

Programiranje u fizici

3. Osnove sintakse programskog jezika C

Prirodno-matematički fakultet u Nišu
Departman za fiziku

Tokeni u C-u

Program napisan u programskog jeziku C sastoji se od različitih tokena. Tokeni mogu biti:

- ključnih reči
- identifikatora
- konstanti
- niza znakova
- simbola

1. Arhitektura računara
2. Programski jezici
3. Osnove programskog jezika C

Komentar, Identifikatori, Ključne reči, Blanko znak

Tipovi podataka

Celobrojni tipovi, Tipovi sa pokretnim zarezom, void tip

Promenljive,

Definicija, inicijalizacija promenljivih

Primer:

Sledeća naredba

```
printf ("Zdravo ! \n");
```

sadrži pet tokena:

```
printf  
(  
"Zdravo ! \n"  
)  
;
```

Znak ;

U C-u znak ; označava kraj naredbe. Znači svaka pojedinačna naredba programa mora biti završena znakom ; .

Primer:

```
printf ("Zdravo ! \n");  
return 0;
```

Komentari

Komentari predstavljaju pomoćni tekst koji će biti zanemaren od strane prevodioca. Počinje znakom `/*` a završava se znakom `*/` kao što je to ilustrovano sledećim primerom:

Primer:

```
/* Ovo je primer komentara */
```

```
// ovo je komentar do kraj linije
```

Ograničenje:

Nije moguće imati komentar u komentaru niti se komentar može naći u znakovnom nizu.

Identifikatori

Identifikator u C-u su imena koja se koriste za označavanje promenljivih, funkcija ili bilo kog drugog entiteta definisanog od strane programera.

Identifikator mora početi slovom **A** do **Z**, **a** do **z** ili znakom **_** nakon čega sledi ni jedan ili proizvoljan broj slova, znakova **_** i brojeva od **0** do **9**.

Nije dozvoljeno korišćenje znakova interpukcije za označavanje identifikatora. Programski jezik C **pravi razliku između velikih i malih slova (case sensitive)**.

Primer:

```
Mika  mika
```

su dva različita indifikatora **!!!!**

U nastavku su primeri ispravnih identifikatora:

```
broj          znak abc    ime_prez     a_123
ime 23       _temp j     a234b9      retVal1
```

Ključne reči

U tabeli u nastavku data je lista ključnih reči programskog jezika C:

auto	else	Long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	Unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	_packed
double			

Ograničenje:

Rezervisane reči se ne smeju koristiti kao imena konstanti, promenljivih ili bilo kojih drugih identifikatora.

Blanko znak

Blanko znak se upotrebljava za razdvajanje delova naredbe i omogućava prevodiocu gde počinje a gde se završava neki identifikator.

Primer:

```
int tezina;
```

mora biti barem jedan blanko znak između identifikatora `int` i `tezina` kako bi ih prevodilac razlikovao.

U sledećem primeru:

```
voce = jabuke + kruske;
```

blanko znak pre i posle znaka `=` može ali i ne mora da stoji. Obično postoje blanko znakovi zbog čitljivijeg koda.

Tipovi podataka

Tipom podataka u C definišemo skup vrednosti koje može da „ima“ promenljiva tog tipa. Tipovi u C-u mogu se klasifikovati na sledeći način:

#	Naziv	Opis
1	Osnovni tipovi	Dele se na: a) celobrojni tipovi i b) tipovi sa pokretnim zarezom
2	Nabrojivi tipovi	Promenljive tog tipa mogu „uzimati“ samo određene (diskretne) celobrojne vrednosti
3	Tip void	Poseban tip koji označava da nema vrednosti
4	Izvedeni tipovi	Uključuju: a) Pokazivačke tipove b) Nizove c) Strukture d) Unije e) Funkcijske tipove

- 0. Uvod
- 1. Arhitektura računara
- 2. Programski jezici
- 3. Osnove programskog jezika C

- Tokeni
- ;, Komentar, Identifikatori, Ključne reči, Blanko znak
- Tipovi podataka
- Celobrojni tipovi**, Tipovi sa pokretnim zarezom, void tip
- Promenljive,
- Definicija, inicijalizacija promenljivih

Celobrojni tipovi

Promenljive ovog tipa mogu da imaju samo celobrojne vrednosti. U nastavku su pobrojani standardni celobrojni tipovi:

tip	Veličina mem.	Opseg
char	1 bajta	-128 do 127 ili 0 to 255
unsigned char	1 bajta	0 do 255
signed char	1 bajta	-128 do 127
int	2 ili 4 bajta	-32 768 do 32 767 ili -2 147 482 648 do 2 147 482 647
unsigned int	2 ili 4 bajta	0 do 65 535 ili 0 do 4 294 967 295
short	2 bajta	-32 768 do 32 767
unsigned short	2 bajta	0 do 65 535
long	4 bajta	-2 147 482 648 do 2 147 482 647
unsigned long	4 bajta	0 do 4 294 967 295

Problem: Odrediti opseg brojeva koji se mogu predstaviti u jednom 16-bitnom računaru.

Rešenje: Od 16 bitova jedan bit je rezervisan za znak a ostalih 15 ostaje za kodiranje broja u binarnom brojnom sistemu od 0 do 1111111111111111.

$$(1 \times 2^{14}) + (1 \times 2^{13}) + \dots + (1 \times 2^1) + (1 \times 2^0)$$

što iznosi 32767 ili $2^{15} - 1$.

Znači u 16-bitnom računaru može se smestiti decimalni broj iz opsega -32767 do 32767.

Pošto je nula definisana sa 0000000000000000 broj 1000000000000000 bi po analogiji predstavljao „minus nula“. Na račun tog broja proširuje se opseg na -32768 do 32767

1. Arhitektura računara
2. Programski jezici
3. Osnove programskog jezika C

;, Komentar, Identifikatori, Ključne reči, Blanko znak

Tipovi podataka

Celobrojni tipovi, Tipovi sa pokretnim zarezom, void tip

Promenljive,

Definicija, inicijalizacija promenljivih

Celobrojni tipovi

Korišćenjem **sizeof** operatora moguće je saznati koliko koji tip zauzima memorije u određenom operativnom sistemu.

Primer:

```
#include <stdio.h>
#include <limits.h>
```

```
int main()
{
    printf("Velicina memorije za int : %d \n", sizeof(int));
    return 0;
}
```

1. Arhitektura računara
2. Programski jezici
3. Osnove programskog jezika C

;, Komentar, Identifikatori, Ključne reči, Blanko znak

Tipovi podataka

Celobrojni tipovi, Tipovi sa pokretnim zarezom, void tip

Promenljive,

Definicija, inicijalizacija promenljivih

Tipovi sa pokretnim zarezom

Promenljive ovog tipa mogu imati racionalne vrednosti. U nastavku su pobrojani standardni tipovi sa pokretnim zarezom:

tip	Veličina mem.	Opseg	Precisnost
float	4 bajta	1.2E-38 do 3.4E+38	6 decimalnih mesta
double	8 bajtova	2.3E-308 do 1.7E+308	15 decimalnih mesta
long double	10 bajtova	3.4E-4932 do 1.1E+4932	19 decimalnih mesta

Racionalni brojevi r prikazuju se u obliku:

$$r = \pm mb^e$$

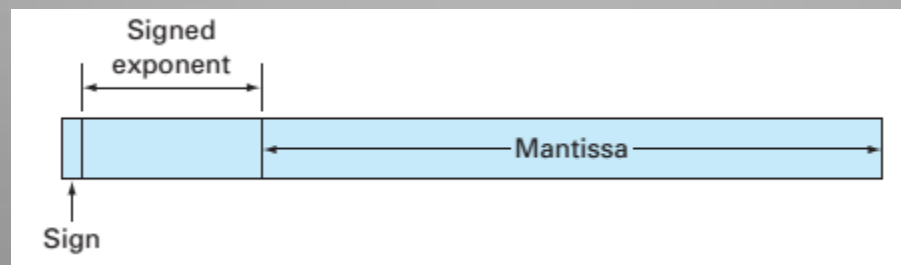
pri čemu je:

b – osnova, 10 – za dekadni a 2 – za binarni br. sistem

e – eksponent, celi broj u određenom rasponu,

m – manitsa, racionalni broj za koji vredi $1/b \leq m < 1$
(tj. mantisa je normalizovana).

$$156.78 \text{ -----} \rightarrow 0.15678 \times 10^3$$



Tipovi sa pokretnim zarezom

Korišćenjem **sizeof** operatora takođe je moguće saznati koliko koji tip zauzima memorije u određenom operativnom sistemu.

Primer:

```
#include <stdio.h>
```

```
#include <float.h>
```

```
int main()
```

```
{
```

```
    printf("Velicina mem. za float: %d \n", sizeof(float));
```

```
    printf("Min float pozitivna vrednost: %E\n", FLT_MIN );
```

```
    printf("Max float pozitivna vrednost: %E\n", FLT_MAX );
```

```
    printf("Preciznost: %d\n", FLT_DIG );
```

```
    return 0;
```

```
}
```

void tip

Ovim tipom se označava da vrednost nije definisana. Koristi se u sledećim situacijama:

- ako funkcija ne vraća nikakvu vrednost
`void exit (int status);`
- ako funkcija nema nikakav ulazni parametar
`int rand (void);`

Promenljive

- Promenljive predstavljaju imena delova operativne memorije u koje program upisuje ili čita vrednost.
- Svaka promenljiva u C-u **mora** imati tačno definisan tip čime se definiše opseg vrednosti koje ona može da sadrži, veličinu memorijskog prostora koji zauzima i skup operacije koje se mogu izvršiti nad njome.
- Ime promenljive može biti sastavljeno od slova, cifara i znaka `_`. Ono mora početi ili sa slovom ili sa znakom `_`.
- Treba obratiti pažnju da C razlikuje mala i velika slova !!!!

0. Uvod

1. Arhitektura računara
2. Programski jezici
3. Osnove programskog jezika C

Tokeni

;, Komentar, Identifikatori, Ključne reči, Blanko znak

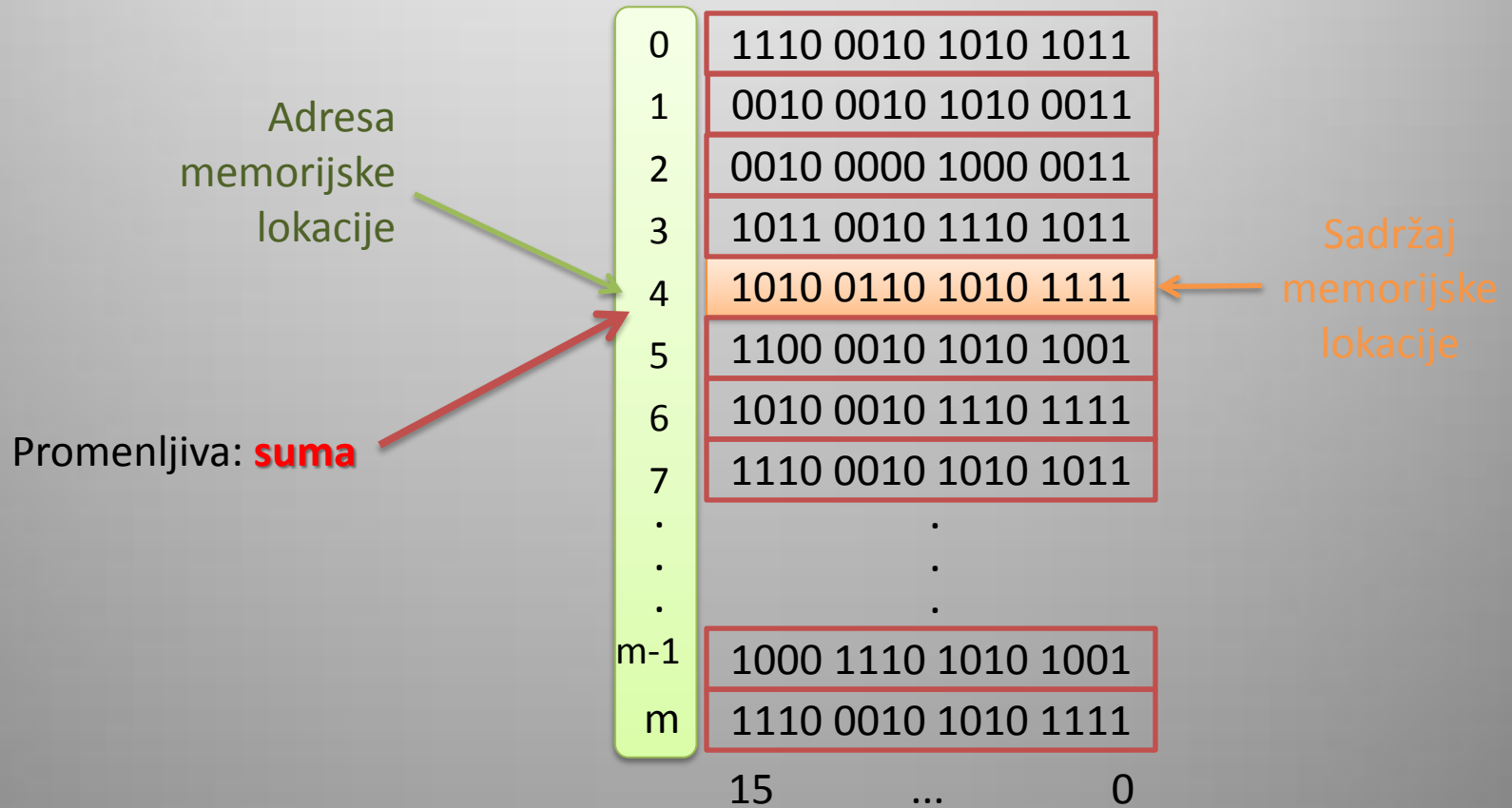
Tipovi podataka

Celobrojni tipovi, Tipovi sa pokretnim zarezom, void tip

Promenljive,

Definicija, inicijalizacija promenljivih

7-8



1. Arhitektura računara
2. Programski jezici
3. Osnove programskog jezika C

;, Komentar, Identifikatori, Ključne reči, Blanko znak

Tipovi podataka

Celobrojni tipovi, Tipovi sa pokretnim zarezom, void tip

Promenljive,

Definicija inicijalizacija promenljivih

Definicija promenljivih

```
tip lista_promenljivih;
```

Primer:

```
int i, j, k;
```

```
char c, ch;
```

```
float f, suma;
```

```
double d;
```

1. Arhitektura računara
2. Programski jezici
3. Osnove programskog jezika C

Definicija i inicijalizacija promenljivih

```
tip lista_promenljivih = vrednost;
```

Primer:

```
int d = 3, f = 5;
```

```
char c = 'a';
```

```
float f = 3.45;
```

Primer:

```
#include <stdio.h>
int main ()
{
// definicija promenljivih
    int a, b;
    int c;
    float f;

// inicijalizacija vrednosti promen.
    a = 10;
    b = 20;
    c = a + b;

    printf("Vrednost prom. c : %d \n", c);

    f = 70.0/3.0;
    printf("Vrednost prom. f : %f \n", f);

    return 0;
}
```

Vrednost prom. c : 30
Vrednost prom. f : 23.333334