

**Muharem Kozić**

# **S K R I P T A**

**UVOD U INFORMATIKU I RAČUNARSTVO**

- Brojni sistemi
- Kod i kodiranje
- Računarska aritmetika
- Računarska logika
- Algoritmi

Mostar, 2006



---

# UVOD U INFORMATIKU I RAČUNARSTVO

## Pojam informatike kao nauke

Informatika je jedna od najmlađih ali i jedna od najkompleksnijih disciplina današnjice. Ona je danas i praktična djelatnost i nauka savremenog tipa, pri čemu nije uvijek lako odrediti granicu između naučnog i praktičnog. Postoji već nekoliko decenija ali njeno ime je utemeljeno tek 1962. Stvorio ga je Filip Drefis (Philippe Dreyfus) tako što je spojio prva dva sloga francuske riječi information (Enformasion) i posljednja dva sloga riječi automatique (otomatik). Na engleskom govornom području koriste se termini Computer science i Information systems

Informatika je u mnogo čemu povezana i sa ostalim naukama kao što su: matematika, formalna logika, teorija informacija, elektronika i dr.

*Francuska akademija nauka je 1966.godine definisala informatiku kako slijedi:*

*“Nauka o racionalnoj obradi informacija, prije svega pomoću automatskih mašina, time da se informacija smatra nosiocem ljudskog znanja i komunikacija u oblasti tehnike, ekonomije i drugih nauka”.*

Drugim riječima informatika je nauka koja se bavi smisaonom obradom informacija. Informacije su u stvari činjenice o stvarima i događajima iz svih oblasti ljudskog bitisanja. Naglašeno je da se informacije obrađuju pomoću automatskih mašina. Te mašine su, naravno, kompjuteri ili računari. Oblasti primjene informatike su gotovo nemjerljive. Informatiku srećemo svugdje, u preduzećima, školama, bankama, fabrikama, svakodnevnom životu itd.

---

Prema nekim procjenama oko 80% informacionih sistema služi za obradu podataka u ekonomsko-komercijalne svrhe. Istorijski gledano ključne oblasti poslovanja o čijoj je kvaliteti ovisila uspješnost čitavog preduzeća bile su finansije, nabava, proizvodnja i prodaja i na kraju informatika (informacioni sistemi za obradu poslovnih podataka i informacija)

Važnost informacionih sistema ima znatno drugačiju ulogu nego prva 4 nabrojana faktora. Prvo zbog toga što informacija u principu ne podliježe zakonima materije i energije ( ne troši se korištenjem, ne smanjuje se raspodjelom), drugo zbog toga što koncepcija kreiranja jedinstvenog inf. sistema bitno kvalitetnije povezuje pojedine podsisteme preduzeća uzajamno.

Informacioni sistem se bazira na sistematično uspostavljenom i primjenjivom skupu pravila u vezi s nosiocima zadatka informisanja. U okviru svakog informacionog sistema mogu se definisati i razlikovati 4 osnovne aktivnosti:

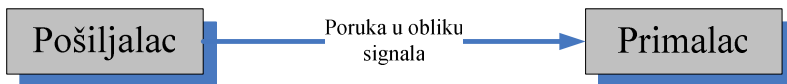
1. Prikupljanje podataka
2. Obrada podataka,
3. Memorisanje (čuvanje) podataka
4. Distribuiranje podataka korisnicima

## **Pojam informacije, podatka i informatike**

Informacije postoje od pamtivjeka. Ljudi su od prahistorijskog doba razmjenjivali informacije putem znakova, glasova, dimnih signala, svjetlosti i sl. Iz niza glasova razvija se govor, pa pisana riječ a tek prije nekoliko vijekova štampana. Pisane i štampane riječi su se prenosile konjanicima, poštanskim kolima, raznim prevoznim sredstvima a zatim i elektronskim putem (telegraf, telefon, radio, tv, internet,...)

---

Poruke koje se šalju idu uvijek od pošaljioaca ka primaocu. Primalac na određeni način obrađuje poruku i na bazi dobijenih rezultata donosi odgovarajuće odluke.



Slika **Prenos poruke**

Poruku čini niz podataka koji primaoca poruke potiču na neku akciju. U svakodnevnom životu se pojam informacija često poistovjećuje sa pojmom podatak. U informatičkoj terminologiji ta dva pojma se razlikuju. U opštem slučaju podatak je zapis o nekoj pojavi iz okoline. Opis svake pojave predstavljen je skupom znakova npr. "A", "U", "T" i "O" je skup znakova koji označava pojavu auto (automobil). Pojave se često razlikuju po nekim osobinama **atributima** (npr boja, broj sjedišta, broj vrata...) koji se također mogu predstaviti nekim znakovima. Na kraju atributi mogu imati različitu vrijednost (npr. plava boja, crvena boja, troja vrata, petora vrata, ...) Kada se spoje **pojava, atributi i vrijednosti atributa** nastaje **informacija**. Ako takva informacija pozitivno utiče na odlučivanje onda je to **korisna informacija**. Danas je sve veći broj ljudi koji su zaposleni na poslovima obrade podataka. Nauka koja se bavi izučavanjem načina, metoda i sredstava za prikupljanje i obradu informacija naziva se **informatika**.

### **Računar i program**

Računar je uređaj koji na osnovu zadatog postupka i postojećih podataka rješava problem i daje odgovarajuće rezultate. Njegov zadatak je da na bazi određenih podataka pruži korisniku informacije neophodne za donošenje odluka.

Postupak rješavanja problema na računaru naziva se program. U programu se navode načini na koji će se obezbijediti unos podataka potrebnih za odvijanje programa, načini obrade i ispisa rezultata.

---

Danas je aktuelan trend da se pod informatikom podrazumijeva proučavajuće računara kao savremenih sredstava za obradu informacija, sa jedne strane, i primjena ovih računara u raznim drugim oblastima u realizaciji tzv. računarski baziranih obrada podataka i informacionih sistema, s druge.

Bez računara nema ni informatike. Za računare su vezana dva vrlo bitna pojma *Hardware* (Hardver) i *Software* (Softver).

**Hardware** je, u stvari, sva računarska oprema. Na primjer CPU, matična ploča, grafička kartica, RAM, ROM, tvrdi disk, disketna jedinica, CD ROM, tastatura, monitor, miš, štampač, skener, modem,...

Nizovi operacionih instrukcija (koji se zovu programima) koji upravljaju kompjuterskom obradom informacija i kontrolišu je zovu se *softver*.

**Softver** (software) su svi programi i naredbe koji omogućavaju rad računara. On upravlja računarm. Njime pokrećemo i prekidamo rad računara. U širem smislu u softver pripadaju i svi projektantsko-razvojni alati za izgradnju informacijskih sistema.

### **Softver obuhvata:**

**Sistemske softver**, kao što je operativni sistem, koji kontrolišu i podržava funkcije kompjuterskog sistema.

**Aplikativni softver** koji se sastoji iz programa koji upravljaju konkretnim poslovima obrade podataka.

### **Oblasti primjene aplikativnih softvera:**

Baze podataka (Access, ORACLE, SQL, dBASE, FoxPro...)

Inženjerstvo (AutoCAD, Matlab, ...)

Obrada teksta i stono izdavaštvo (Word, ...)

Tablični proračuni (Excel, Lotus,...)

---

Obrada grafike (Corel, PhotoShop,...)

Multimedia

Internet aplikacije

## **RAČUNARSKI SISTEM**

Pod računarskim sistemom, a susretat ćemo i nazive: sistem za obradu podataka, automatizirani sistem za obradu podataka i sl., mislimo na sve ono što obavlja poslove rješavanja zadataka na računarskoj opremi, od središnje jedinice računara, perifernih jedinica, svih programa koji omogućavaju da računar uopće radi i da rješava neki zadatak, te uposlenih, koji mu pišu programe ili ga uključuju u rad.

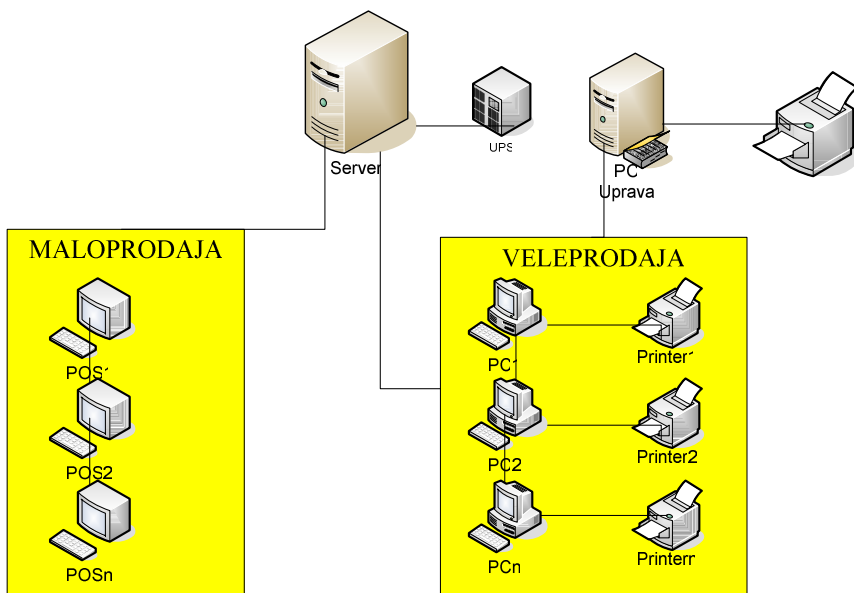
Na slici je prikazani su glavni dijelovi računarskog sistema: hardver, softver i lajfvver.

Lajfvver (lifeware) su kadrovi koji izrađuju programe i rukuju računarm.

Prilikom uvođenja u primjenu potrebno je uskladiti sve ove elemente s postojećom organizacijom u preduzeću ili ustanovi gdje se uvođenje obavlja.

Ova zavisnost definisana je na sljedeći način (Srića, 1991.):

Ako imate hardver pete generacije, softver četvrte, kadrove treće i organizaciju druge generacije, sistem će raditi u drugoj generaciji.



Slika Komponente računarskog sistema

Za razumijevanje rada računara, potrebno je pojasniti tri temeljna sistema na kojima se zasniva funkcioniranje računara:

- aritmetički sistem (brojni sistemi),
- algebarski sistem (algebra logike) i
- elektronsko čuvanje podataka.

## BROJNI SISTEMI

Brojni sistem je način označavanja ili izražavanja brojeva, nizova znakova ili naziva. Usporedo s razvojem pisma kroz čovjekovu istoriju razvijali su se i različiti brojni sistemi koji se po strukturi dijele na:

- 1.) aditivne,
- 2.) aditivno-multiplikativne.

Aditivni sistem je niz znakova u kojima je broj jednak zbroju znakova od kojih je sastavljen, npr. kao kod starih Rimljana:

$$\mathbf{XXXVII} = \mathbf{10} + \mathbf{10} + \mathbf{10} + \mathbf{5} + \mathbf{2} = \mathbf{37}$$

Ovakvi sistemi nisu omogućavali računске operacije kao što omogućavaju aditivno-multiplikativni brojni sistemi, kod kojih pojedini brojevi (znamenke) predočavaju veličinu pojedinih grupa datog niza s kojom se pomnože i sve grupe zbroje:

$$\mathbf{"stotinu četrdeset i pet"} = \mathbf{1} * \mathbf{100} + \mathbf{4} * \mathbf{10} + \mathbf{5} * \mathbf{1} = \mathbf{145}$$

Osnov aditivno-multiplikativnog brojnog sistema je BAZA, koja ulazi kao multiplikant u komponente oznake ili naziva broja. Danas je u općoj upotrebi DEKADNI BROJNI SISTEM, aditivno-multiplikativni brojni sistem s OSNOVOM (BAZOM) deset (10). Općenito se broj "N" u aditivno-multiplikativno sistemu s osnovom "B" može napisati u obliku:

$$N_B = a * B^n + a * B^{n-1} + \dots + a * B^2 + a * B^1 + a * B^0$$

- **N** je broj brojnog sistema s bazom "**B**" izražen brojem „**a**“,
- **a** je bilo koji znamenka brojnog sistema u opsegu od **0** do **B-1**, a to su u dekadnom sistemu znakovi 0,1,2,3,4,5,6,7,8 i 9 koji predstavljaju brojni raspon unutar baze „**B**“,
- **B** je baza (osnova) brojnog sistema, koja u dekadnom sistemu iznosi **10** i ukazuje da u sistemu veličine grupe

---

ima **10** različitih stanja na jednom mjestu za jednu znamenku „**a**“.

Navedenim izrazom izračunava se dekadna vrijednost broja "**N**" bilo kojeg brojnog sistema.

## **DEKADNI brojni sistem**

Ljudi broje i računaju po dekadnom brojnom sistemu i vrlo često ne razmišljaju da je nastao na osnovu deset čovjekovih prstiju s kojima se pomagao u računanju.

Koristi se poziciono označavanje brojeva npr. broj 1953 sadrži četiri znamenke od kojih svaka u ovisnosti o mjestu gdje se nalazi označava broj jedinica, desetica, stotica itd. Svakoj znamenki pridruži se njena TEŽINA koja ovisi o njenom mjestu u broju. Najmanju težinu ima znamenka na desnom kraju broja, a najveću težinu ima znamenka na lijevom kraju broja.

Dekadni broj tumači se na slijedeći način:

$$1953_{10} = 1 * 1000 + 9 * 100 + 5 * 10 + 3 * 1 = 1 * 10^3 * 9 * 10^2 + 5 * 10^1 + 3 * 10^0$$

### ***0,1,2,3 su težinske vrijednosti***

Osnova sistema je broj 10 a težinska vrijednost znamenke je eksponent osnove u skladu s udaljenosti znamenke od mjesta najmanje težine.

S negativnim eksponentom mogu se prikazati brojevi manji od jedan kao na primjer:

$$0,12 = 1 * 10^{-1} + 2 * 10^{-2}$$

Često se u svakodnevnoj praksi opisuju događaji kojima je osnov brojanja drugačiji, npr. sunca ima ili nema, živ ili mrtav, moker ili suh i slični. Tim opisima pridružena su DVA različita stanja.

---

Elektronika u tom pogledu nije iznimka. Vrlo je složen elektronički sklop koji bi amplitude signala razlikovao u 10 nivoa veličine. Sistem bi bio neotporan na svaku smetnju koja bi izmjenila veličinu amplitude.

Jednostavnije je definisati dvije situacije, impulsa ima (pozitivan impuls) ili ga nema (odsustvo ili negativan impuls). Simbolička oznaka postojanja impulsa je "1", a oznaka nepostojanja je "0". Sklop koji razlikuje postojanje i nepostojanje impulsa mnogo je jednostavniji, te se stoga računari dizajniraju da računске i logičke operacije vrše s brojnim sistemom koji koristi znamenke "0" i "1" i ima osnovu "2".

Takav sistem naziva se BINARNI BROJNI SISTEM u kojemu se na mjestu znamenke mogu pojaviti "0" ili "1", što predstavlja 50% vjerovatan događaj za pojavu jedne od znamenki. To znači da mjesto znamenke sadrži količinu informacije od 1 bit-a.

Nadalje u opisu za brojeve koji nemaju oznaku osnove podrazumjeva se osnova 10, a za brojeve drugih brojnih sistema označiti će im se pripadna osnova ili će se na kraju broja dopisati početno slovo sistema kojemu pripada (10B, 16H, ...).

Za slučaj decimalnog broja 1845,34 imamo puni zapis:

$$1845,34 = 1 * 10^3 + 8 * 10^2 + 4 * 10^1 + 5 * 10^0 + 3 * 10^{-1} + 4 * 10^{-2}$$

## **BINARNI brojni sistem**

Kod dekadnog brojnog sistema brojimo "nula, jedan, dva, tri, četiri, pet, šest, sedam, osam, devet, DESET", a "deset" je u suštini "0 jedan dalje". Analogno navedenom može se izgraditi binarni sistem brojeva prema primjeru u tablici.

Dekadno	Binarno	
0	0	
1	1	Slijedi „0 jedan dalje“
2	10	
3	11	Slijedi „00 jedan dalje“
4	100	
5	101	
6	110	
7	111	Slijedi „000 jedan dalje“
8	1000	
...	...	

Opći oblik za pretvaranje binarnog broja u dekadni je:

$$N_{10} = a * 2^n + a * 2^{n-1} + \dots + a * 2^2 + a * 2^1 + a * 2^0$$

- **N** je broj brojnog sistema izražen znamenkama „**a**“,
- **a** znamenke sistema: „**0**“ ili „**1**“,
- **2** je baza (osnova) brojnog sistema.

### Pretvaranje dekadnog broja u binarni broj

Pretvaranje dekadnog broja u binarni broj može se izvršiti na dva načina:

- *dijeljenjem s 2 ili*
- *pomoću tablica.*

Pretvaranje dijeljenjem sa dva je postupak koji se općenito može primijeniti za pretvaranje dekadnih brojeva, u brojeve bilo kojeg sistema, dijeljenjem sa osnovicom tog sistema.

Pretvaranje dijeljenjem s dva, vrši se sukcesivnim dijeljenjem s 2. Ostatak dijeljenjem predstavljaju brojke 0 ili 1. Kad se dijeljenjem dođe do operacije  $1 : 2 = 0$  i 1 ostatak, dijeljenje je završeno. Čitanje rezultata vrši se odozdo prema gore. Na ovaj način se vrši

pretvaranje cijelih brojeva dekadnog brojnog sistema u binarni brojni sistem.

**Primjeri:**

$$(125)_{10} = ( )_2$$

125	:	2	=	62	Ostatak	1	↑ Čitanje	
62	←	:	2	=	31	“		0
31	←	:	2	=	15	“		1
15	←	:	2	=	7	“		1
7	←	:	2	=	3	“		1
3	←	:	2	=	1	“		1
1	←	:	2	=	0	“		1

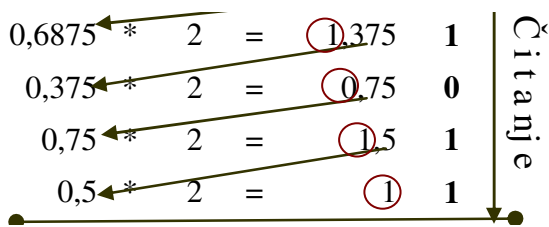
$$(125)_{10} = (1111101)_2$$

Pretvaranje decimalnih brojeva vrši se sukcesivnim množenjem s 2 i upisivanjem dobivenog cjelobrojnog dijela (0 ili 1) kao brojke binarnog broja.

Množenje se nastavlja dok se ne dobije rezultat u decimalnom dijelu = 0 ili dovoljno mala vrijednost.

**Primjer:**

$$(0,6875)_{10} = ( )_2$$



Smjer čitanja je odozgo prema dolje.

$$(0,6875)_{10} = (0,1011)_2$$

Pretvaranje mješovitih brojeva vrši se tako, da se prvo pretvori cjelobrojni dio broja po pravilima za pretvaranje cjelobrojnih brojeva, a zatim decimalni dio broja po pravilima za razlomljene brojeve.

Pretvaranje pomoću tablica vrši se korištenjem tablice koja predstavlja vrijednosti mjesta binarnog brojnog sistema.

Ako se ta vrijednost može prikazati kao  $2^n$ , gdje je  $n$  broj iz skupa prirodnih brojeva, onda se uzimanjem prve niže vrijednosti i prikazivanjem te vrijednosti u obliku  $2^n$ , a zatim pridruživanjem vrijednosti 1 za vrijednosti  $2^n$  koje su upotrebljene za prikaz tog broja odnosno 0, ako vrijednosti  $2^n$  nisu upotrebljene prilikom prikazivanja tog broja, može pretvoriti broj iz dekadnog u binarni oblik (Tablica koja slijedi).

Vrijednosti $2^n$	0	1	2	3	4	5	6	7	8	9
$2^n$	1	2	4	8	16	32	64	128	256	512

Primjer:

$$(156)_{10} = ( )_2$$

$$(156)_{10} = 128 + 28 = 2^7 + 16 + 12 = 2^7 + 2^4 + 8 + 4 = \\ = 2^7 + 2^4 + 2^3 + 2^2 =$$

$$\longrightarrow \textcircled{1} \textcircled{0} \textcircled{0} \textcircled{1} \textcircled{1} \textcircled{1} \textcircled{0} \textcircled{0}$$

$$128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1$$

$$1 \cdot 2^7 \quad 0 \cdot 2^6 \quad 0 \cdot 2^5 \quad 1 \cdot 2^4 \quad 1 \cdot 2^3 \quad 1 \cdot 2^2 \quad 0 \cdot 2^1 \quad 0 \cdot 2^0$$

$$(156)_{10} = (10011100)_2$$

### Pretvaranje binarnog broja u dekadni broj

Pretvaranje binarnih brojeva u dekadne, može se izvršiti na više načina. Jedan od postupaka je i zbrajanje mjesnih vrijednosti.

Primjer:

(	1	0	1	1	0	) <sub>2</sub>		broj
<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>				mjesne vrijednosti

$$16 \times 1 + 8 \times 0 + 4 \times 1 + 2 \times 1 + 0 \times 1 = 22$$

$$(10110)_2 = (22)_{10}$$

Primjer:

Broj  $(101100,11)_2$  u binarnom sistemu ima vrijednost:

$$(101100,11)_2 = 1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 0x2^0 + \\ 1x2^{-1} + 1x2^{-2} = (44,75)_{10}$$

Druga metoda je takozvani cik - cak postupak, koji je univerzalan za pretvaranje brojeva iz bilo kojeg brojnog sistema u dekadni.

Prava linija u ovom cik - cak postupku odnosno mreži, znači množenje, a kosa zbrajanje.

Primjer:

$$(10101000011)_2$$
$$(1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1)_2 = (1347)_{10}$$

2 2 2 2 2 2 2 2 2 2

### Aritmetika u binarnom brojnom sistemu

Pravila za osnovne računске operacije su:

*a) zbrajanje:*

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= 0 \text{ (1 prenos)} \end{aligned}$$

Binarno zbrajanje obavlja se isto kao i decimalno zbrajanje, osim što se prenos na sljedeće značajno mjesto obavlja nakon postignutog zbroja  $2(1+1)$ .

*b) oduzimanje:*

$$\begin{aligned} 0 - 0 &= 0 \\ 1 - 0 &= 1 \\ 1 - 1 &= 0 \\ 10 - 1 &= 1 \text{ (1 prenos)} \end{aligned}$$

Binarno oduzimanje obavlja se kao i decimalno oduzimanje, osim što se posuđuje 1 od bita veće težine.

**c) množenje:**

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

Kod binarnog množenja djelomičan umnožak pomiče se za jedno mjesto udesno po navedenim pravilima, a zatim se umnošci zbroje.

**d) dijeljenje:**

$$0 / 0 = \text{nedjeljivo}$$

$$1 / 0 = Y$$

$$0 / 1 = 0$$

$$1 / 1 = 1$$

Dijeljenje binarnih brojeva obavlja se na isti način kao i dekadnih.

Praktično se dijeljenje svodi na množenje i oduzimanje.

**Primjeri:**

**1. Zbrajanje:**

$$\begin{array}{r} \text{a) } 1001011_2 = 75_{10} \\ + 1011110_2 = 94_{10} \\ \hline 10101001_2 = 169_{10} \end{array}$$

$$\begin{array}{r} \text{b) } 1111001_2 = 121_{10} \\ + 110100_2 = 52_{10} \\ \hline 10101101_2 = 173_{10} \end{array}$$

**2. Množenje:**

$$\text{a) } \underline{11001} \times 1110 = 101011110_2 \quad (25 * 14=350)_{10}$$

$$\begin{array}{r} 11001 \\ 11001 \\ 11001 \\ \hline 00000 \end{array}$$

$$101011110 = 2^8 + 2^6 + 2^4 + 2^3 + 2^2 + 3^1 = 350$$

b)  $111 \times 1111 \equiv 1101001 \text{ (} 7 \cdot 15 = 105 \text{)}_{10}$

$$\begin{array}{r} 111 \\ 111 \\ 111 \\ \underline{111} \\ 1101001 = 2^6 + 2^5 + 2^3 + 2^0 = 105_{10} \end{array}$$

### 3. Oduzimanje:

$$\begin{array}{l} \text{a) } \begin{array}{r} 1110_2 = 14_{10} \\ - 1101_2 = 13_{10} \\ \hline 0001_2 = 1_{10} \end{array} \qquad \text{b) } \begin{array}{r} 1110_2 = 14_{10} \\ - 100_2 = 4_{10} \\ \hline 1010_2 = 10_{10} \end{array} \end{array}$$

### 4. Dijeljenje:

$11011 : 11 = 1001$  Provjera:  $27 : 3 = 9$

$$\begin{array}{r} - 11 \\ 00011 \\ \underline{- 11} \\ 00 \end{array}$$

## OKTALNI brojni sistem

Osim binarnog brojnog sistema u računarima se koristi i OKTALNI brojni sistem s bazom 8 i koji koristi osam znamenki dekadnog brojnog sistema i to znamenke 0,1,2,3,4,5,6 i 7. Brojevi ovog sistema prikazani su u narednoj tablici:

dekadno	0	1	2	3	4	5	6	7
oktalno	0	1	2	3	4	5	6	7
dekadno	8	9	10	11	12	13	14	15

oktalno	<b>10</b>	11	12	13	14	15	16	17
dekadno	<b>16</b>	17	18	19	20	21	22	23
oktalno	<b>20</b>	21	22	23	24	25	26	27

**Tabela Generiranje dekadnog i oktalnog niza cijelih brojeva.**

Opći oblik za pretvaranje oktalnog broja u dekadni je:

$$N_{10} = a * 8^n + a * 8^{n-1} + \dots + a * 8^2 + a * 8^1 + a * 8^0$$

- **N** je broj brojnog sistema izražen znamenkama „*a*“,
- **a** znamenke sistema: **0, 1, 2, 3, 4, 5, 6** ili **7**,
- **8** je baza (osnova) brojnog sistema.

### Pretvaranje dekadnog broja u oktalni broj

Pretvaranje dekadnog broja u oktalni broj može se vršiti univerzalnim postupkom, sukcesivnim dijeljenjem s bazom sistema. U ovom slučaju, to je broj 8:

Primjeri:

$$(1016)_{10} = ( )_8$$

1016	:	8	=	127	Ostatak	<b>0</b>	↑ Čitanje
127	:	8	=	15	“	<b>7</b>	
15	:	8	=	1	“	<b>7</b>	
1	:	8	=	0	“	<b>1</b>	

$$(1016)_{10} = (1770)_8$$

## Aritmetika u oktalnom brojnom sistemu

Operacije u oktalnoj aritmetici izvode se na sljedeći način:

### *Primjeri zbrajanja:*

$$\begin{array}{r|l} (1025)_8 & 5+6=11; 11 : 8 = 1 \text{ (prenos) } + 3 \text{ (piše se)} \\ + (536)_8 & (2+3)+1 \leftarrow 6; \text{ (manje od baze, samo se piše)} \\ \hline (1563)_8 & 0+5=5 \\ & 1+0=1 \end{array}$$

Zbrajanje u oktalnom brojnom sistemu obavlja se zbrajanjem znamenki kao i kod dekadskog brojnog sistema. Ukoliko je zbroj veći od 8 (baze) dijelimo ga sa bazom (8), te rezultat prenosimo za sljedeće zbrajanje znamenki, a ostatak predstavlja znamenku rezultata zbrajanja.

### *Primjeri oduzimanja:*

$$\begin{array}{r|l} (1025)_8 & 8+5=13; 13-6=7; \text{ (prenos } 1) \\ - (536)_8 & 1+3=4; 8+2=10; 10-4=6; \text{ (prenos } 1) \\ \hline (267)_8 & 1+5=6; 8+0=8; 8-6=2; \text{ prenos } 1 \\ & 1-1=0 \end{array}$$

Oduzimanje u oktalnom brojnom sistemu obavlja se kao i u dekadnom brojnom sistemu. Na primjeru je pojašnjena ova operacija. Prvo oduzimamo 5 - 6. S obzirom da je 5 manja vrijednost dodajemo bazu 8; 5 + 8 = 13. Sada oduzimamo 13 - 6 = 7 i bilježimo 1 prenos. Dodajemo ovaj prenos sljedećoj znamenki 3: 1+3=4. Sada oduzimamo 2-4, s obzirom da je 2 manje od 4 posuđujemo bazu 8 I imamo (8+2)-4 = 6 I bilježimo prenos. Dodajemo ovaj prenos sljedećoj znamenki 5: 1 + 5 = 6. Kad posudimo bazu 8 oduzimamo 8 - 6 = 2 i bilježimo 1 prenos. Dodajemo 1 iz prenosa i oduzimamo 1 - 1 = 0. Rezultat predstavljaju znamenke koje su rezultati oduzimanja, čitane odozdo prema gore: 267.

### *Primjeri množenja:*

$(325)_8 * (167)_8$	$6 * 5 = 30 : 8 = \mathbf{3}$ (prenos) $\mathbf{6}$ (ostatak se piše)
$(325)_8$	$6 * 2 = 12 + \mathbf{3} = 15 : 8 = \mathbf{1}$ (prenos) $\mathbf{7}$ (ostatak)
$(2376)_8$	$6 * 3 = 18 + \mathbf{1} = 19 : 8 = \mathbf{2}$ (prenos) $\mathbf{3}$ (ostatak)
	<hr/> <b>2376</b>
$(2723)_8$	$7 * 5 = 35 : 8 = \mathbf{4}$ (prenos) $\mathbf{3}$ (ostatak)
$(61403)_8$	$7 * 2 = 14 + \mathbf{4} = 18 : 8 = \mathbf{2}$ (prenos) $\mathbf{2}$ (ostatak)
	$7 * 3 = 21 + \mathbf{2} = 23 : 8 = \mathbf{2}$ (prenos) $\mathbf{7}$ (ostatak)
	<hr/> <b>2723</b>

Množenje u oktalnom brojnom sistemu obavlja se množenjem svake znamenke jednog broja sa svim znamenkama drugog broja. Rezultati množenja se potpisuju pomicanjem za jedno mjesto udesno. Pojasnit ćemo na primjeru. Množimo  $1 \times 325$  i rezultat  $325$  potpisujemo za zbrajanje. Sada množimo znamenku  $6$  sa svim znamenkama broja  $325$ . Dobijemo  $6 \cdot 5 = 30$ . Broj  $30$  dijelimo s bazom  $8$  i imamo  $6$  ostatak i  $3$  prenos. Množimo znamenku  $6 \cdot 2 = 12$  i dodamo  $3$  iz prenosa  $12 + 3 = 15$ . Dijelimo s bazom i dobijemo  $15 : 8 = 1$  prenos i  $7$  ostatak. Na kraju množimo  $6 \cdot 3 = 18$  i dodamo prenos  $1 = 18 + 1 = 19$ . Dijelimo  $19 : 8 = 2$  prenos i  $3$  ostatak. Više nema znamenki za množenje te čitamo rezultat uzimajući zadnji prenos i sve ostatke odozdo nagore:  $2376$ . Potpišemo pomicanjem za jedno mjesto udesno. Na isti način množimo znamenku  $7$  s brojem  $325$ . Rezultat  $2723$  potpišemo i sve zbrojimo.

### ***Primjeri dijeljenja:***

$$(61406)_8 : (32)_8 = (1717)_8$$

$$\begin{array}{r} \underline{-32} \\ 274 \\ \underline{-266} \\ 0060 \\ - \underline{32} \\ 266 \\ - \underline{266} \\ 000 \end{array}$$

Kao u dekadnom brojnog sistema. Neophodno je voditi računa o bazi brojnog sistema.

### **HEKSADEKADNI brojni sistem**

Kod heksadecimalnog brojnog sistema osnova sistema je **16**, te se pored poznatih oblika znamenki 0,1,2,3,4,5,6,7,8 i 9 za preostale znamenke sistema koriste slova A,B,C,D,E i F kako se za brojeve veće od 9 ne bi koristila dva znaka.

Dakle, znamenke heksadecimalnog sistema su od 0 do F po heksadekadnom označavanju, odnosno od 0 do 15 po dekadnom shvaćanju njihove vrijednosti. Brojevi heksadecimalnog sistema prikazani su u narednoj tablici:

<b>dekadno</b>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>heksadek.</b>	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
<b>dekadno</b>	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
<b>heksadek.</b>	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
<b>dekadno</b>	32	33	34	35	36	37	38	39	40	41	42	43				
<b>heksadek.</b>	20	21	22	23	24	25	26	27	28	29	2A	2B				

Tabela Generiranje dekadnih i heksadekadnih nizova.

Opći oblik za pretvaranje heksadecimalnog broja je:

$$N_{10} = a * 16^n + a * 16^{n-1} + \dots + a * 16^2 + a * 16^1 + a * 16^0$$

- **N** je broj brojnog sistema izražen znamenkama „a“,
- **a** znamenke sistema: *0,1,2,3,4,5,6,8,9,A,B,C,D,E* ili *F*,
- **16** je baza (osnova) brojnog sistema.

### Pretvaranje dekadnog broja u heksadecimalni broj

Pretvaranje dekadnog broja u heksadecimalni vrši se dijeljenjem s osnovicom 16.

$$(508)_{10} = ( )_{16}$$

508	:	16	=	31	Ostatak 12	<b>C</b>	↑ Čitanje
31	:	16	=	1	Ostatak 15	<b>F</b>	
1	:	16	=	0	Ostatak 1	<b>1</b>	

$$(508)_{10} = (1FC)_{16}$$

## Pretvaranje oktalnog zapisa broja u heksadecimalni zapis broja i obrnuto

Dijeljenjem binarnog broja u grupe po 4 znamenke u grupi, može se vrlo jednostavno izvršiti njegova pretvaranje u heksadecimalni.

$$1101001011111_2 = \overset{2}{\quad} \begin{array}{cccc} 0001 & 1010 & 0101 & 1111 \\ | & | & | & | \\ 1 & A & 5 & F \end{array} = 1A5F_{16}$$

Prvoj grupi predhode "0" da bi se popunila i bila vjerodostojnija u prikazu, a što je matematički ispravno. Pretvaranje heksadecimalnog broja u binarni vrši se obratnim postupkom:

$$\begin{aligned} 1A5F_{16} &= \begin{array}{cccc} 1 & A & 5 & F \\ | & | & | & | \\ 0001 & 1010 & 0101 & 1111 \end{array} = \\ &= 2^{12} + 2^{11} + 2^9 + 2^6 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = \\ &= 1 \times 16^3 + 10 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 = 6751_{10} \end{aligned}$$

Šesnaest bit-ni binarni broj može se upotrebom heksadecimalnog brojnog sistema vrlo prikladno prikazati. Pretvaranje je dosta jednostavna i omogućava brzo saznanje o očitanim sadržajima u memoriji računara ili nekom njegovom drugom sklopu. To je i razlog o potrebi poznavanja prikazanih brojnih sistema.

Pretvaranje iz oktalnog u heksadecimalni sistem i obratno je jednostavna, broj se pretvori u binarni i onda se grupira u grupe od četiri ili od tri znamenke i pretvara u drugi oblik.

$$6154_8 = 110\ 001\ 101\ 100_2 = 1100\ 0110\ 1100_2 = C6C_{16} = 3180_{10}$$

Direktna pretvaranje dekadnog broja u oktalni moguća je po istom načelu kao pretvaranje u binarni oblik. No najjednostavnije je dekadni broj pretvoriti u binarni a onda binarni broj grupiranjem binarnih znamenki pretvoriti u oktalni ili heksadekadni, već prema potrebi.

Pretvaranje oktalnog zapisa u heksadecimalni zapis nekog broja ne može se izvršiti naposredno, već se vrši preko binarnog broja. Jedna tetrada binarnog broja predstavlja heksadecimalni broj. Isto tako, tri znamenke binarnog broja predstavljaju jednu znamenku u oktalnom brojnom sistemu. Prema tome, oktalni broj se treba prevesti u binarni, a iz binarnog grupiranjem po četiri znamenke izračuna se heksadecimalna vrijednost.

***Primjeri zbrajanja:***

$$\begin{array}{r|l}
 (1F4C)_{16} & 3+C=3+12=15=\mathbf{F} \\
 + (2E83)_{16} & 8+4=12=\mathbf{C} \\
 \hline
 (4D83)_{16} & 14+15=29:16=\mathbf{I}(\text{prenos}) \quad 13 = \mathbf{D} (\text{ostatak}) \\
 & 1+2+\mathbf{I}=\mathbf{4}
 \end{array}$$

***Primjeri oduzimanja:***

$$\begin{array}{r|l}
 (21A3)_{16} & 3+16=19-15=\mathbf{4} ((\mathbf{I}) \text{ prenos}) \\
 - (1FFF)_{16} & 10+16=26-(15+\mathbf{I}) = 10 (\mathbf{A}) ((\mathbf{I}) \text{ prenos}) \\
 \hline
 ( \ 1A4)_{16} & 1+16=17-(15+\mathbf{I})=\mathbf{1}
 \end{array}$$

***Primjer množenja:***

$$\begin{array}{r}
 \underline{(A9E4F)_{16} \times (8A7)_{16}} \\
 (54F278)_{16} \\
 (6A2F16)_{16} \\
 \hline
 4A5429)_{16} \\
 (5BDFBD89)_{16}
 \end{array}$$

---

množi se sukcesivno sa brojevima:

Primjena binarnog brojnog sistema u računarskoj tehnici opravdana je zbog dvije prednosti koje sistem omogućava:

- **Pouzdanost u radu,**
- **Ekonomičnost.**

Pouzdanost se lako i sigurno ostvaruje jer elektronički sklop treba da zauzme samo dva stanja: ima i nema napona, odnosno "1" ili "0", uprošteno "radi" ili "ne radi". Ako se uz to svakoj binarnoj kombinaciji pridoda određeni broj bit-a na način da se za svaku osigura ukupan paran broj jedinica ili nula (provjera na parnost), ili se izračunava ukupni brojevi iznos kao zbir svih brojki unutar određenog "bloka" podataka - kontrolni zbir (check sum) koji se uz pripadni mu blok prenosi, te slično navedenom, pridonosi se ukupnoj pouzdanosti sistema i kontroli na pojavu greške. Ekonomičnost se očituje u potrebi za najmanjim brojem vodova za prenos signala na daljinu.

No zbog preglednosti i upravljanja radom sistema reprezentiranje korisniku je u drugim brojnim sistemima.

Osnovni i najmanji element u kombinaciji impulsa je mogućnost da na jedno mjesto u kombinaciji impulsa dođe pozitivni ili negativni impuls, odnosno "0" ili "1". Izbor između "0" i "1" predstavlja najmanji mogući izbor i predstavlja mjeru količine informacija nazvanu **BIT (BInary digiT=binarna znamenka)**. Prema ASCII kodu skup od 8b (osam bit-a) predstavlja jedan znak. Prema dogovoru ta je kombinacija nazvana **BAJT (BYTE)**.

Dakle:

$$8 \text{ b (bit-a)} = 1 \text{ B (Bajt)}$$

Veće jedinice za mjerenje količine informacije od navedenih su:

**1 kB (kilo Bajt) = 1 024 B**

**1 MB (Mega Bajt) = 1 024 kB = 1 048 576 B**

**1 GB (Giga Bajt) = 1 024 MB = 1 048 576 kB = 1 073 741 824 B**

Ako kažemo da neki memorijski medij ima KAPACITET od 4 MB, pojednostavljeno rečeno to znači da je u njega moguće spremiti 4'194'304 B, odnosno znakova, u veličini od 8b (osam bit-a) svaki.

Multiplikator 1024 rezultat je matematičkog izraza:

$$2^{10} = 1024$$

a to je dekadni iznos binarnog broja: 100'0000'0000B.

Kako računar (digitalno) koristi isključivo brojeve, to znači da se svi znakovi, instrukcije i podaci moraju pretvoriti u brojeve kako bi računaru bili razumljivi. Znakovi i instrukcije najčešće se unose preko tastature koja šalje računaru odgovarajuće kombinacije impulsa. Uopćeno, tastatura je elektromehanički pretvarač koji znak na tipki po pritisku pretvara u pripadnu mu binarnu kombinaciju. Svakom znaku pripada njemu svojstvena binarna kombinacija. Skup znakova i binarnih kombinacija naziva se KOD, a sam postupak kreiranja binarnih kombinacija naziva se KODIRANJE.

Iz skorije čovjekove istorijui postupak pridruživanja impulsa, električnih ili svijetlosnih ili sličnih, pojedinim znakovima pisma poznat je pod nazivom "MORZE-ova abeceda". Računar koristi ista načela u naprednijem obliku.

Iz navedenog je jasno da računari ne mogu uspješno razmjenjivati podatke ako ne koriste isti kod, te je od velikog značaja standardizacija koda i njegovo poštivanje, ali i njegovo poznavanje.

---

## KOD I KODIRANJE

Pretpostavka uspješnog komuniciranja računara u razmjeni ili obradi podataka je DOGOVOR o skupu znakova koji će se u radu s računarm koristiti te pripadnim binarnim kombinacijama za svaki od znakova i zadaći koja se za svakog ima izvršiti. Dogovor se vremenom proširuje i dograđuje, a on je ili ozakonjeni standard države ili preporuka neke međunarodne organizacije.

Skup svih znakova koji se tako koriste naziva se APSTRAKTNA ABECEDA, koja zajedno s pripadnim binarnim (ili nekim drugim) kombinacijama tvori KOD. Pojedini znakovi u kodu nazivaju se KODNI ELEMENTI, a pripadni im zamjenjitelj, bilo da je u pitanju binarna kombinacija, neki drugi znak ili nešto treće, nazivaju se KODNA ZAMJENA.

Broj kodnih elemenata u apstraktnoj abecedi naziva se OBIM KODA. Kod računara broj upotrebljenih elemenata proporcionalan je broju bit-a koji se žele upotrijebiti za kodnu zamjenu prema slijedećem izrazu:

$N = 2^m$ , gdje N obim koda; broj elementa, a m je broj bit\_a upotrebljen za kodnu zamjenu

### *Primjer*

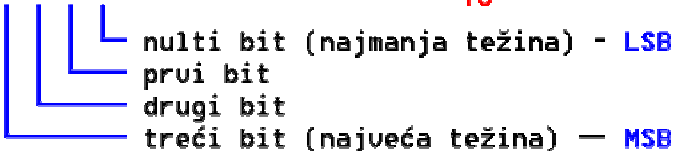
Koliko se znakova može kodirati ako se koristi kodna zamjena od:

- A.) 2 bit-a,
- B.) 4 bit-a,
- C.) 8 bit-a,

A.) S dva bit-a moguće je ostvariti slijedeće kodne zamjene:

- |    |     |     |   |                           |
|----|-----|-----|---|---------------------------|
| 1. | --- | 0 0 | → | prva kodna zamjena        |
| 2. | --- | 0 1 |   |                           |
| 3. | --- | 1 0 |   | $N = 2^2 = 4$ kombinacije |
| 4. | --- | 1 1 | → | četvrta kodna zamjena     |

B.) Proširenjem na četiri bit-a po kodnoj zamjeni daje veći broj mogućih upotrebljenih kodnih elemenata:

- |     |     |         |       |                            |
|-----|-----|---------|-------|----------------------------|
| 1.  | --- | 0 0 0 0 | ..... | ( 0 <sub>10</sub> )        |
| 2.  | --- | 0 0 0 1 |       |                            |
| 3.  | --- | 0 0 1 0 |       |                            |
| 4.  | --- | 0 0 1 1 |       |                            |
| 5.  | --- | 0 1 0 0 |       |                            |
| 6.  | --- | 0 1 0 1 |       |                            |
| 7.  | --- | 0 1 1 0 |       | $N = 2^4 = 16$ kombinacija |
| 8.  | --- | 0 1 1 1 |       |                            |
| 9.  | --- | 1 0 0 0 |       |                            |
| 10. | --- | 1 0 0 1 |       |                            |
| 11. | --- | 1 0 1 0 |       |                            |
| 12. | --- | 1 0 1 1 |       |                            |
| 13. | --- | 1 1 0 0 |       |                            |
| 14. | --- | 1 1 0 1 |       |                            |
| 15. | --- | 1 1 1 0 |       |                            |
| 16. | --- | 1 1 1 1 | ..... | ( 15 <sub>10</sub> )       |
- 

Bit najveće težine (prvi s lijeve strane) često se naziva VODEĆI BIT binarne kombinacije. Najmanje značajan bit (Least Significant Bit) obično ima oznaku **LSB**, a vodeći bit ima najveći značaj i oznaku **MSB** (Most Significant Bit).

C.) Mikroračunari najčešće koriste kod od osam bit-nih kodnih zamjena te je:

1.	---	00000000	( 0 <sub>10</sub> )
2.	---	00000001	( 1 <sub>10</sub> )
3.	---	00000010	
:			
:			
127.	---	01111110	
128.	---	01111111	( 127 <sub>10</sub> )
129.	---	10000000	( 128 <sub>10</sub> )
130.	---	10000001	
:			
:			
254.	---	11111101	
255.	---	11111110	
256.	---	11111111	( 255 <sub>10</sub> )

$$N = 2^8 = 256 \text{ kombinacija}$$

Koji će se znak pridružiti određenoj kodnoj kombinaciji (kodnoj zamjeni) ovisiti će o prihvaćenom dogovoru korisnika.

Postupak pretvaranja kodnog elementa u kodnu zamjenu naziva se KODIRANJE, a postupak pretvaranja kodne zamjene u odgovarajući znak naziva se DEKODIRANJE. Kod računara postupak kodiranja u pravilu se odvija preko tastature, a postupak dekodiranja očituje se kao prikaz na ekranu monitora računara ili na štampaču.

Morze-ov kod nije imao kombinacije impulsa raznolike samo po rasporedu već i po broju impulsa i njihovoj duljini trajanja. Takav neujednačeni kod nije prikladan u računarskoj tehnici zbog otežane sinhronizacije i prepoznavanja. Prikladnije je kad svaka kodna zamjena jednako vremenski traje.

U računarskoj tehnici najraširenija je upotreba BCD i ASCII koda.



					pomoćni bit za drugi skup									
					0	0	0	0	1	1	1	1		
					0	0	1	1	0	0	1	1		
					0	1	0	1	0	1	0	1		
redak	b3	b2	b1	b0	r\s	0	1	2	3	4	5	6	7	
	s t u p a c													
		0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
		0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
		0	0	1	0	2	STX	DC2	"	2	B	R	q	r
		0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
		0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
		0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
		0	1	1	0	6	ACK	SYN	&	6	F	U	f	v
		0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
		1	0	0	0	8	BS	CAN	(	8	H	X	h	x
		1	0	0	1	9	HT	EM	)	9	I	T	i	t
		1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
	1	0	1	1	B	UT	ESC	+	;	K	[	k	{	
	1	1	0	0	C	FF	FS	.	<	L	\	l		
	1	1	0	1	D	CR	GS	-	=	M	]	m	}	
	1	1	1	0	E	SO	RS	.	>	N	^	n	~	
	1	1	1	1	F	SI	US	/	?	O	_	o	DEL	

Tabela Prvih 128 elemenata ASCII koda.

Kod je zamišljen da omogući što veću fleksibilnost i brzinu. Tako na primjer sve instrukcije imaju bit-ove "b6" i "b5" jednake nuli, a bit "b7" je nula ako su u pitanju standardizirani znakovi, odnosno jedan ako su u pitanju znakovi definisani od korisnika, koji čine drugu grupu od 128 znakova ASCII koda. Znakovi velikih i malih slova poklapaju se u prva četiri bit-a (b0-b3) i prepoznaju se analizom "b5", a za sva slova "b6" je jedan.

Iz tablice lako se pronade binarna kodna zamjena znaka čitajući bit-ove "b7-b0" i heksadecimalna vrijednost znaka čitajući stupac-redak.

### *Primjer*

III

Izračun vrijednosti znakova ASCII koda izražen različitim brojnim sistemima.

$$\text{"STX"} = 0000\ 0010_2 = 03_{16} = 3_{10}$$

$$\text{" W "} = 0101\ 0111_2 = 57_{16} = 87_{10}$$

$$\text{" k "} = 0110\ 1011_2 = 6B_{16} = 107_{10}$$

Pretvaranje iz heksadecimalnog u dekadni oblik nije preteška jer su u pitanju samo dvije heksadecimalne znamenke.

Dakle, svaki znak ili instrukcija opisani su s kombinacijom od po 4 bit-a višeg i nižeg reda i grupirani su tako da se brzo prepoznaje o kojem je znaku riječ. Bit-ovi od "b0-b6" nazivaju se informativni bit-ovi, a bit "b7" pomoćni bit.

Značenje pojedinih instrukcija ASCII koda je:

- **NUL** -- logički uređaj bez utjecaja ili odziva na poruku
- **SOH** -- oznaka početka kontrolnih podataka pred blokom poruke
- **STX** -- oznaka kraja kontrolnih podataka i početak bloka poruke
- **ETX** -- oznaka kraja bloka poruke
- **EOT** -- oznaka kraja prenosa podatka (kontrolne i poruke)
- **ENQ** -- zahtijev za odgovor odredišta
- **ACK** -- potvrda odredišta o prijemu
- **BEL** -- aktiviranje zvučnog poziva
- **BS -** -- brisanje kursorom unatrag (u lijevo) u reduku
- **HT** -- kontrola tabulacije (skokova) u reduku
- **LF** -- postavljanje kursora u novi red
- **VT** -- kontrola tabulacije (skokova) po redovima
- **FF** -- postavljanje papira štampača na početak nove stranice
- **CR** -- vraćanje na polazni položaj u istom reduku
- **SO** -- početak bloka znakova drugačijeg značenja

- **SI** -- kraj bloka znakova drugačijeg značenja
- **DLE** -- promjena značenja kontrolnih znakova
- **DC1 od DC4** -- korisnički definisani kontrolni znaci
- **NAK** -- odgovor prijemnika o neostvarenom prijemu
- **SYN** -- održavanje sinhronizacije (usklađenosti) prenosa
- **ETB** -- oznaka kraja bloka podataka
- **CAN** -- poruka o pogrešno otposlanim podacima
- **EM** -- fizički kraj pošiljanja podataka
- **SUB** -- slijedi zamjena za neispravne podatke
- **ESC** -- daje mogućnost uporabe proširenja koda
- **FS** -- rastavljač blokova podataka (datoteka)
- **GS** -- rastavljač grupe podataka u bloku
- **RS** -- rastavljač pojedinih zapisa u grupi
- **US** -- rastavljač jedinica
- **SP** -- razmak (prazan znak)
- **DEL** -- brisanje znaka s desne strane kursora

1968. godine međunarodna organizacija za telekomunikacije i promet (CCITT) ustanovila je MEĐUNARODNI KOD br.5, koji se u suštini ne razlikuje od ASCII koda.

Poširenje ASCII koda (drugih 128 znakova) sadrži uglavnom znakove grčkog pisma, znakove lokaliteta i znakove za kreiranje okvira.

## RAČUNARSKA ARITMETIKA

Bez obzira u kojem će se brojnem sistemu prikazivati i unositi podaci u digitalno računar, sve radnje u njemu odvijaju se u binarnom brojnem sistemu iz već navedenih razloga. U suštini nema razlike u postupku izvođenja aritmetičkih operacija u binarnom i dekadnom brojnem sistemu (ili nekom drugom), jer sva pravila koja vrijede kod osnovnih aritmetičkih operacija u dekadnom brojnem sistemu vrijede i za binarni brojni sistem.

---

## RAČUNARSKA LOGIKA

Kako je već u predhodno rečeno, rad digitalnog računara temelji se na DVA definisana fizikalna stanja:

1. ima impulsa (napona) — simbolička oznaka "1"
2. nema impulsa (napona) — simbolička oznaka "0"

Znači da se elektronički sklopovi, koji u računaru obavljaju razne operacije, ponašaju slično prekidačkim elementima koji različitim električnim izvedbama izvršavaju operacije sa stanjima "1" i "0" po zakonima LOGIČKIH PRIJEDLOGA koji mogu biti ISTINITI ili NEISTINITI.

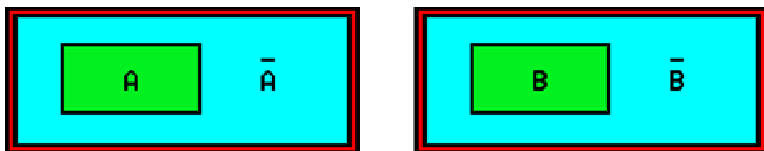
Osnovu ove grane logike, koja datira još od Aristotela, praktički je obradio i definisao njene simbole matematičar George Boole. Po njemu je ova grana matematike (zakoni istinitosti) nazvana BOOLE-ova ALGEBRA, a bavi se međusobnim odnosima elemenata u skupu i između skupova.

### Logika skupova

SKUP je grupa elemenata koji posjeduju barem jednu zajedničku karakteristiku u jednoj jedinstvenoj grupi elemenata (opći skup). To znači da se grupa elemenata može podijeliti u dvije podgrupe-PODSKUPA i to:

1. Podskup elemenata s zajedničkim svojstvima - npr. "A"
2. Podskup elemenata koji ne pripada pod 1. -- npr. "non A"

Tu podijelu u skupu jasnije predočava slijedeća slika.



**Slika Opći skupovi elemenata.**

Podskupovi "A" i "non A" su komplementarni, "A" se predstavlja kao ISTINIT i pridružuje mu se stanje "1", a njemu komplementaran skup "non A" kao NEISTINIT i pridružuje mu se stanje "0". Oba stanja kao rezultat obrade ne mogu istovremeno postojati. Isto vrijedi za podskupove "B" i "non B".

Operacije koje povezuju navedena stanja u skupu i između pojedinih skupova su:

1. **Operacija NO ( NE )** --- negacija postojećeg stanja
2. **Operacija OR ( ILI )** --- zahtijeva barem jedno istinitostanje za rezultat "1" pri obradi
3. **Operacija AND ( I )** --- zahtijeva sva istinita stanja za rezultat "1" pri obradi
4. **Operacija NOR ( NILI )** --- negacija OR
5. **Operacija NAND ( NI )** --- negacija AND
6. **Operacija EXOR ( isključivi ILI )** --- zahtijeva samo jedno istinito stanje za rezultat "1" pri obradi

Prve tri operacije su osnovne, a preostale su iz njih izvedene. Rezultati odnosa između skupova i u skupu prikazuju se TABLICAMA ISTINE kako slijedi.

1. Tablica istine operacije NE:

	stanje skupa A pred operaciju		stanje skupa A po operaciji		A	NE
stanje I	1	(istina)	0	(neistina)	1	0
stanje II	0	(neistina)	1	(istina)	0	1

U lijevi stupac tablice upisuju se sva moguća stanja koje skup "A" može imati, kao i kombinacije svih stanja skupova ako ima više skupova. U desnom stupcu ili stupcima upisuje se stanje koje nastaje kao rezultat izvršene operacije ili operacija nad stanjima prikazanim u lijevom stupcu.

2. Tablica istine operacija ILI, NILI i isključivi ILI izvedenim nad različitim kombinacijama elementa dva skupa.

stanje skupova pred operaciju		rezultat operacije nad elementima skupova		
A	B	ILI	NILI	EXILI
0	0	0	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	0	0

3. Tablica istine operacija I i NI izvedenim nad različitim kombinacijama elementa dva skupa.

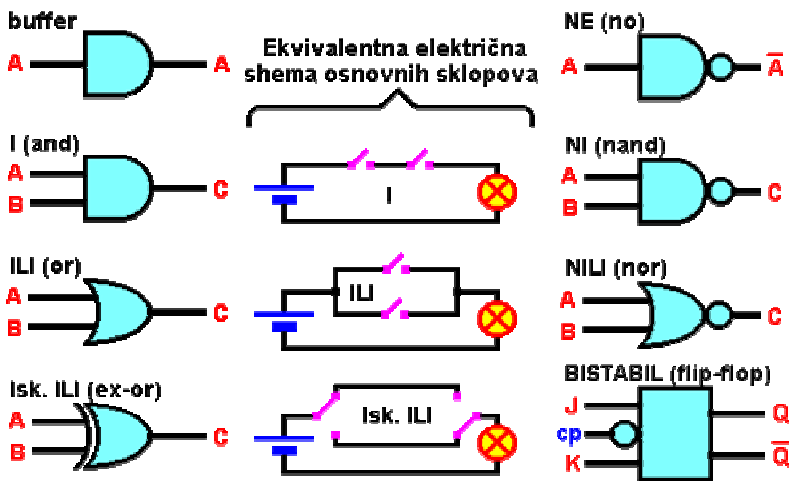
A	B	I	NI
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Iz tablice proizlazi da neke operacije između skupova odgovaraju operacijama s binarnim brojevima. Ako se "1" i "0" ne koristi za oznaku istine i neistine, već za postojanje i nepostojanje impulsa, posebno konstruisani logički električni sklopovi obavljat će logičke operacije s impulsima i omogućiti obavljanje aritmetičkih operacija. Zbog toga se takvi električni sklopovi, u pravilu izvedeni u integrisanoj tehnici, nazivaju INTEGRISANI LOGIČKI ELEKTRIČNI SKLOPOVI.

### Osnovni logički sklopovi računara

Razvoj digitalnih računara usko je povezan s razvojem tehnologije izrade integrisanih sklopova, koja omogućava nebrojne mogućnosti kreacije spojeva logičkih sklopova u jednom kućištu vrlo malog volumena.

Na ulaze logičkih integrisanih sklopova dovode se impulsi nad kojima se želi izvršiti operacija, a rezultat operacije se očituje na izlazu odmah po promjeni ulaznog stanja ili po dopuštenju kojim upravlja nekakav generator takta.



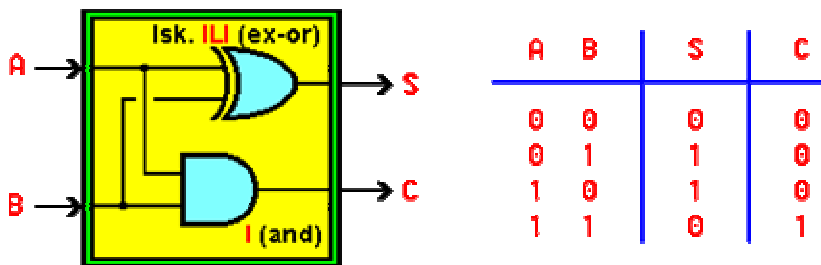
Slika Osnovni logički sklopovi digitalnog računara.

Povezivanjem logičkih sklopova u jedinstvenu funkcionalnu cjelinu ostvaruju se pojedine zadaće kao aritmetičke operacije i druge.

Prikazani simboli na slici uvršteni su u anglosaksonskoj stručnoj literaturi, iako se mogu susresti i drugačije ali slične oznake. No za ilustraciju označavanja i crtanja logičkih sklopova dovoljni su i primjeri sa slike.

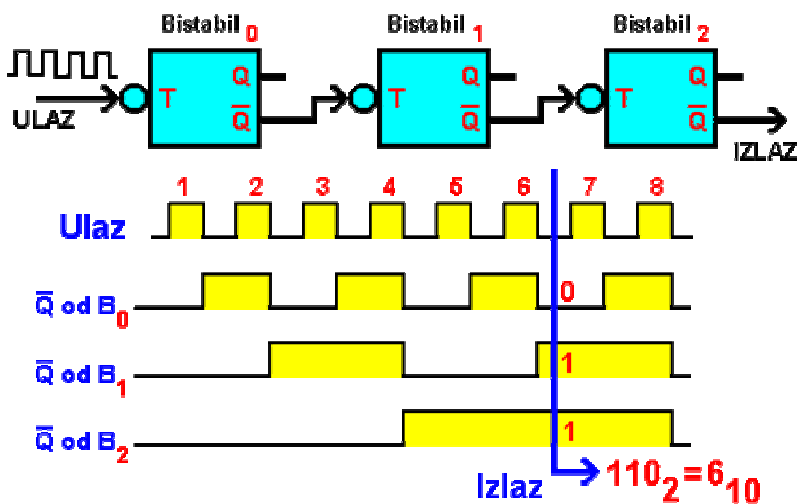
**Primjer**

Realizacija zbrajanja binarnih brojeva u računaru:



Sklop na predhodnoj slici obavlja samo operaciju zbrajanja bez prenosa. Sklop će obaviti zadaću za nulti bit binarnog broja (prva znamenka). Da bi se prenos obuhvatio pri zbrajanju uz naredne bit-ove binarnog broja (daljnje znamenke) to znači da se izlaz 'C' mora pridružiti ulazima 'A' i 'B' za svaku narednu znamenku, odnosno jedan polusumator vršio bi binarno zbrajanje znamenki, a drugi bi polusumator izračunatom zbroju pridodao predhodni prenos i tako za svaku narednu znamenku.

Jedna od osnovnih kombinacija logičkih sklopova u digitalnoj tehnici je impulsno brojilo. Zadaća mu je pored binarnog brojanja i dijeljenje frekvencije. Osnovni element brojila je bistabil. Najjednostavniji primjer u izvedbi s T-bistabilom prikazan je na narednoj slici.



Slika Asinhrono binarno brojilo.

T-bistabil ima osobinu, prema izvedbi na slici, da mijenja stanje na izlazu pri svakom prelasku ulaznih impulsa iz stanja '1' u stanje '0'. Ako se izlaz iz jednog bistabila upotrebi kao ulaz u drugi osim

---

efekta dijeljenja frekvencije, što se vidi prema duljini impulsa na svakom od izlaza, binarne kombinacije na izlazima bistabila daju upravo binarne vrijednosti broja impulsa koji su pridošli na ulaz u brojilo.

Brojilo s tri bistabila brojati će do  $2^3=8$ . S većim brojem bistabila i međusobnim povratnim spregama s izlaza jednog prema ulazu nekog od predhodnih može se postići brojanje do deset.

Brojila imaju osobito veliki značaj pri dizajnu električnih komponenti za mjerne svrhe. I samo digitalno računaru u svojoj konstrukciji sadrži nekoliko brojila od kojih rad jednog od njih prepoznamo kroz prikaz datuma i vremena na monitoru računara.

---

# ALGORITMI I STRUKTURE PODATAKA

Def.: Algoritam je metod (postupak) za rješavanje problema koji je prihvatljiv za kompjutersku implementaciju.

Def.: Algoritam je metod (postupak) koji:

1. ukoliko problem ima rješenje, daje rješenje tog problema,
2. ukoliko problem nema rješenje, daje odgovor da problem nema rješenje.

Ovo je generalna definicija algoritma.

Program je implemntacija algoritma.

Riječ algoritam potiče iz IX vijeka, od bagdadskog astronoma i matematičara Muhameda ibni Musa El-Havarizmije koji je na zapadu poznat kao Al-Gorezmije.

## Algoritmi

Pojam algoritma predstavlja temelj za razumijevanje računarstva općenito i stoga mu je u ovom dijelu posvećena posebna pažnja.

## Neformalni algoritmi

Obzirom na naučeno gradivo trebali bi znati definisati algoritam koji, primjerice, pretvara različite formate numeričkih podataka, upravlja raspodjelom procesorskog vremena u višezadaćnom okruženju i sl. Algoritmom možemo opisati i izvođenje samog algoritma u stroju kao:

Sve dok se ne pojavi instrukcija za zaustavljanje, ponavljaj:

Dohvaćaj instrukciju

Dekodiraj instrukciju

Izvedi instrukciju

---

Algoritmi nisu ograničeni samo na tehničke aktivnosti. Pomoću algoritma možemo opisati mađioničarski trik ili neku svakodnevnu aktivnost, primjerice čišćenje graška kao:

Uoči košaru sa neočišćenim graškom i praznu zdjelu

Sve dok ima graška u košari, radi:

Uzmi mahunu iz košare

Prelomi je da se na njoj pojavi otvor

Istresi grašak u zdjelu

Baci mahunu

Mnogi istraživači vjeruju da je bilo koja aktivnost ljudskog uma, uključujući zamišljanje, kreativnost i donošenje odluka, u osnovi rezultat izvođenja nekog od algoritama što je za posljedicu imalo razvoj znanstvene discipline poznate pod imenom Umjetna inteligencija.

*Primjer 1.*

Napiši algoritam i dijagram toka za računanje razlike, proizvoda i količnika dva zadana broja.

**Algoritam:**

*Početak*

-----  
*upiši A, B*

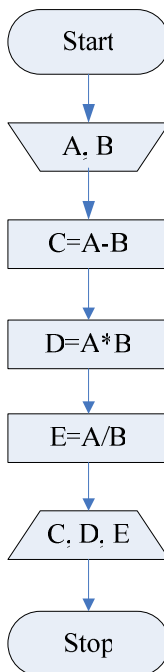
$C = A - B$

$D = A \times B$

$E = A / B$

*ispiši C, D, E*

-----  
*kraj*



*Primjer 2.*

Napiši algoritam i dijagram toka za računanje prosjeka 4 zadana broja.

**Algoritam:**

*Početak*

---

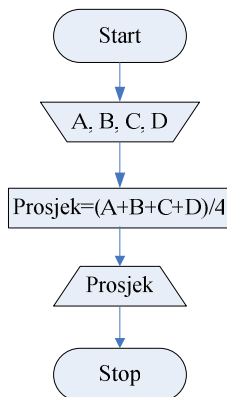
*upiši A,B,C,D*

$$PROSJEK = (A + B + C + D) / 4$$

*ispiši PROSJEK*

---

*kraj*



*Primjer 3.*

Napiši algoritam i dijagram toka za pretvaranje centimetara u metre. Zadaje se mjerni broj u centimetrima, a treba odrediti koliko je to metara.

**Algoritam:**

*Početak*

---

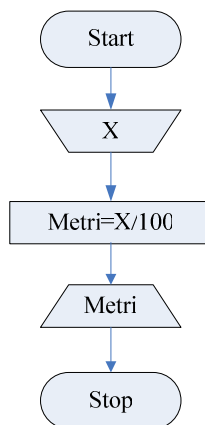
*upiši X*

$$METRI = X / 100$$

*ispiši METRI*

---

*kraj*



#### Primjer 4

Napiši algoritam i dijagram toka koji zadani broj  $A$  uvećava 100 puta ako je  $A$  manji od 100, a inače  $A$  umanjuje za 100. Ispisati dobiveni rezultat.

#### Algoritam:

Početak

upiši  $A$

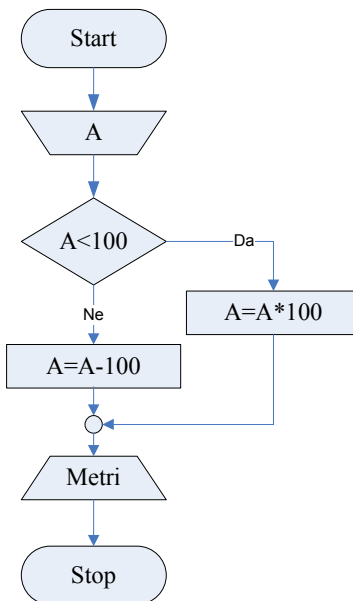
ako je  $A < 100$

onda  $A = A \times 100$

inače  $A = A - 100$

ispiši  $A$

kraj



### Primjer 5

Napiši algoritam i dijagram toka za računanje opsega i površine kvadrata, ako je zadana dužina stranice kvadrata  $A$ . Prije računanja provjeriti je li  $A$  veće od 0, a ako nije ispisati poruku: "Stranica kvadrata mora biti pozitivan broj."

#### **Algoritam:**

*Početak*

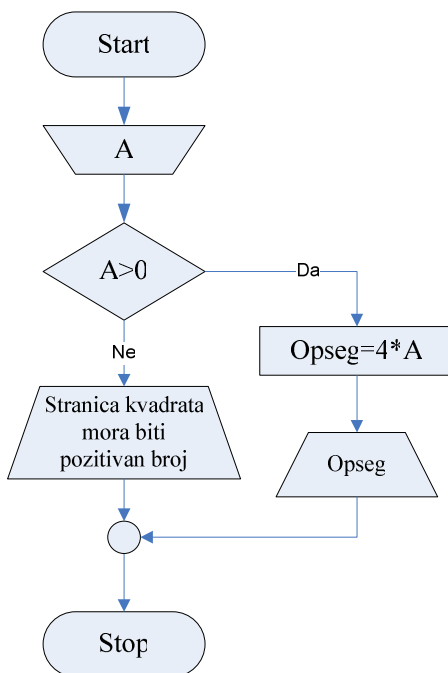
*upiši  $A$*

*ako je  $A > 0$*

*onda opseg =  $4 \times A$ , ispiši opseg*

*inače ispiši "Stranica kvadrata mora biti pozitivan broj."*

*kraj*



### Primjer 6

Napiši algoritam i dijagram toka za računanje apsolutne vrijednosti cijelog broja. Uputa: Apsolutna vrijednost je udaljenost cijelog broja od nule. ( $|-3|=3$ ,  $|3|=3$ ,  $|0|=0$ ).

### Algoritam:

Početak

upiši A

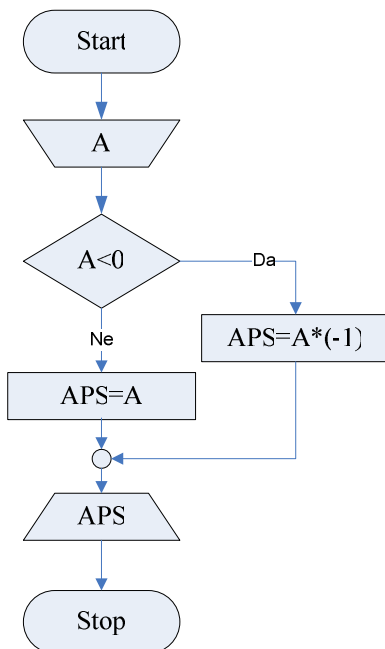
ako je  $A < 0$

onda  $APS = (-1) * A$

inače  $APS = A$

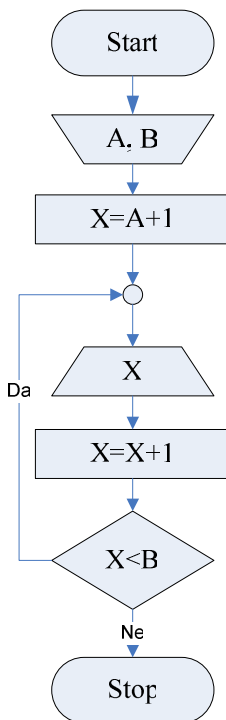
ispiši APS

kraj



### Primjer 7

Napiši algoritam i dijagram toka za ispisivanje svih prirodnih brojeva između zadanih brojeva A i B. Pretpostavite da su A i B zadani tako da je  $A+1 < B$  (treba ispisati barem jedan broj).



### Primjer 8

Napiši algoritam i dijagram toka za zbrajanje prvih  $N$  prirodnih brojeva.

#### Algoritam:

*Početak*

$X=0$

$ZBROJ = 0$

*upiši  $N$*

$X = X + 1$

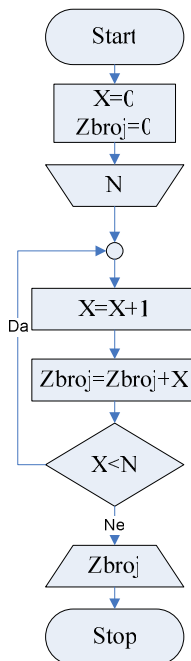
$ZBROJ = ZBROJ + X$

*ako je  $X < N$*

*onda vrati se na naredbu  $X = X + 1$*

*ispiši  $ZBROJ$*

*kraj*



---

### *Primjer 9*

Napiši algoritam i dijagram toka koji učitava  $N$  cijelih brojeva. Izračunati zbroj svih pozitivnih i zbroj svih negativnih brojeva.

#### **Algoritam:**

*Početak*

*upiši  $N$*

*broj = 0*

*pozitivni = 0*

*negativni = 0*

*upiši  $x$*

*ako je  $x > 0$*

*onda   pozitivni = pozitivni +  $x$*

*inače   negativni = negativni +  $x$*

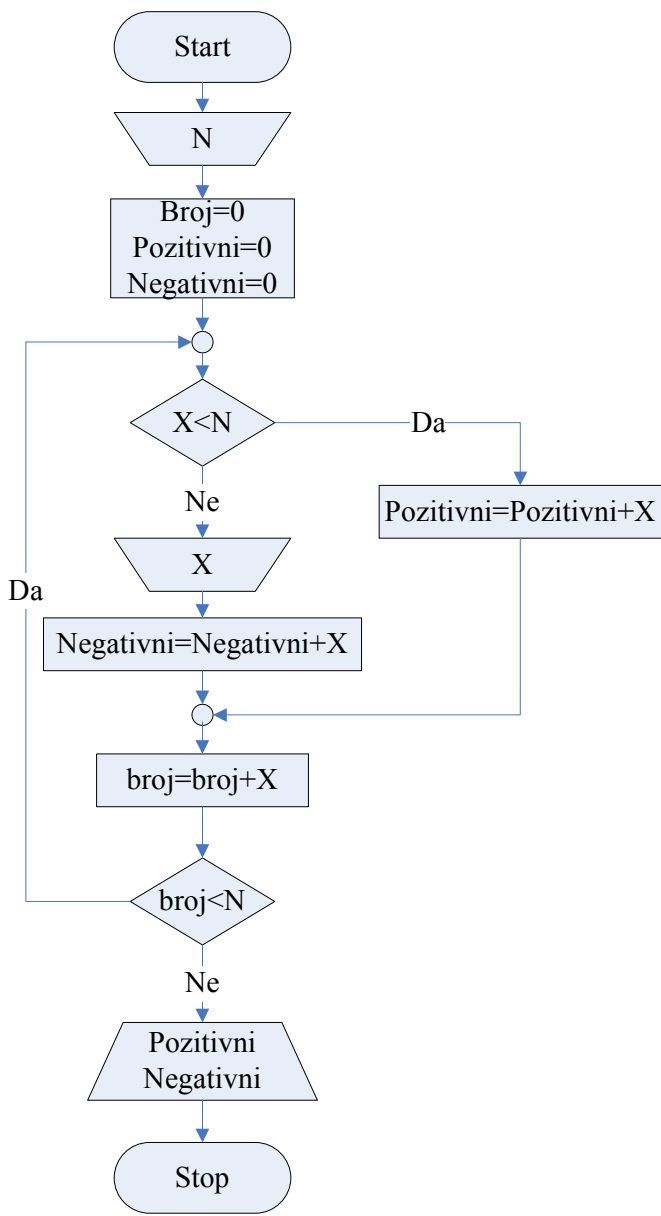
*broj = broj + 1*

*ako je broj <  $N$*

*onda vrati se na naredbu: upiši  $x$*

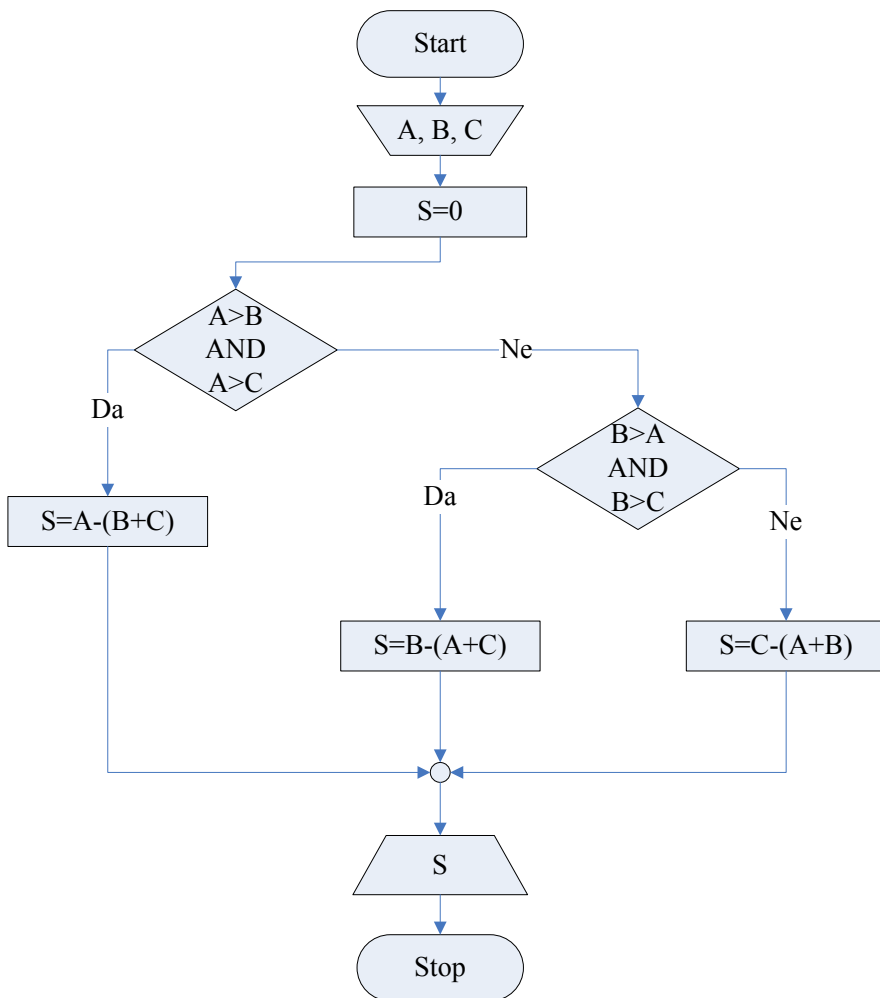
*ispiši pozitivni, negativni*

*kraj*



*Primjer 10*

Nacrtati dijagram toka za program koji će učitati tri broja i ispisati razliku između najvećeg i zbroja preostala dva.



---

## Primjeri za vježbu:

1. Napiši algoritam i dijagram toka koji čita tri broja i ispisuje najveći od njih.
2. Napiši algoritam i dijagram toka za ispis prestupnih godina 20. stoljeća.
3. Napiši algoritam i dijagram toka koji čita tri broja i ispisuje najmanji od njih.
4. Napiši algoritam i dijagram toka za ispis godina koje nisu prestupne u 20. stoljeća.
5. Napiši algoritam i dijagram toka: Učitati  $a$ . Ako je  $a$  djeljivo sa  $2$   $b$  postaje  $2$ , inace  $b$  je  $1$ .
6. Napiši algoritam i dijagram toka: Učitati  $x$ , Ako je  $x$  pozitivno  $z$  postaje suma  $a$  i  $b$ , inace  $z$  je razlika  $a$  i  $b$ .
7. Napiši algoritam i dijagram toka: Ispisati prirodne brojeve od  $10$  do  $6$  unazad.
8. Napiši algoritam i dijagram toka: Ispisati dvostruku vrijednosti prirodnih brojeva od  $1$  do  $5$ .
9. Napiši algoritam i dijagram toka: Ispisati dvostruku vrijednosti prirodnih brojeva od  $6$  do  $14$ .
10. Napiši algoritam i dijagram toka: Ispisati dvostruku vrijednosti prirodnih brojeva od  $8$  do  $16$ .
11. Napiši algoritam i dijagram toka: Ispisati trostruku vrijednosti prirodnih brojeva od  $15$  do  $27$ .
12. Napiši algoritam i dijagram toka: Ispisati trostruku vrijednosti prirodnih brojeva od  $k$  do  $n$ .

**Napomena:** Godina s 366 dana naziva se prestupna godina. Godina je prestupna ako je djeljiva sa  $4$ .

---

## **Literatura:**

*predavanja* prof. dr Diana Protić

*predavanja* prof. dr Novica Nosović

*predavanja* prof. dr Faruk Turčinhodžić

---

## Sadržaj

UVOD U INFORMATIKU I RAČUNARSTVO .....	3
Pojam informatike kao nauke .....	3
Pojam informacije, podatka i informatike .....	4
Računar i program .....	5
RAČUNARSKI SISTEM .....	7
BROJNI SISTEMI .....	8
DEKADNI brojni sistem .....	10
BINARNI brojni sistem .....	11
Pretvaranje dekadnog broja u binarni broj .....	12
Pretvaranje binarnog broja u dekadni broj .....	15
Aritmetika u binarnom brojnom sistemu .....	16
OKTALNI brojni sistem .....	18
Pretvaranje dekadnog broja u oktalni broj .....	19
Aritmetika u oktalnom brojnom sistemu .....	20
HEKSADEKADNI brojni sistem .....	22
Pretvaranje dekadnog broja u heksadecimalni broj .....	23
Pretvaranje oktalnog zapisa broja u heksadecimalni zapis broja i obrnuto ...	24
KOD I KODIRANJE .....	28
BCD kod .....	31
ASCII kod .....	31
RAČUNARSKA ARITMETIKA .....	34
RAČUNARSKA LOGIKA .....	35
Logika skupova .....	35
Osnovni logički sklopovi računara .....	38
ALGORITMI I STRUKTURE PODATAKA .....	42
Algoritmi .....	42
Primjeri za vježbu: .....	54