



**SVEUČILIŠTE U ZAGREBU**  
**Fakultet prometnih znanosti**  
**Zavod za inteligentne transportne sustave**  
**Vukelićeva 4, Zagreb, HRVATSKA**



# **Računalstvo**

## **Prikaz podataka i operatori**

**Doc. dr. sc. Edouard Ivanjko, dipl.ing.**

# Sadržaj

---

- Uvod
- Brojevni sustavi
- Prikaz podataka
- Aritmetički operatori
- Relacijski operatori

# Uvod

---

- Računala konstruirana da obrađuju numeričke podatke
- Svi podaci prikazani binarnim kôdom
  - Ovisno o tipu podatka isti binarni kôd ima drugačije značenje
    - Cijeli broj s predznakom
    - Decimalni broj s predznakom
    - Niz znakova
    - Multimedija (slika, zvuk, video)
- Binarni kôd se koristi zbog jednostavnog prikaza elektroničkim sklopovima
  - Nema napona -> „0”
  - Ima napona -> „1”

# Uvod

---

- Svaki podatak je potrebno kôdirati radi prikaza u računalu
- Načini kôdiranja su propisani normama
  - Univerzalno korištenje podataka
    - Različita računala
    - Različiti operacijski sustavi
  - Podrška u svim višim programskim jezicima
- Kod pisanja programa bitno odrediti kako podatak prikazati numerički
  - Potrebna pretvorba podatka
  - Dio potrebnih pretvorbi računalu obavlja automatski

# Brojevni sustavi

---

- Danas koristimo položajne brojevne sustave
  - Nazivaju se i pozicijski brojevni sustavi
  - Položaj znamenke ( $a$ ) definira njen doprinos vrijednosti broja
- Brojevni sustav definiran
  - Pravilom kôdiranja vrijednosti broja
  - Bazom ( $b$ )
  - Znamenakama ( $Z$ )
    - Svaki brojevni sustav ima znamenaka ( $N_z$ ) koliko mu iznosi baza
    - Svaka znamenka ima svoj vlastiti jednoznačni simbol

$$Z = \{z_1, \dots, z_{N_z}\} = \{0, 1, \dots, b - 1\}$$

# Brojevni sustavi

- Uz broj se navodi i njegova baza
- Za prikaz brojeva koriste se znamenke poredane da tvore željeni broj
  - Doprinos znamenke se dobiva množenjem znamenke broja ( $a_i$ ) i potencijom baze brojevnog sustava ( $b$ )
    - Desno od decimalnog zareza negativne potencije
    - Lijevo od decimalnog zareza pozitivne potencije
      - Prva znamenka lijevo od decimalnog zareza ima potenciju doprinosa baze 0

$$\begin{aligned}
 P &= a_2 a_1 a_0, a_{-1} a_{-2} b \\
 &= a_2 \cdot b^2 + a_1 \cdot b^1 + a_0 \cdot b^0 + a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2}
 \end{aligned}$$

# Brojevni sustavi – Raspon prikaza

- Raspon ( $R$ ) koji možemo prikazati ovisi o raspoloživoj količini memorije
  - Memorija definira najveću količinu znamenaka ( $n$ ) dostupnu za prikaz

$$R = [0, b^n - 1]$$

- Prikaz brojeva s predznakom smanjuje raspon
  - Predznak se također kôdira
  - Obično zauzima jedan bit
    - Početni, najznačajniji bit

# Brojevni sustavi – Dekadski sustav

- Baza dekadskog brojevnog sustava je 10
  - Najčešće se ne navodi
- Ljudima prirodan zbog broja prstiju
  - Osnovni brojevni sustav za pretvorbu u druge

- Sadrži 10 znamenaka

$$Z = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

- Najlakše ga se razumije na primjeru plaćanja
  - Dostupne samo kovanice i novčanice vrijednosti dobivene potenciranjem baze 10

$$P = 1234,56_{10}$$

$$= 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0 + 5 \cdot 10^{-1} + 6 \cdot 10^{-2}$$



# Brojevni sustavi – Binarni sustav

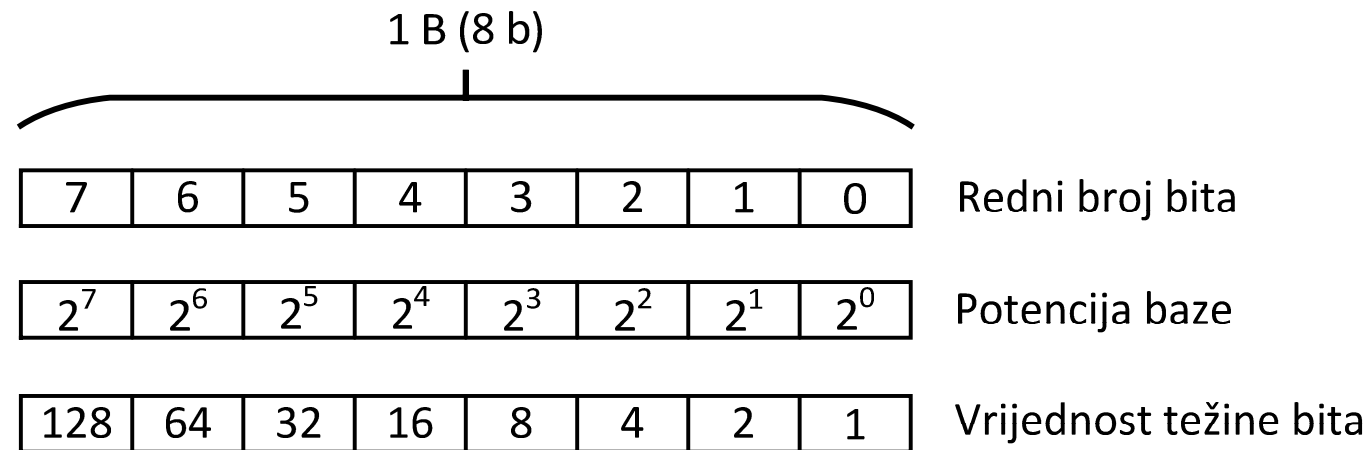
- Baza binarnog brojevnog sustava je 2
  - Služi za prikaz podataka u memoriji računala
- Sadrži dvije znamenke

$$Z = \{0, 1\}$$

- Pretvorba se radi uzastopnim cjelobrojnim dijeljenjem dekadskog broja novom bazom
  - Dijeljenje završava kada je rezultat 0
  - Ostatak dijeljenja je znamenka u novoj bazi
    - Nova baza je binarna -> ostatak dijeljenja 0 ili 1
  - Zadnji ostatak dijeljenja predstavlja najznačajniju znamenku

# Brojevni sustavi – Binarni sustav

- Vrijednost težina za 8-bitni binarni broj
  - Ovisi o položaju pojedinog bita
  - Množi se s vrijednošću pripadnog bita prilikom pretvorbe



- Analogno za druge položajne brojevne sustave
  - Potrebno promijeniti vrijednost baze

# Brojevni sustavi – Binarni sustav

- Primjer pretvorbe dekadskog broja 28 u binarni broj

– Procedura cjelobrojnog dijeljenja

$28 : 2 = 14$  *ostatak 0 najmanje značajna znamenka*

$14 : 2 = 7$  *ostatak 0*

$7 : 2 = 3$  *ostatak 1*

$3 : 2 = 1$  *ostatak 1*

$1 : 2 = 0$  *ostatak 1 najznačajnija znamenka*

– Kreiranje binarne znamenke

$$P = 28_{10} = 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 11100_2$$

# Brojevni sustavi – Oktalni sustav

- Baza oktalnog brojevnog sustava je 8
  - Služi za skraćeni prikaz podataka iz računala

- Sadrži osam znamenaka

$$Z = \{0, 1, 2, 3, 4, 5, 6, 7\}$$

- Pretvorba se radi uzastopnim cjelobrojnim dijeljenjem dekadskog broja novom bazom
  - Dijeljenje završava kada je rezultat 0
  - Ostatak dijeljenja je znamenka u novoj bazi
    - Nova baza je oktalna -> ostatak dijeljenja jedna od oktalnih znamenaka
  - Zadnji ostatak dijeljenja predstavlja najznačajniju znamenku



# Brojevnici sustavi – Oktalni sustav

- Primjer pretvorbe dekadskog broja 28 u oktalni broj
  - Procedura cjelobrojnog dijeljenja

$28 : 8 = 3$  *ostatak 4 najmanje značajna znamenka*

$3 : 8 = 0$  *ostatak 3 najznačajnija znamenka*

- Kreiranje oktalne znamenke

$$P = 28_{10} = 3 \cdot 8^1 + 4 \cdot 8^0 = 34_8$$

# Brojevni sustavi – Oktalni sustav

- Binarni broj se može brzo pretvoriti u oktalni grupiranjem bitova u skupine od tri bita
  - Počinje se od najmanje značajnih bitova
  - Vrijedi i obrnuti smjer pretvorbe

Dekadska vrijednost	Binarni kôd	Oktalna znamenka
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

# Brojevni sustavi – Oktalni sustav

- Primjer pretvorbe dekadskog broja 75 u oktalni broj korištenjem binarnog broja
  - Pretvorba u binarni broj

$$75 : 2 = 37 \text{ ostatak } 1 \text{ najmanje značajna znamenka}$$

$$37 : 2 = 18 \text{ ostatak } 1$$

$$18 : 2 = 9 \text{ ostatak } 0$$

$$9 : 2 = 4 \text{ ostatak } 1$$

$$4 : 2 = 2 \text{ ostatak } 0$$

$$2 : 2 = 1 \text{ ostatak } 0$$

$$1 : 2 = 0 \text{ ostatak } 1 \text{ najznačajnija znamenka}$$

- Pretvorba u oktalni broj

$$P = 75_{10} = 1001011_2 = 1\ 001\ 011_2 = 001\ 001\ 011_2 = 113_8$$

# Brojevni sustavi – Heksadecimalni sustav

- Baza heksadecimalnog brojevnog sustava je 16
  - Služi za skraćeni prikaz podataka iz računala
  - Kao oznaka baze pretvorenog broja se koristi i „h”
- Sadrži šesnaest znamenaka
  - Dio znamenaka (10 do 15) se označava slovima A do F
- Pretvorba se radi uzastopnim cjelobrojnim dijeljenjem dekadskog broja novom bazom
  - Dijeljenje završava kada je rezultat 0
  - Ostatak dijeljenja je znamenka u novoj bazi
    - Nova baza je heksadecimalna -> ostatak dijeljenja jedna od heksadecimalnih znamenaka
  - Zadnji ostatak dijeljenja predstavlja najznačajniju znamenku



# Brojevni sustavi – Heksadecimalni sustav

- Primjer pretvorbe dekadskog broja 28 u heksadecimalni broj

## – Procedura dijeljenja

$28 : 16 = 1$  ostatak 12 najmanje značajna znamenka

$1 : 16 = 0$  ostatak 1 najznačajnija znamenka

## – Kreiranje heksadecimalne znamenke

$$P = 28_{10} = 1 \cdot 16^1 + 12 \cdot 16^0 = 1 \cdot 16^1 + C \cdot 16^0 = 1C_{16}$$

# Brojevni sustavi – Heksadecimalni sustav

- Binarni broj se može brzo pretvoriti u heksadecimalni grupiranjem bitova u skupine od četiri bita

$$- 2^4 = 16$$

Dekadska vrijednost	Binarni kôd	Heksadecimalna znamenka
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

# Brojevni sustavi – Heksadecimalni sustav

- Primjer pretvorbe heksadecimalnog broja  $A10_{16}$  u dekadski korištenjem binarnog broja
  - Pretvorba u binarni broj

$$\begin{aligned}
 P &= A10_{16} \\
 &= 1010\ 0001\ 0000_2 \\
 &= 101000010000_2
 \end{aligned}$$

- Pretvorba u dekadski broj

$$\begin{aligned}
 P &= 101000010000_2 \\
 &= 1 \cdot 2^{11} + 1 \cdot 2^9 + 1 \cdot 2^4 \\
 &= 2576_{10}
 \end{aligned}$$

# Prikaz podataka

---

- Osnovni prikaz podataka se u računalu dijeli
  - Cijeli brojevi
  - Cijeli brojevi s predznakom
  - Decimalni brojevi
  - Znakovi
  - Slike
- Osnova za prikaz svake vrste podataka je binarni kôd

# Prikaz podataka – Cijeli brojevi

- Za prikaz cijelih brojeva se koriste ranije objašnjeni brojevnici sustavi
  - Osnovni prikaz je bez predznaka
  - U višim programskim jezicima se ovaj tip označava prefiksom „u” (engl. „unsigned”)
- Ovisno o rasponu koristi se potreban broj bajtova
  - Uobičajeno se koristi 2, 4 i 8 bajtova
  - Količina brojeva (kombinacija) koja se može prikazati iznosi  $2^n$ 
    - Pri tome je  $n$  broj bitova korišten za prikaz
    - Najveći broj je  $2^n - 1$

# Prikaz podataka – Cijeli brojevi s predznakom

- Predznak se kôdira najznačajnijim bitom
  - Vrijednost „0” označava pozitivan broj
  - Vrijednost „1” označava negativan broj
- Kôdiranje predznaka smanjuje mogući raspon prikaza za jedan bit

$$R = [-b^{n-1}, b^{n-1} - 1]$$

- Za prikaz cijelih brojeva s predznakom se najčešće koristi dvojni komplement
  - Kod pretvorbe bitno poznavati količinu memorije koja se koristi za prikaz broja
  - Suma istog pozitivnog i negativnog broja daje nulu

# Prikaz podataka – Cijeli brojevi s predznakom

- Procedura pretvorbe negativnog dekadskog cijelog broja u prikaz dvojnim komplementom
  - Provjera raspona raspoložive memorije
  - Pripadni pozitivan broj se pretvori u binarni
  - Binarni broj se proširuje vodećim nulama da se popuni cijela raspoloživa memorija
  - Prošireni binarni broj se invertira
    - Pretvorba „0” u „1” i obrnuto
  - Invertiranom binarnom broju se dodaje 1

# Prikaz podataka – Cijeli brojevi s predznakom

- Procedura pretvorbe binarnog broja u dvojnomo komplement u pripadni dekadski broj
  - Provjera predznaka binarnog broja
  - Pozitivan binaran broj
    - Samo se pretvori u dekadski primjenom težinske vrijednosti pojedinog bita
  - Negativan binaran broj
    - Prvo se oduzme vrijednost 1
    - Dobiveni binarni broj se invertira
      - Pretvorba „0” u „1” i obrnuto
    - Invertirani binarni broj se pretvori u dekadski primjenom težinske vrijednosti pojedinog bita
    - Dobivenom dekadskom broju se doda negativan predznak



# Prikaz podataka – Cijeli brojevi s predznakom

- Primjer pretvorbe dekadskog broja  $-37_{10}$  u 8-bitni binarni broj korištenjem dvojnog komplementa

- Pretvorba pozitivnog broja u binarni

$$37_{10} = 100101_2$$

- Kreiranje negativnog binarnog broja

$$\begin{array}{r}
 37 \quad \boxed{0\ 0\ 1\ 0\ 0\ 1\ 0\ 1} \\
 \quad \boxed{1\ 1\ 0\ 1\ 1\ 0\ 1\ 0} \\
 -37 \quad + \quad \boxed{1\ 1\ 0\ 1\ 1\ 0\ 1\ 1}
 \end{array}$$

- Provjera

$$\begin{array}{r}
 -37 \quad \boxed{1\ 1\ 0\ 1\ 1\ 0\ 1\ 1} \\
 + \\
 37 \quad \boxed{0\ 0\ 1\ 0\ 0\ 1\ 0\ 1} \\
 1 \quad \boxed{0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}
 \end{array}$$

# Prikaz podataka – Cijeli brojevi s predznakom

- Primjer pretvorbe binarnog broja  $11011011_2$  prikazanog 8-bitnim dvojnim komplementom u dekadski broj

– Provjera predznaka binarnog broja

- Najznačajniji bit ima vrijednost 1 -> negativan broj

– Oduzimanje vrijednosti 1 i invertiranje

$$\begin{array}{r}
 \boxed{1} \boxed{1} \boxed{0} \boxed{1} \boxed{1} \boxed{0} \boxed{1} \boxed{1} \\
 - \phantom{000000} 1 \\
 \hline
 \boxed{1} \boxed{1} \boxed{0} \boxed{1} \boxed{1} \boxed{0} \boxed{1} \boxed{0} \\
 \boxed{0} \boxed{0} \boxed{1} \boxed{0} \boxed{0} \boxed{1} \boxed{0} \boxed{1}
 \end{array}$$

– Pretvorba dobivenog binarnog broja u dekadski

$$100101_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 4 + 1 = 37_{10}$$

– Dodavanjem negativnog predznaka dobiva se  $-37_{10}$

# Prikaz podataka – Cijeli brojevi s predznakom

- Primjer prikaza cijelih brojeva s predznakom u dvojnomo komplementu za slučaj 3 bita
  - Raspon koji se može prikazati

$$R = [-2^{n-1}, 2^{n-1}-1] = [-2^{3-1}, 2^{3-1}-1] = [-2^2, 2^2-1] = [-4, 3]$$

## – Problem preljeva

- Najveća vrijednost + 1 -> Najmanja vrijednost

$$011_2 + 001_2 = 100_2$$

## – Problem podljeva

- Najmanja vrijednost - 1 -> Najveća vrijednost

$$100_2 - 001_2 = 011_2$$

# Prikaz podataka – Cijeli brojevi s predznakom

- Primjer prikaza cijelih brojeva s predznakom u dvojnomo komplementu za slučaj 3 bita
  - U C# je dvojni komplement osnova za prikaz cijelih brojeva s predznakom pomoću tipova short, int i long

Binarni kôd	Dekadska vrijednost
000	+0
001	+1
010	+2
011	+3
100	-4
101	-3
110	-2
111	-1

# Prikaz podataka – Decimalni brojevi

- Prikaz se može napraviti na dva načina
  - Odvojeni prikaz cijelog i decimalnog dijela broja
    - Broj se prvo podjeli na
      - $P = \text{cijeliDio} + \text{decimalniDio}$
    - Cijeli dio broja se pretvara dijeljenjem s bazom
    - Decimalni dio broja se pretvara množenjem s bazom
  - Zapis velikog broja normom IEEE754
    - Koristi se normalizirani eksponencijalni zapis
      - Ima uvijek vodeću jedinicu koja se ne sprema u memoriju
    - Broj se prikazuje
      - Predznakom
      - Mantisom
      - Binarnim eksponentom

# Prikaz podataka – Decimalni brojevi

---

- Odvojeni prikaz cijelog i decimalnog dijela broja
  - Decimalni dio broja se množi novom bazom
    - Dok rezultat množenja nije jednak nuli
    - Dok se ne dobije dovoljan broj decimalnih mjesta
      - Ne mogu se svi decimalni brojevi točno prikazati u novoj bazi
      - Pojava numeričke nestabilnosti u računalnim simulacijama
  - Cijeli dio rezultata množenja predstavlja novu decimalnu znamenku

# Prikaz podataka – Decimalni brojevi

- Odvojeni prikaz cijelog i decimalnog dijela broja
  - Primjer pretvorbe dekadskog decimalnog broja  $0,125_{10}$  u binarni broj

- **Procedura pretvorbe**

$$0,125 \cdot 2 = 0,25 \quad \text{cijeli dio } 0 \quad \text{najznačajnija znamenka}$$

$$0,25 \cdot 2 = 0,5 \quad \text{cijeli dio } 0$$

$$0,5 \cdot 2 = 1,0 \quad \text{cijeli dio } 1 \quad \text{najmanje značajna znamenka}$$

- **Kreiranje binarnog broja**

$$P = 0,125_{10} = 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 0,001_2$$

# Prikaz podataka – Decimalni brojevi

- Odvojeni prikaz cijelog i decimalnog dijela broja
  - Primjer pretvorbe dekadskog decimalnog broja  $0,6_{10}$  u binarni broj

- Procedura pretvorbe

$$0,6 \cdot 2 = 1,2 \quad \text{cijeli dio} \quad 1 \quad \text{najznačajnija znamenka}$$

$$0,2 \cdot 2 = 0,4 \quad \text{cijeli dio} \quad 0$$

$$0,4 \cdot 2 = 0,8 \quad \text{cijeli dio} \quad 0$$

$$0,8 \cdot 2 = 1,6 \quad \text{cijeli dio} \quad 1$$

$$0,6 \cdot 2 = 1,2 \quad \text{cijeli dio} \quad 1 \quad \text{dobiven početni broj}$$

- Kreiranje binarnog broja

$$\begin{aligned} P = 0,6_{10} &= 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} + \dots \\ &= 0,10011001\dots_2 \end{aligned}$$



# Prikaz podataka – Decimalni brojevi

- Norma IEEE754
  - Kod pretvorbe velikih brojeva nastaju dugački nizovi binarnih znamenaka
  - Radi smanjivanja duljine se koristi normalizirani eksponencijalni zapis binarnog broja

Binarni zapis	Normalizirani eksponencijalni zapis
$9_{10} = 1001_2$	$1,001 \cdot 2^{+3}$
$2,5_{10} = 10,1_2$	$1,01 \cdot 2^{+1}$
$0,1875_{10} = 0,0011_2$	$1,1 \cdot 2^{-3}$

# Prikaz podataka – Decimalni brojevi

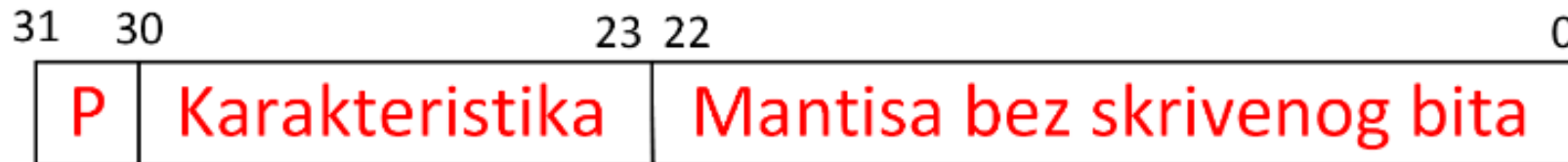
- Norma IEEE754
  - Normalizirani eksponencijalni prikaz binarnog broja ima uvijek vodeću jedinicu
    - Nije ju potrebno spremi u memoriju (skriveni bit)
    - Štedi se jedan bit
  - Daljnje poboljšanje prikaza se radi uvođenjem mantise i binarnog eksponenta

$$1001,011 = \underbrace{1,001011}_{\text{mantisa}} \cdot 2^{\text{binarni eksponent}}$$

↓
skriveni bit

# Prikaz podataka – Decimalni brojevi

- Norma IEEE754
  - Normalizirani eksponencijalni prikaz osnova norme IEEE754
    - Moguć prikaz širokog raspona realnih brojeva
  - Definicija norme IEEE754 za jednostruku preciznost



- 1 bit za pohranu predznaka
- 8 bitova za pohranu karakteristike
- 23 bita za pohranu mantise bez skrivenog bita
- U programskom jeziku C# jednostruka preciznost označava tip podatka float

# Prikaz podataka – Decimalni brojevi

- Norma IEEE754



- P oznaka predznaka

- 1 -> negativan broj
- 0 -> pozitivan broj

- BE oznaka binarnog eksponenta normaliziranog broja ( $BE \in [-126, 127]$ )

- K oznaka karakteristike ( $K \in [0, 255]$ )

- Vrijednosti 0 i 255 označavaju posebne slučajeve
- Prikaz negativnog i pozitivnog binarnog eksponenta

$$K = BE + 127$$

# Prikaz podataka – Decimalni brojevi

- Norma IEEE754
  - F je oznaka mantise bez skrivenog bita
  - Posebni slučajevi
    - Ništica uz  $K = 0$  i  $F = 0$
    - Preljev uz  $K = 255$  i  $F = 0$
    - Podljev uz  $K = 0$  i  $F \neq 0$
    - Neispravan broj uz  $K = 255$  i  $F \neq 0$ 
      - Oznaka NaN (engl. „Not a Number“)
  - Ispravno pretvoren broj dan izrazom
$$X = (-1)^P \cdot 1, F \cdot 2^{BE}$$
  - Detalji se mogu vidjeti na poveznici  
<http://babbage.cs.qc.edu/courses/cs341/IEEE-754.html>



# Prikaz podataka – Decimalni brojevi

- Hrvatski pravopis koristi decimalni zarez
- Programski jezici koriste decimalnu točku
- Programski jezik C# koristi tipove podataka
  - Jednostruka preciznost -> float
  - Dvostruka preciznost -> double
  - Točniji prikaz decimalnih mjesta -> decimal

Ključna riječ	Količina memorije (bit/bajt)	Najmanja vrijednost	Najveća vrijednost
float	32b / 4B	$1,5 \cdot 10^{-45}$	$3,4 \cdot 10^{38}$
double	64b / 8B	$5,0 \cdot 10^{-324}$	$1,7 \cdot 10^{308}$
decimal	128b / 16B	$1,0 \cdot 10^{-28}$	$7,9 \cdot 10^{28}$

# Prikaz podataka – Znakovi

---

- Računala obrađuju i nenumeričke (simboličke) podatke
  - Tekst, slike, zvuk, video snimke
- Simbolički podaci se također prikazuju binarnim kôdom
  - Pomoću  $n$  bitova prikaz  $2^n$  različitih znakova
- Za prikaz znakova odnosno slova koriste se
  - ASCII kôd (engl. „American Standard Code for Information Interchange“)
  - UNICODE (engl. „UNIversal CODE“)

# Prikaz podataka – Znakovi

- ASCII
  - 8-bitni američki standardni kôd za razmjenu informacija
    - Prikaz 256 znaka
  - Zapis slova abecede, znamenki i posebnih znakova

Skupina znakova	Numerička vrijednost (kôd)
Mala slova engleske abecede ( <i>a...z</i> )	97...122
Velika slova engleske abecede ( <i>A...Z</i> )	65...90
Dekadske znamenke 0...9	48...57
Interpunkcijski znakovi	32...47, 58...64 91...96, 123...126
Specijalni/upravljački znakovi	0...31 i 127



# Prikaz podataka – Znakovi

- ASCII
  - Svaki znak ima svoj numerički kôd (vrijednost)
  - Znakovi prikazani u tablici
    - BNMT – bitovi manje težine
    - BNVT – bitovi veće težine

BNMT / BNVT	0010	0011	0100	0101	0110	0111
0000	SP	0	@	P		p
0001	!	1	A	Q	a	q
0010	“	2	B	R	b	r
0011	#	3	C	S	c	s
0100	\$	4	D	T	d	t
0101	%	5	E	U	e	u
0110	&	6	F	V	f	v
0111	/	7	G	W	g	w
1000	(	8	H	X	h	x
1001	)	9	I	Y	i	y
1010	*	:	J	Z	j	z
1011	+	;	K	[	k	{
1100	,	<	L	\	l	
1101	-	=	M	]	m	}
1110	.	>	N	^	n	~
1111	/	?	O		o	DEL

# Prikaz podataka – Znakovi

---

- UNICODE
  - Nastao kao proširenje ASCII kôda
  - 16-bitni kôd
    - Ukupno prikaz 65.536 znakova
  - Kôdiranje slova abecede indoeuropskih (engleski, njemački, hrvatski) i ostalih skupina jezika
  - <http://unicode.org/>

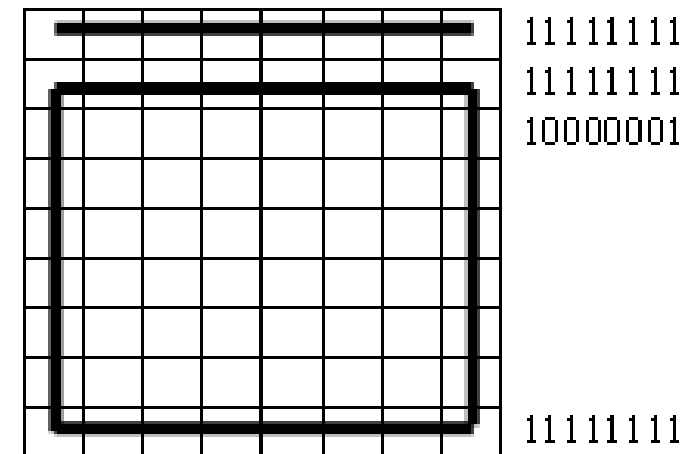
# Prikaz podataka – Slike

---

- Za prikaz slike u računalu se koristi podjela na osnovne elemente slike
  - Mali kvadratić pixel (engl. „Picture Element“)
- Svaki osnovni element (pixel) sadrži broj
  - Definira svjetlinu i boju
- Slika se u memoriju računala sprema po recima
  - Brojevi se zapisuju redak po redak u nizu
- Vrste slika
  - Crno-bijela
  - Siva slika
  - Slika u boji

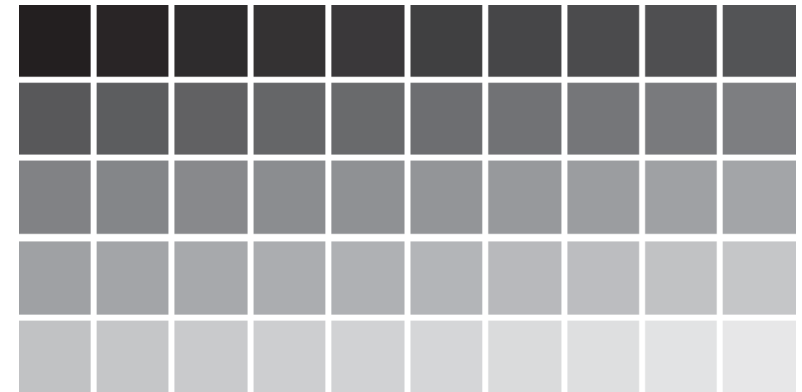
# Prikaz podataka – Slike

- Crno-bijela slika
  - Naziva se i binarna slika
    - Za spremanje slike potreban jedan bit po pixelu
  - Osnovni element slike može biti samo u dvije boje
    - Crna boja -> vrijednost 1
    - Bijela boja -> vrijednost 0
  - Početni prikaz slika u računalu
    - Danas se koristi radi ubrzavanja obrade slike u računalu
      - Računalni vid
      - Stvarnovremenska obrada
        - » Prepoznavanje vozila
        - » Prepoznavanje registarskih oznaka



# Prikaz podataka – Slike

- Siva slika
  - Svakom pixelu pridružena veća količina memorije
    - Omogućen prikaz nijansi sive boje
    - Npr. 8 bita -> 256 nijansi sive boje
  - Količina memorije potrebna za spremanje slike jednaka umnošku broja pixela (rezolucije) i količine memorije po jednom pixelu
    - Veća količina memorije po pixelu povećava broj nijansi, ali i potrebnu količinu memorije



# Prikaz podataka – Slike

- Slika u boji
  - Postoji nekoliko formata prikaza boje
  - Najčešće se koristi RGB format
    - Prikaz slike u boji miješanjem tri osnovne boje
      - Crvena (engl. „Red)
      - Zelena (engl. „Green“)
      - Plava (engl. „Blue“)
    - Potrebna boja se odabire iz jedne od  $2^n$  kombinacija
    - Danas se koristi „True color“ prikaz
      - Najmanje 8 bita po osnovnoj boji
      - Uz 8 bita ukupno  $3 \cdot 8 = 24$  bita po elementu slike
        - » Ukupno 16.777.216 različitih boja
        - » Ljudsko oko razaznaje do 10.000.000 različitih boja



# Prikaz podataka – Slike

---

- Primjer količine memorije potrebne za prikaz slike rezolucije 1.024 x 768

- Crno-bijela slika

$$\text{Memorija} = 1.024 \cdot 768 \cdot 1\text{b} = 783.360\text{b} = 97.920 \text{ B} = 96,625 \text{ kB}$$

- 8-bitna siva slike rezolucije 1.024 x 768

$$\text{Memorija} = 1.024 \cdot 768 \cdot 8\text{b} = 6.291.456\text{b} = 786.432 \text{ B} = 768 \text{ kB}$$

- 8-bitna RGB slika u boji

$$\begin{aligned} \text{Memorija} &= 1.024 \cdot 768 \cdot 3 \cdot 8\text{b} = 18.874.368\text{b} = 2.359.296 \text{ B} \\ &= 2.304 \text{ kB} = 2,25 \text{ MB} \end{aligned}$$

# Prikaz podataka – Slike

---

- Opisani prikaz slike predstavlja rasterski prikaz slike
  - Svaki pixel se sprema zasebno
  - Kvaliteta ovisi o gustoći mreže elemenata (pixela)
    - Gušća mreža, bolja razlučivost (slika oštija), duži zapis
- Slike koje sadrže jednostavne i geometrijski pravilna tijela predstavljaju se pomoću koordinata i funkcija -> vektorski prikaz
  - Kvadrat definiran duljinom stranice i koordinatom jednog vrha
  - Kružnica definirana polumjerom i koordinatom središta



# Aritmetički operatori

---

- Aritmetički operatori definiraju osnovne matematičke operacije
  - Zbrajanje
  - Oduzimanje
  - Množenje
  - Dijeljenje
    - Dijeljenje decimalnih brojeva
    - Dijeljene cijelih brojeva
    - Ostatak dijeljenja cijelih brojeva
- Definicija operacije uvijek ista, a oznaka operatora ovisi o programskom jeziku

# Aritmetički operatori

Operator	Raptor	C#
Množenje	broj1 * broj2	broj1 * broj2
Dijeljenje	broj1 / broj2	broj1 / broj2 (tip podatka float, double, decimal)
Cjelobrojno dijeljenje	<b>floor</b> ( broj1 / broj2 )	broj1 / broj2 (tip podatka short, int, long)
Ostatak cjelobrojnog dijeljenja	broj1 <b>REM</b> broj2 broj1 <b>MOD</b> broj2	broj1 % broj2
Potencija	broj1 ^ broj2 broj1 ** broj2	<b>Math.Pow</b> ( broj1 , broj2 )
Drugi korijen	<b>SQRT</b> ( broj )	<b>Math.Sqrt</b> ( broj )
Apsolutna vrijednost	<b>abs</b> ( broj )	<b>Math.Abs</b> ( broj )
Prirodni logaritam	<b>log</b> ( broj )	<b>Math.Log</b> ( broj )
Najmanja vrijednost	<b>min</b> ( broj1, broj2 )	<b>Math.Min</b> ( broj1, broj2 )
Najveća vrijednost	<b>max</b> ( broj1, broj2 )	<b>Math.Max</b> ( broj1, broj2 )

# Relacijski operatori

---

- Relacijski operatori uspoređuju dvije vrijednosti
- Rezultat usporedbe je logička vrijednost
  - Relacija je istinita
    - Logička vrijednost 1, DA, engl. „true”
  - Relacija nije istinita (lažna)
    - Logička vrijednost 0, NE, engl. „false”
- Definicija operatora uvijek ista, a oznaka operatora ovisi o programskom jeziku

# Relacijski operatori

- U jednoj relaciji moguće usporediti najviše dvije varijable
  - Za usporedbu više varijabli koriste se zagrade i logički operatori

Operator	Raptor	C#
Identično	=	==
Različito	/= !=	!=
Veće	>	>
Veće ili jednako	>=	>=
Manje	<	<
Manje ili jednako	<=	<=