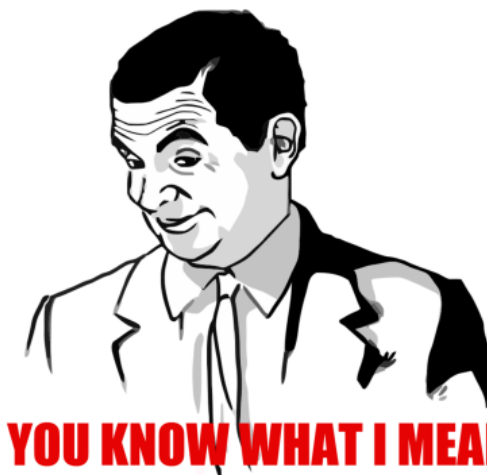
\* i bitas parodo, koks skaičius eina prieš mantisę

Kaip atsiminti eiliškumą pagal dydį:  
Iš pradžių būna trumpas, po to ilgas,  
o po to jau ir vidinis...

	Ženklo bitas	Charakteristika	i bitas	Mantisė
Trumpas realus	1 bitas	8b (eilė + 127 <sub>10</sub> )	NĖRA	23 bitai
Ilgas realus		11b (eilė + 1023 <sub>10</sub> )		52 bitai
Vidinis realus		15b (eilė + 16383 <sub>10</sub> )	1 bitas	63 bitai

Normalizuota forma:  $(-1)^S * 2^{\text{eilė}} * 1, \text{mantisė (i bitas = 1)}$

Nenormalizuota forma:  $(-1)^S * 2^{\text{eilė}} * i, \text{mantisė}$



# Floating point formatai (2)



- S (Sign) bitas – parodo, koks yra skaičiaus ženklas.  
Skaičius neigiamas  $\rightarrow$  Sign TRUE  $\rightarrow$  S=1  
Skaičius teigiamas  $\rightarrow$  Sign FALSE  $\rightarrow$  S=0
- Charakteristikos skaičiavimas:
  - Kablelį turime „nustumti“ taip, kad jis būtų po vyriausio vieneto
  - Eilė: per kiek pozicijų „stumsime“ kablelį. Jei stumsime į kairę pusę – eilė teigiama, jeigu į dešinę pusę – eilė neigiama.

---

*Pvz.: Verčiam 75,45 į floating point formatą. Pradedam dirbti su sveikąją dalimi: pasivertę 75 į dvejetainį pavidalą, gaunam 1001001, trupmeninė\_dalis. Pastūmus kablelį: 1,001001trupmeninė\_dalis. Stūmėm į kairę per 6 pozicijas  $\rightarrow$  eilė = +6.*

# Floating point formatai (3)



- Nebūtina atsiminti 127 (7Fh), 1023 (3FFh) ir 16383 (3FFFh). Užtenka atsiminti, kiek bitų užima charakteristika (8, 11 arba 15) ir surašyt į vyriausią poziciją nulį, o visas kitas – vienetus.

*Pvz.: Verčiame skaičių į ilgą realų formatą. Vadinasi, charakteristiką užims 11 bitų. Reiškia, charakteristika, dar nepridėjus eilės, bus tokia: 011 1111 1111*

- Nebūtina atsiminti, kiek bitų užima mantisė. Užtenka žinoti, kiek bitų išvis užima formatas ir kiek bitų užima ženkle, charakteristikos ir i bitas (jei jis yra). Visi likę bitai bus skirti mantisei.  
*Pvz.: verčiam skaičių į vidinį realų formatą. Vadinasi, ženkle, charakteristikos ir i bitai užims 17 bitų (1+15+1). Reiškia, mantisei liks 63 bitai (80 – 17 = 63).*
- [Insert your\_trick here]

# Vertimas į floating point formatą (1)

Turime skaičių -155,48. Norime paversti jį į 4 baitų floating point formatą, o atsakymą užrašyti šešioliktainėje sistemoje.

Skaičius neigiamas → Sign bitas = 1.

Įsiminę sign bitą, toliau dirbame tik su teigiamu skaičiumi (155,48). Verčiant atskirai dirbame su sveikąja ir trupmenine dalimis.

## **Sveikoji dalis:**

- Pasiverčiame sveikąją dalį į dvejetainį pavidalą.  $155_{10} = 10011011_2$
- Po vyriausio vieneto yra 7 bitai → jau turime 7 mantisės bitus.
- Mantisė iš viso sudaryta iš 23 bitų. Reiškia, trūksta dar 16-kos ( $23-7=16$ ).

# Vertimas į floating point formatą (2)

## Trupmeninė dalis:

- Skaičiaus trupmeninę dalį (0,48) reikia versti į dvejetainę. Mum reikia versti tol, kol gausime 16 bitų (nes tik tiek trūksta iki pilnos mantisės).  
*Pastaba: iš kuo mažiau baitų sudarytas floating-point formatas, tuo mažesnis jo tikslumas.*
- Vertimo procesas:
  - Dauginam trupmeninę dalį iš 2
  - Pasiimam sveikąją dalį (ją įrašysim į mantisę)
  - Toliau dirbam **tik** su trupmenine dalim



# Vertimas į floating point formatą (3)

$0,48 * 2 = 0,96$	$0,68 * 2 = 1,36$	$0,88 * 2 = 1,76$	$0,08 * 2 = 0,16$
$0,96 * 2 = 1,92$	$0,36 * 2 = 0,72$	$0,76 * 2 = 1,52$	$0,16 * 2 = 0,32$
$0,92 * 2 = 1,84$	$0,72 * 2 = 1,44$	$0,52 * 2 = 1,04$	$0,32 * 2 = 0,64$
$0,84 * 2 = 1,68$	$0,44 * 2 = 0,88$	$0,04 * 2 = 0,08$	$0,64 * 2 = 1,28$

- Kitas būdas versti – greitesnis, bet pavojingesnis. Šiuo būdu trupmeninę dalį dauginam iš 16, pasiimam sveikąją dalį, ją paverčiam į dvejetainę formą ir turim reikiamus bitukus.

$0,48 * 16 = 7,68$	$0,68 * 16 = 10,88$	$0,88 * 16 = 14,08$	$0,08 * 16 = 1,28$
$7_{10} = 0111_2$	$10_{10} = 1010_2$	$14_{10} = 1110_2$	$1_{10} = 0001_2$

## Turime:

- S bitas = 1

- Eilė = +7, nes kablelį stumiam per 7 vietas į kairę ( $1001\ 1011,011... \rightarrow 1,0011011011...$ )

- Charakteristika =  $127 + 7 = 134 = 1000\ 0110$

- Mantisė sudaryta iš 23 bitų (sudaryta iš 7 bitų iš sveikosios dalies ir 16 bitų iš trupmeninės dalies)

1100	0011	0001	1011	0111	1010	1110	0001
C	3	1	B	7	A	E	1

Ats.: C31B7AE1h

# Pastebėjimai

- Verčiant trupmeninę dalį pastebėjus besikartojantį pattern'ą (ciklą), galima dalį dauginimo praleisti.

---

*Pvz.: Versdami 5,2 į dvejetainę formą, gaunam 101, (0011). Normalizuota forma: 1,01(0011). Reiškia, į mantisę įrašę 01, toliau rašysime 00110011... tol, kol neužpildysime visos likusios mantisės.*
- Verčiant skaičių, kurio modulis mažesnis už 1, kablelis stumsis į dešinę ir eilė bus neigiama.

---

*Pvz.: Verčiam -0,04 į 4 baitų floating point formatą.  $0,04_{10} = 0,0000011110..._2$ . Stumiam kablelį iki vyriausio vieneto ir gaunam normalizuota formą: 0,000001,11110. Pastūmėm per 6 vietas į dešinę → charakteristika =  $127 - 6 = 121$ .*
- Mūsų sprendžiamuose uždaviniuose laikysime, kad mantisė nukerpama ir jos neapvalinsime.
- Pasitikrinimui naudokite interneto IEEE-754 konverterius (pvz. [šita](#)).

# Vertimas iš floating point formato

Verčiant iš floating point formato į dešimtainę sistemą, reikia pritaikyti atvirkščią algoritmą prieš tai aprašytam.

- Šešioliktainiai skaičiai verčiam į dvejetainius skaičius
- Išsitraukti ženklą, charakteristikos, (i) ir mantisės reikšmes
- Pasirašyti 1, mantisė
- Charakteristika pasiverčiam į dešimtainį skaičių. Atėmus 7F/3FF/3FFF, gauname eilę.
- Atitinkamai patraukti kablelį į reikiamą pusę per eilės reikšmę
- Gautą skaičių versti į dešimtainę sistemą (*pastaba: jis nebūtinai gausis toks pats, koks buvo prieš vertimą*)
- Pasirašyti atitinkamą ženklą (nusprendus iš ženklo bito)
- Vuolia

# Uždaviniai

1. Užrašykite dešimtainį skaičių  $-33,33$  slankaus kablelio formatu 8 baituose šešioliktaine sistema.
2. Užrašykite dešimtainį skaičių  $0,00003$  slankaus kablelio formatu 4 baituose šešioliktaine sistema.
3. Užrašykite dešimtainį skaičių  $-13,2$  slankaus kablelio vidiniu realiu formatu.
4. Užrašykite dešimtainį skaičių  $67,67$  slankaus kablelio formatu 4 baituose šešioliktaine sistema.

# Atsakymai

1. C040 AA3D 70A3 D70A
2. 37FB A882
3. C002 D333 3333 3333 3333
4. 4287 570A