Številski sistemi
Number Systems

Decimalni sistem / The Decimal System

- Sistem, ki temelji na decimalnih števkah za prikaz števil
- System based on decimal digits to represent numbers

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9$$

- Številka 83 na primer pomeni osem desetic plus tri:
- For example the number 83 means eight tens plus three:

$$83 = \boxed{}$$

- Številka 4728 pomeni štiri tisočice, sedem stotic, dve desetici in osem:
- The number 4728 means four thousands, seven hundreds, two tens, plus eight:

$$4728 = \boxed{}$$

Osnova / Base

- Desetiški sistem ima **osnovo** 10. To pomeni, da se vsaka števka števila pomnoži z 10 na potenco, ki ustreza položaju te števke:

- The decimal system is said to have a ***base*, or *radix*,** of 10. This means that each digit in the number is multiplied by 10 raised to a power corresponding to that digit's position:

$$83 = \boxed{}$$

$$4728 = \boxed{}$$

Decimalni ulomki / Decimal Fractions

- Enako načelo velja za decimalne ulomke. A se pri njih se uporabljajo negativne potence 10. Decimalni ulomek 0,256 pomeni tako 2 desetini plus 5 stotin plus 6 tisočin:

- The same principle holds for decimal fractions, but negative powers of 10 are used. Thus, the decimal fraction 0.256 stands for 2 tenths plus 5 hundredths plus 6 thousandths:

$$0.256 = \boxed{\phantom{2\times10^{-1}+5\times10^{-2}+6\times10^{-3}}}$$

- Številka s celim in delnim delom ima števke pomnožene s pozitivnimi in negativnimi potencami števila 10:

- A number with both an integer and fractional part has digits raised to both positive and negative powers of 10:

$$442.256 = \boxed{\phantom{4\times10^{2}+4\times10^{1}+2\times10^{0}+2\times10^{-1}+5\times10^{-2}+6\times10^{-3}}}$$

Pomembnost števk / Digit significance

- Najpomembnejša števka
  - Najbolj leva števka (nosi največjo vrednost)
- Najmanj pomembna številka
  - Skrajna desna števka

- Most significant digit
  - The leftmost digit (carries the highest value)
- Least significant digit
  - The rightmost digit

- Pozicijska razlaga decimalnega števila
- Positional Interpretation of a Decimal Number

4	7	2	2	5	6
100s 10^2 position 2	10s 10^1 position 1	1s 10^0 position 0	tenths 10^{-1} position −1	hundredths 10^{-2} position −2	thousandths 10^{-3} position −3

509 ?

$$X = \sum_i (d_i \times 10^i)$$

Pozicijski številčni sistemi / Positional Number Systems

- Vsako število je predstavljeno z nizom števk, v katerem ima števka na položaju i utež r^i, kjer je r osnova številčnega sistema.
- Splošna oblika števila v takšnem sistemu z osnovo r je

- Each number is represented by a string of digits in which each digit position *i* has an associated weight r^i, where *r* is the *radix*, or *base,* of the number system.
- The general form of a number in such a system with radix *r* is

$$(\ldots a_3 a_2 a_1 a_0 . a_{-1} a_{-2} a_{-3} \ldots)_r$$

kjer je vrednost poljubne števke a_i celo število v območju $0 < a_i < r$. Pika med a_0 in a_{-1} se imenuje točka osnove.

where the value of any digit a_i *is an integer in the range* $0 \leq a_i < r$. The dot between a_0 *and* a_{-1} is called the ***radix point.***

- Pozicijska razlaga števila z osnovo 7

- Positional Interpretation of a Number in Base 7

Position	4	3	2	1	0	−1
Value in exponential form	7^4	7^3	7^2	7^1	7^0	7^{-1}
Decimal value	2401	343	49	7	1	1/7

Binarni sistem / The Binary system (2)

- Samo dve števki, 1 in 0
- Predstavljene v osnovi 2
- Števki 1 in 0 v binarnem zapisu imata enak pomen kot v decimalnih zapisih:

- Only two digits, 1 and 0
- Represented to the base 2
- The digits 1 and 0 in binary notation have the same meaning as in decimal notation:

$$0_2 = 0_{10}$$
$$1_2 = 1_{10}$$

Binarni sistem / The Binary system (2)

- Za predstavljanje večjih številk ima vsaka številka v binarnem številu vrednost, odvisno od svojega položaja:
- To represent larger numbers each digit in a binary number has a value depending on its position:

$$10_2 = \boxed{} = 2_{10}$$

$$11_2 = \boxed{} = 3_{10}$$

$$100_2 = \boxed{} = 4_{10}$$

in tako naprej. Ponovno so frakcijske vrednosti predstavljene z negativnimi potencami osnove:

and so on. Again, fractional values are represented with negative powers of the radix:

$$1001.101 = \boxed{} = 9.625_{10}$$

Pretvarjanje med binarnim in decimalnim številom / Converting Between Binary and Decimal

- Binarni zapis v decimalni zapis:
  - Vsako binarno števko pomnožite z ustrezno močjo 2 in prištevamo rezultate
- Decimalni zapis v binarni zapisu:
  - Celi del in delež se obdelujeta ločeno

- Binary notation to decimal notation:
  - Multiply each binary digit by the appropriate power of 2 and add the results
- Decimal notation to binary notation:
  - Integer and fractional parts are handled separately

Cela števila / Integers (1)

- V binarnem zapisu celo število predstavimo z zapisom:

- In binary notation, an integer is represented by

$$b_{m-1}b_{m-2}\ldots b_2b_1b_0 \qquad b_i = 0 \text{ or } 1$$

in ima v desetiškem vrednost

and has the decimal value of

$$(b_{m-1} * 2^{m-1}) + (b_{m-2} * 2^{m-2}) + \ldots + (b_1 * 2^1) + b_0$$

- Recimo, da je treba pretvoriti decimalno celo število N v binarno obliko. Če v decimalnem sistemu delimo N z 2, dobimo količnik N1 in preostanek R0, kar lahko zapišemo

- Suppose it is required to convert a decimal integer *N* into binary form. If we divide *N* by 2, in the decimal system, and obtain a quotient N_1 and a remainder R_0, we may write

$$N = 2 * N_1 + R_0 \qquad R_0 = 0 \text{ or } 1$$

Continued . . .

Cela števila / Integers (2)

- Nato delimo dobljeni količnik N_1 z 2. Predpostavimo, da je novi količnik N_2, nov preostanek R_1. Potem

- Next, we divide the quotient N_1 by *2*. Assume that the new quotient is N_2 and the new remainder R_1. Then

$$N_1 = 2 * N_2 + R_1 \qquad R_1 = 0 \text{ or } 1$$

torej

so that

$$N = 2(2N_2 + R_1) + R_0 = (N_2 * 2^2) + (R_1 * 2^1) + R_0$$

- Če delimo naprej

- If next

$$N_2 = 2N_3 + R_2$$

dobimo

we have

$$N = (N_3 * 2^3) + (R_2 * 2^2) + (R_1 * 2^1) + R_0$$

Continued . . .

Cela števila / Integers (3)

- Ker je $N > N_1 > N_2 \dots$, bo nadaljevanje tega zaporedja sčasoma ustvarilo količnik $N_{m-1} = 1$ (razen decimalnih celih števil 0 in 1, katerih binarni ekvivalenti sta 0 in 1) in preostanek R_{m-2}, ki je 0 ali 1. Potem je

- Because $N > N_1 > N_2 \dots$, continuing this sequence will eventually produce a quotient $N_{m-1} = 1$ (except for the decimal integers 0 and 1, whose binary equivalents are 0 and 1, respectively) and a remainder R_{m-2}, which is 0 or 1. Then

$$N = (1 * 2^{m-1}) + (R_{m-2} * 2^{m-2}) + \dots + (R_2 * 2^2) + (R_1 * 2^1) + R_0$$

binarna oblika N. Zato pretvorimo iz osnove 10 v osnovo 2 s ponavljanjem deljenja z 2. Ostanek in končni količnik 1 nam dajo (po naraščajočem pomenu) binarne števke N.

which is the binary form of *N*. Hence, we convert from base 10 to base 2 by repeated divisions by 2. The remainders and the final quotient, 1, give us, in order of increasing significance, the binary digits of *N*.

Primer / Example

$$12_{10} = x_n \cdot 2^n + \ldots + x_1 \cdot 2^1 + 0$$

$$12_{10} = 2\underbrace{(x_n \cdot 2^{n-1} + \ldots + x_1 \cdot 2^0)}_{6} + 0$$

$$12 = 2 \cdot 6 + \boxed{0}$$

$$6_{10} = x_n \cdot 2^{n-1} + \ldots + x_2 \cdot 2^1 + x_1 \cdot 2^0$$

$$6_{10} = 2 \cdot \underbrace{(x_n \cdot 2^{n-2} + \ldots + x_2 \cdot 2^0)}_{3} + 0$$

$$6 = 2 \cdot 3 + \boxed{0}$$

$$3_{10} = 2 \cdot \underbrace{(x_n \cdot 2^{n-3} + \ldots + x_3 \cdot 2^0)}_{1} + 1$$

$$3_{10} = 2 \cdot \underbrace{(x_3 \cdot 2^0)}_{1} + 1$$

$$3 = 2 \cdot 1 + \boxed{1}$$

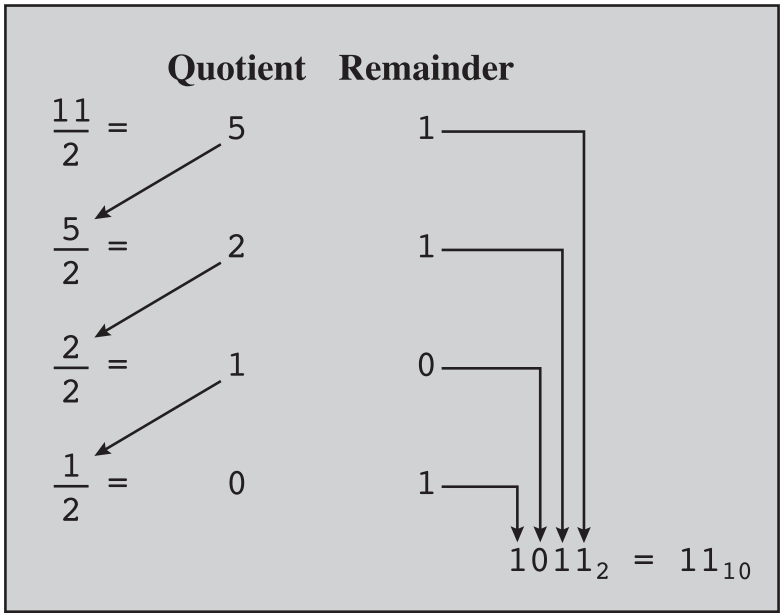
$$1_{10} = x_3 \cdot 2^0$$

$$1_{10} = 2 \cdot 0 + 1$$

$$1 = 2 \cdot 0 + \boxed{1}$$

2 1

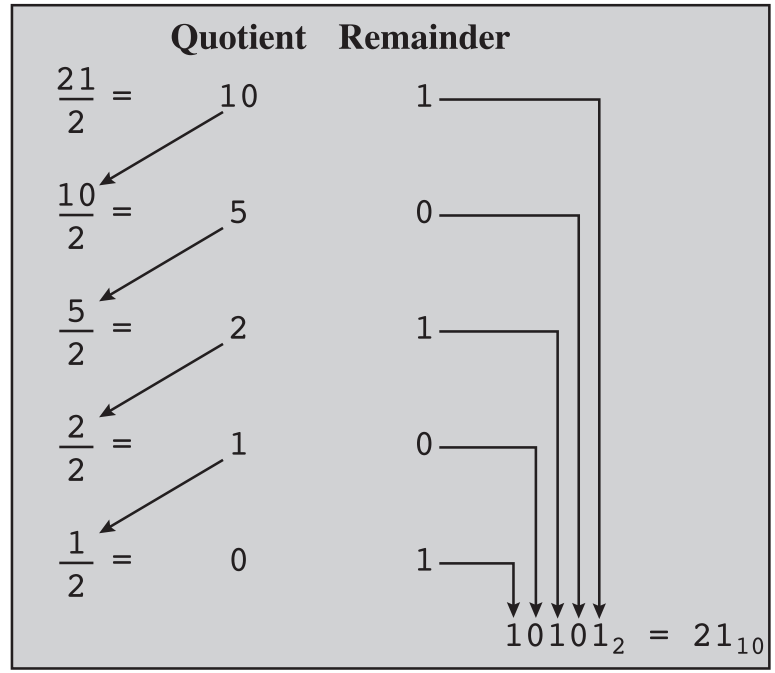
Primeri pretvorbe iz decimalnih zapisov v dvojiški zapis za celi števili:



(a) 11_{10}

Naredite: 2048, 36782

Examples of Converting from Decimal Notation to Binary Notation for Integers:



(b) 21_{10}

You do: 2048, 36782

Ulomki / Fractions (1)

- Za delež se spomnite, da je v binarnem zapisu število z vrednostjo med 0 in 1 predstavljeno s
- For the fractional part, recall that in binary notation, a number with a value between 0 and 1 is represented by

$$0.b_{-1}b_{-2}b_{-3}\ldots \qquad b_i = 0 \text{ or } 1$$

in ima vrednost

and has the value

$$(b_{-1} * 2^{-1}) + (b_{-2} * 2^{-2}) + (b_{-3} * 2^{-3})\ldots$$

To lahko zapišemo kot

This can be rewritten as

$$2^{-1} * (b_{-1} + 2^{-1} * (b_{-2} + 2^{-1} * (b_{-3} + \ldots)\ldots))$$

Continued . . .

Ulomki / Fractions (2)

- Recimo, da želimo število F (0 < F < 1) pretvoriti iz decimalnega v binarni zapis. Vemo, da se lahko F izrazi v obliki

- Suppose we want to convert the number *F (0 < F < 1)* from decimal to binary notation. We know that F can be expressed in the form

$$F = 2^{-1} * (b_{-1} + 2^{-1} * (b_{-2} + 2^{-1} * (b_{-3} + . . .) . . .))$$

- Če F pomnožimo z 2, dobimo

- If we multiply *F* by 2, we obtain

$$2 * F = b_{-1} + 2^{-1} * (b_{-2} + 2^{-1} * (b_{-3} + . . .) . . .)$$

Continued . . .

Ulomki / Fractions (3)

$$2 * F = b_{-1} + 2^{-1} * (b_{-2} + 2^{-1} * (b_{-3} + . . .) . . .)$$

- Iz te enačbe vidimo, da je celi del (2 * F), ki mora biti bodisi 0 ali 1, ker je 0 < F <1, preprosto b_{-1}. Torej lahko rečemo (2 * F) = $b_{-1} + F_1$, kjer je 0 < F_1 <1 in kjer

- From this equation, we see that the integer part of (2 * *F),* *which must be* either 0 or 1 because 0 < *F < 1, is simply* b_{-1}. *So we can say (2 * F) =* $b_{-1} + F_1$, where 0 < F_1 < *1 and where*

$$F_1 = 2^{-1} * (b_{-2} + 2^{-1} * (b_{-3} + 2^{-1} * (b_{-4} + . . .) . . .))$$

Continued . . .

Ulomki / Fractions (4)

$$F_1 = 2^{-1} * (b_{-2} + 2^{-1} * (b_{-3} + 2^{-1} * (b_{-4} + \ldots) \ldots))$$

- Da bi našli b_{-2}, ponovimo postopek. Na vsakem koraku se delež števila iz prejšnjega koraka pomnoži z 2. Števka na levi strani decimalne pike v rezultatu bo 0 ali 1 in prispeva k binarni predstavitvi, začenši z najpomembnejšo številko. Delež rezultata se v naslednjem koraku uporabi za množenje z 2.

- To find b_{-2}, we repeat the process. At each step, the fractional part of the number from the previous step is multiplied by 2. The digit to the left of the decimal point in the product will be 0 or 1 and contributes to the binary representation, starting with the most significant digit. The fractional part of the product is used as the multiplicand in the next step.

Continued . . .

Examples of Converting from Decimal Notation To Binary Notation For Fractions

Product	Integer Part	0.110011_2
$0.81 \times 2 = 1.62$	1	
$0.62 \times 2 = 1.24$	1	
$0.24 \times 2 = 0.48$	0	
$0.48 \times 2 = 0.96$	0	
$0.96 \times 2 = 1.92$	1	
$0.92 \times 2 = 1.84$	1	

(a) $0.81_{10} = 0.110011_2$ (approximately)

Product	Integer Part	0.01_2
$0.25 \times 2 = 0.5$	0	
$0.5 \times 2 = 1.0$	1	

(b) $0.25_{10} = 0.01_2$ (exactly)

Vaja / Practice

V dvojiško

- 0,98021
- 0,1123
- 234,234
- 99,101

V desetiško

- 1001
- 11011,101
- 1111,0001
- 0,110011

Into the binary

- 0.98021
- 0.1123
- 234.234
- 99.101

Into the decimal

- 1001
- 11011.101
- 1111.0001
- 0.110011

Šestnajstiška predstavitev / Hexadecimal Notation

- Binarne števke so združene v sklope štirih bitov, ki se imenujejo polbajti ali polzlogi
- Vsaka možna kombinacija štirih binarnih števk dobi simbol:

- Binary digits are grouped into sets of four bits, called a *nibble*
- Each possible combination of four binary digits is given a symbol, as follows:

0000 = 0	0100 = 4	1000 = 8	1100 = C
0001 = 1	0101 = 5	1001 = 9	1101 = D
0010 = 2	0110 = 6	1010 = A	1110 = E
0011 = 3	0111 = 7	1011 = B	1111 = F

- Ker je uporabljenih 16 simbolov, se notacija imenuje šestnajstična, 16 simbolov pa šestnajstična
- Tako

- Because 16 symbols are used, the notation is called *hexadecimal* and the 16 symbols are the *hexadecimal digits*
- Thus

$$2C_{16} = (2_{16} * 16^1) + (C_{16} * 16^0) = (2_{10} * 16^1) + (12_{10} * 16^0) = 44$$

Decimal (base 10)	Binary (base 2)	Hexadecimal (base 16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	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
16	0001 0000	10
17	0001 0001	11
18	0001 0010	12
31	0001 1111	1F
100	0110 0100	64
255	1111 1111	FF
256	0001 0000 0000	100

Prednosti / Advantages

- Razlogi za uporabo šestnajstiškega zapisa so naslednji:
  - Je bolj kompakten kot binarni zapis.
  - Na večini računalnikov so binarni podatki predstavljeni z večkratnikom 4 bitov in s tem večkratnikom šestnajstiških števk.
  - Izjemno enostavno je pretvoriti med binarnim in šestnajstnim zapisom.

- The reasons for using hexadecimal notation are as follows:
  - It is more compact than binary notation.
  - In most computers, binary data occupy some multiple of 4 bits, and hence some multiple of a single hexadecimal digit.
  - It is extremely easy to convert between binary and hexadecimal notation.

Primer / Example

$$\begin{array}{ccc} 1101 & 1110 & 0001 = \mathbf{DE1}_{16} \\ \mathbf{D} & \mathbf{E} & \mathbf{1} \end{array}$$

Poskusite

- 1001 1011 1110 0110
- 0001 0100 0010 1000
- 0111 1110 0101 1010
- 1101 1111 0000 1110

Try out

- 1001 1011 1110 0110
- 0001 0100 0010 1000
- 0111 1110 0101 1010
- 1101 1111 0000 1110

Vaje / Exercises (1)

- Naslednja binarna števila pretvorite v decimalna števila:

  a. 001100
  b. 000011
  c. 011100
  d. 111100
  e. 101010
- Naslednja binarna števila pretvorite v njihove decimalna števila:
  a. 11100.011
  b. 110011.10011
  c. 1010101010.1

- Convert the following binary numbers to their decimal equivalents:
  a. 001100
  b. 000011
  c. 011100
  d. 111100
  e. 101010
- Convert the following binary numbers to their decimal equivalents:
  a. 11100.011
  b. 110011.10011
  c. 1010101010.1

Vaje / Exercises (2)

- Naslednje decimalne številke pretvorite v binarna števila:
  a. 64
  b. 100
  c. 111
  d. 145
  e. 255
- Naslednje decimalne številke pretvorite v binarna števila :
  a. 34.75
  b. 25.25
  c. 27.1875

- Convert the following decimal numbers to their binary equivalents:
  a. 64
  b. 100
  c. 111
  d. 145
  e. 255
- Convert the following decimal numbers to their binary equivalents:
  a. 34.75
  b. 25.25
  c. 27.1875

Vaje / Exercises (2)

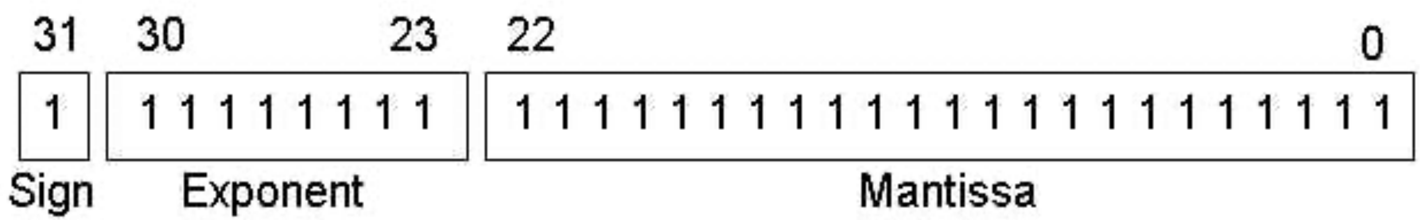
- Naslednja šestnajstiška števila pretvorite v decimalna števila:
  a. F,4
  b. D3,E
  c. 1111,1
  d. 888,8
  e. EBA,C
- Naslednja decimalna števila pretvorite v šestnajstiška števila:
  a. 16
  b. 80
  c. 2560
  d. 3000
  e. 62,52

- Convert the following hexadecimal numbers to their decimal equivalents:
  a. F.4
  b. D3.E
  c. 1111.1
  d. 888.8
  e. EBA.C
- Convert the following decimal numbers to their hexadecimal equivalents:
  a. 16
  b. 80
  c. 2560
  d. 3000
  e. 62.52

Plavajoča vejica / Floating point notation (1)

- IEEE kratko realno: 32 bitov za znak, 8 bitov za eksponent in 23 bitov za mantiso. Imenuje se tudi enojna natančnost.

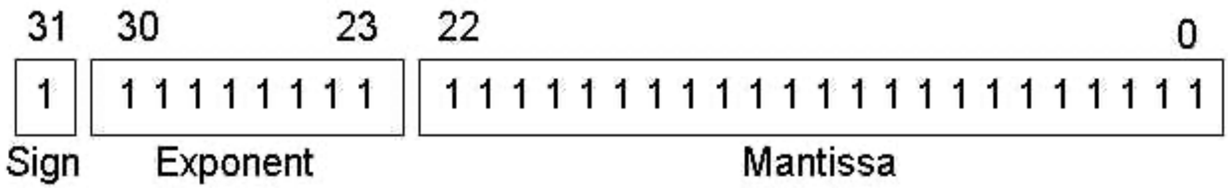
- IEEE Short Real: 32 bits1 bit for the sign, 8 bits for the exponent, and 23 bits for the mantissa. Also called single precision.



- IEEE dolgo realno: 64 bitov 1 bit za znak, 11 bitov za eksponent in 52 bitov za mantiso. Imenuje se tudi dvojna natančnost

- IEEE Long Real: 64 bits1 bit for the sign, 11 bits for the exponent, and 52 bits for the mantissa. Also called double precision

Plavajoča vejica / Floating point notation (2)



- Predznaknak binarne številke s plavajočo vejico predstavlja en bit. 1 bit pomeni negativno število, 0 bit pa pozitivno število.
- Eksponent lahko izračunamo iz bitov 24-31 tako, da odštejemo 127
- Mantisa (znana tudi kot signifikand ali ulomek) je shranjena v bitih 1-23. Nevidni vodilni bit (dejansko ni shranjen) z vrednostjo 1.0 je postavljen spredaj, nato ima bit 23 vrednost 1/2, bit 22 ima vrednost 1/4 itd
- **Podkoračitev**:
  Če eksponent doseže -127 (binarna 00000000 ali najmanjša vrednost (vse nič)), se upoštevajo posebna pravila za denormalizirane vrednosti. Vrednost eksponenta je nastavljena na 2^{-126} in "nevidni" vodilni bit za mantiso se ne uporablja več.

- The sign of a binary floating-point number is represented by a single bit. A 1 bit indicates a negative number, and a 0 bit indicates a positive number.
- The exponent can be computed from bits 24-31 by subtracting 127
- The mantissa (also known as significand or fraction) is stored in bits 1-23. **An invisible leading bi**t (i.e. it is not actually stored) **with value 1.0 is placed in front**, then bit 23 has a value of 1/2, bit 22 has value 1/4 etc.
- **Underflow:**
  If the exponent reaches -127 (binary 00000000 or minimum value (all zero)), special rules for denormalised values are followed. The exponent value is set to 2^{-126} and the "invisible" leading bit for the mantissa is no longer used.

Plavajoča vejica / Floating point notation (2)

- Kako pretvorimo 0xbf000000 v decimalni zapis? How do we convert 0xbf000000 to decimal notation?

```
hex: b    f    0    0    0    0    0    0
bin: 1011 1111 0000 0000 0000 0000 0000 0000

fp:  1    01111110 00000000000000000000000
     sign exponent mantissa

sign = 1 (-)
exponent = 126 - 127 = -1
mantissa = 1.0
result: -1.0 * 2^-1 = -0.5
```

Pretvorite / Convert

- Binarno v število s plavajočo vejico
- Binary to floating point

-1.11

+1101.101

-.00101

+100111.0

+.0000001101011

Vprašanja / Questions?