

# ***Programiranje 2***

**Odsek za Softversko inženjerstvo**

**Elektrotehnički fakultet  
Beograd**

# ***UVOD***

## **Literatura:**

- ✓ **Laslo Kraus**, *Programski jezik C sa rešenim zadacima*, Akademska misao, 9. izdanje, Beograd, 2014.
- ✓ **Laslo Kraus**, *Rešeni zadaci iz programskog jezika C*, Akademska misao, 4. izdanje, Beograd, 2014.
- ✓ ...

# C

- **Viši programski jezik opšte namene**
- **Počeo da se razvija 1972, Dennis Ritchie, Bell Labs**
- **Cilj – razvoj višeg programskog jezika za sistemsko programiranje**

# ***Prethodnici jezika C***

- **Algol (apstraktan i opšti)**
- **CPL (Combined PL), Cambridge,**  
nepregledan, težak za implementaciju
- **BCPL (Basic CPL), MIT,**  
za sistemsko programiranje
- **B (uprošćenje BCPL-a),**  
Ken Thompson, Bell Labs,  
na njemu pisan UNIX za PDP-7
- **B i BCPL - efikasan interfejs prema HW**  
(bez tipova podataka)

# ***Cilj***

## **Osobine jezika:**

- ✓ jezik nezavisan od računara
- ✓ komfor višeg programskog jezika
- ✓ zamena za zavisne simboličke mašinske jezike
- ✓ konciznost i efikasnost

⇒ nezavisnost ostvarena uvođenjem tipova podataka

# ***Specifičnosti***

- **manipulacija sa bitovima**
- **korišćenje procesorskih registara**
- **pristup podacima pomoću adrese**
- **operatori orijentisani ka hardveru**

# ***Osobine jezika C***

- relativno **“nizak jezik”**  
sa komforom **višeg jezika**
  - ✓ radi sa istom vrstom objekata kao i HW, npr. adrese, brojevi, karakteri
- pogodan za **sistemske**  
**programiranje**, ali opšte namene
  - ✓ UNIX je skoro čitav napisan u C-u
  - ✓ prevodioci za druge jezike

# ***Osobine jezika C***

- omogućava kompaktno strukturirane programe, manje čitljive od Pascal-a
- velika fleksibilnost
- relativno mali jezik, prenosiv
- nije za početnika u programiranju
- standardi: ANSI C X3.159-1989 (C89), ISO C99, ISO C11



# ***Struktura C programa***

- **Primer prostog programa**

```
// Program ispisuje zbir dva broja.  
#include <stdio.h>  
int main () {  
    int a,b,c;  
    printf ("a,b? ");  
    scanf ("%d%d", &a, &b);  
    c=a+b;  
    printf ("a+b=%d\n", c);  
}
```

## ***Struktura C programa***

- **program se sastoji od funkcija, samo *main* obavezna, bibliotečke funkcije**
- **sa *main* počinje izvršavanje**
- **{...} - programski blok = deklaracije + naredbe (oboje opciono)**
- **definicija promenljivih obavezna**

# ***Notacija***

**C koristi skup znakova koji sadrži:**

- **velika slova**
- **mala slova**
- **cifre**
- **specijalne znakove**
- **neke znakovne sekvence koje počinju sa \**

# ***Identifikatori***

- **imena konstanti, promenljivih, tipova podataka, funkcija, labela**
- **moraju da počinju slovom (obično mala slova za promenljive, a velika za konstante)**
- **sastoje se od slova (uključujući i \_) i cifara**
- **ne preporučuje se da počinju sa \_ (često ih koriste bibliotečka imena)**

# ***Identifikatori***

- **velika i mala slova se razlikuju**  
**(a ≠ A)**
- **dužina proizvoljna**
  - ✓ **interna imena**  
**(oblast važenja unutar datoteke)**  
**63 značajna znaka po Standardu**
  - ✓ **spoljašnja imena**  
**(preko granica datoteka, rukuje i OS)**  
**prvih 31 znakova su značajni**

# ***Separatori***

- **razdvajaju leksičke elemente**
- **razmak, tabulacija, kraj linije, komentar**
- **terminator naredbe ;**
- **komentar počinje sa /\*  
završava se sa \*/**
- **komentar koji se završava u istom  
redu počinje sa //**

# ***Rezervisane reči***

- **ključne reči - simboli sa utvrđenim značenjem**
- **ne mogu se koristiti kao identifikatori i u drugom kontekstu od onog koji im je eksplicitno dat**

# ***Rezervisane reči***

auto break case char const continue default do double else enum extern	float for goto if inline (od C99) int long register restrict (od C99) return short	signed sizeof static struct switch typedef union unsigned void volatile while	_Alignas (od C11) _Alignof (od C11) _Atomic (od C11) _Bool (od C99) _Complex (od C99) _Generic (od C11) _Imaginary (od C99) _Noreturn (od C11) _Static_assert (od C11) _Thread_local (od C11)
---	--	---	--

- Prva specifikacija jezika: K&R C (po knjizi iz 1978. godine)
- Prvi Standard: X3.159-1989 (ANSI C, C89) prihvaćen od *International Standards Organization* kao ISO/IEC 9899:1990 (C90)
- Drugo izdanje ISO Standarda ISO/IEC 9899:1999 (C99)
- Treće, važeće izdanje ISO Standarda ISO/IEC 9899:2011 (C11)



# ***Tipovi podataka***

- **apstrakcija i zanemarivanje mašinskog predstavljanja podataka**
- **tip definiše:**
  - ✓ **skup vrednosti**
  - ✓ **operacije nad njima**
- **podaci:**
  - ✓ **konstante**
  - ✓ **promenljive**

# ***Podela tipova podataka***

- **skalarni (prosti)**
  - ✓ **numerički**
  - ✓ **pokazivački**
  - ✓ **nabrajani**
- **struktuirani (složeni)**
  - ✓ **niz**
  - ✓ **struktura**
  - ✓ **unija**

## ***Numerički tipovi***

- **Znakovni**
- **Celobrojni**
- **Realni**

# ***Znakovni tip***

- **char**
  - ✓ **1 bajt**
  - ✓ **kod znaka iz lokalnog skupa znakova**
  - ✓ **može da se koristi kao ceo broj u aritmetičkim operacijama**
  - ✓ **standard ne definiše da li je označen ili ne (neodređenost se izbegava prefiksom)**
- **signed char**  
**(-128 ÷ 127)**
- **unsigned char**  
**(0 ÷ 255)**
- **postoje u C11 char16\_t i char32\_t (iz uchar.h) kao i wchar\_t (iz wchar.h) čija širina je implementaciono zavisna (kodiranja UTF-16, UTF-32, UNICODE)**

# ***Celobrojni tip***

- ✓ **short int (short)**                      tipično 16 bita
- ✓ **unsigned short int**
- ✓ **int**    tipično 16 ili 32 bita
- ✓ **unsigned int (unsigned)**
- ✓ **long int (long)**                      tipično 32 bita
- ✓ **unsigned long int**
- ✓ **long long int (long long)** tipično 64 bita
- ✓ **unsigned long long int**

# ***Celobrojni tip***

- **int** - obično jedna mašinska reč
- **varijante:**
  - ✓ **short** – dužina nije veća od **int**
  - ✓ **long** – dužina nije manja od **int**
  - ✓ **unsigned** – neoznačeni brojevi

## ***Realni tip***

- **float - jednostruka tačnost (32 bita)**  
**(6-7 značajnih cifara)**
- **double - dvostruka tačnost (64 bita)**  
**(15-16 značajnih cifara)**
- **long double – višestruka tačnost**  
**(najčešće kao double ili više – npr. 96 bita)**

# ***Konstante***

- **celobrojne**
- **realne**
- **znakovne**
- **znakovni nizovi**
- **simboličke**
- **nabrajane**



# ***Celobrojne konstante***

- **decimalne**
  - ✓ cifre 0 ÷ 9
  - ✓ ne smeju počinjati sa 0 (nula)
  - ✓ opcioni predznak
- **oktalne**
  - ✓ počinju sa 0
  - ✓ cifre: 0 ÷ 7
- **heksadecimalne**
  - ✓ počinju sa 0x ili 0X
  - ✓ cifre: 0 ÷ 9, a ÷ f (A ÷ F)

# ***Celobrojne konstante***

- **označeni broj tipa int**  
**(ili long ako je većeg opsega)**
- **za 0 i 0X predznak je implicitno određen vodećim bitom**
- **sufiks u ili U  $\Rightarrow$  unsigned**
- **sufiks l ili L  $\Rightarrow$  long**
- **sufiks ll ili LL  $\Rightarrow$  long long**

# ***Realne konstante***

## **Decimalne**

- **Format  $mEk$** 
  - ✓ opciono znak
  - ✓  $m$  - celobrojni deo, tačka, decimalni deo može bez cel. ili dec.
  - ✓ e ili E
  - ✓ označeni eksponent
  - ✓ opciono sufiks
- **primer:**
  - 3.5667E-11L**
  - 1e5**
  - +123.45F**
  - .12L**

## **Heksadecimalne**

- **Format  $0XmPk$ :**
  - ✓ celobrojni deo  $m$  u heksadecimalnom
  - ✓ eksponent  $k$  u decimalnom
  - ✓ vrednost  $m \times 2^k$
  - ✓ mora 0X ili P da bi se razlikovalo od dekadne
- **primer:**
  - 0xA0.Cp+2**
  - 0X1P5**
  - 0Xfe.8p0**

# ***Realne konstante***

- celobrojni deo ili decimalni deo (ali ne oba) može da se izostavi  
primer: .325, -34.
- tačka ili eksponent (ali ne oboje) mogu izostati  
primer: -4E22, 23.754
- Podrazumeva se double
- sufiks f ili F  $\Rightarrow$  float
- sufiks l ili L  $\Rightarrow$  long double

# ***Znakovne konstante***

- broj čija je vrednost jednaka kodu znaka (najčešće ASCII)
- štampanci: 32 ÷ 126
- upravljački: 0 ÷ 31 i 127
- beli znakovi:  
razmak, FF, LF, CR, HT, VT
- znakovna konstanta okružena apostrofima (npr. 'a', '+')
- tip je nominalno int

# ***Znakovne konstante***

## **Reprezentacija negrafičkih karaktera:**

- **'\n'**      **novi red (CR + LF)**
- **'\t'**      **horizontalna tabulacija**
- **'\v'**      **vertikalna tabulacija**
- **'\b'**      **pomak unazad**
- **'\r'**      **CR**
- **'\a'**      **zvučni signal**
- **'\f'**      **FF**
- **'\''**      **apostrof**
- **'\"'**
- **'\?'**
- **'\\'**
- **'\ddd'**    **kod sa najviše tri oktalne cifre**
- **'\xhh'**    **kod sa najviše dve hexa cifre**

# ***Konstante u vidu znakovnih nizova***

- **stringovi između duplih navodnika, npr. "Ovo je niz znakova"**
- **prevodilac ubacuje završni znak sa vrednošću 0 ('\0') za prepoznavanje kraja**
- **"niz znakova" isto što i "niz " "znakova"**

# ***Konstante u vidu znakovnih nizova***

- razlika između 'A' i "A"
- prazan niz "" prevodi se u '\0'
- izbegavati hexa i oktavno navođenje znakova (zbog portabilnosti i mogućnosti greške)
- ako treba " unutar niza - \"



# ***Simboličke konstante***

- **dodela simboličnog imena sa #define**
  - ✓ jedan red
  - ✓ jedna naredba
  - ✓ jedna konstanta
- **ime obično velikim slovima**
- **primeri**
  - ✓ `#define ZVONO '\007'`
  - ✓ `#define PORUKA "Kraj rada"`
  - ✓ `#define PI 3.14159`

# ***Nabrajane konstante***

- celobrojne simboličke konstante
- vrednosti zadate eksplicitno ili implicitno nabranjem identifikatora
- naredba enum (zamenjuje posebne `#define` za identifikatore konstanti)
- identifikatori moraju biti jedinstveni, ali vrednosti ne moraju

# ***Nabrojane konstante***

## **Dodela vrednosti:**

- ✓ **eksplicitno**

- (proizvoljan konstantan izraz tipa int)**

- ✓ **implicitno**

- (za 1 večja od prethodne, počinje od 0)**

- ✓ `enum {NE, DA};`

- ✓ `enum {ZELENA, CRVENA=5, PLAVA};`

- ✓ `enum dani {PON=1, UTO, SRE, CET, PET, SUB, NED};`

# ***Definisanje promenljivih***

- **svaka promenljiva mora da se definiše pre korišćenja**
- **definicijom se određuje osnovni tip i neke osobine promenljive**
- **osnovni tip može biti:**
  - ✓ **elementarni skalarni tip**
  - ✓ **korisnički definisani (npr. nabrajani tip)**

# ***Definisanje promenljivih***

- **promenljivoj se može dati početna vrednost (izraz koji sadrži konstante i ranije definisane podatke koji imaju vrednost)**
- **ako se ne inicijalizuje, ima slučajnu početnu vrednost ili vrednost 0**
- **kvalifikatori ne menjaju tip promenljive**
- **primer:**

```
float eps = 1.0e-8;  
enum dani dan1 = UTO;  
const int boja = PLAVA - 1;  
long dan = 24L*60L*60L*1000L;
```

# ***Definisanje promenljivih***

**const**

- **promenljiva se ne menja tokom programa (obavezna inicijalizacija)**
- **ne može se koristiti tamo gde se eksplicitno zahteva konstanta**
- **pomaže prevodiocu pri optimizaciji**

# ***Definisanje promenljivih***

## **volatile**

- **promenljiva može da promeni vrednost van kontrole programa**
- **prevodioc ne sme vršiti optimizaciju**
- **npr. promenljiva koja predstavlja registar ulaznog uređaja**

# ***Definisanje tipova***

- naredba **typedef**
- obično identifikator tipa ima veliko prvo slovo (da se razlikuje od promenljivih)
- navodi se osnovni tip i izvedeni tip

- **primer:**

```
typedef int Duzina;  
typedef unsigned long int Ceo;  
typedef enum dani D;  
typedef Ceo Racun;
```



# ***Nizovi***

- osnovni struktuirani tip
- komponente istog tipa  
se identifikuju rednim brojem
- niz ima dimenziju k,  
ako su mu komponente  
nizovi sa k-1 dimenzija

# ***Nizovi***

- **dužina niza u []**
- **dužina celobrojna konstanta, a može biti i promenljiva ako je niz automatske trajnosti.**
- **nizovski tipovi se mogu definisati sa typedef**
- **ako je oznaka tipa 1-D niz i ako je ime niza opet 1-D, promenljiva će biti 2-D**
- **smeštanje višedimenzionalnih nizova po vrstama !**

# ***Nizovi***

- mogu se inicijalizovati (vrednosti istog tipa u {} razdvojene zarezima)
- dužina se može izostaviti, ako je inicijalizacija potpuna
- ako ima manje vrednosti od dužine, dodeljuju se samo prvima, ostali nula
- ne sme više vrednosti od dužine

# ***Nizovi***

## **Primeri:**

```
float vek[50], mat[100][30]; // fiksna duz.
int vek2[n], mat2[2*n][n-2]; // prom. duz.
int x[5]={1, 2, 3, 4, 5};
int y[]={5, 4, 3, 2, 1};
int z[2][3]={{5, 10, 15}, {3, 6, 9}};
int t[2][3]={{5}, {3, 6}};
const char p1[]="End";
const char p2[]={ 'E', 'n', 'd', '\\0' };
typedef double Niz[10];
Niz p, q[3];
```

# ***Nizovi***

## **Selekcija komponente:**

- ✓ **indeksi (proizvoljni izrazi)  
idu od 0 do n-1**
- ✓ **nepredvidljive posledice  
ako je indeks izvan opsega**

**Dodela na nivou elementa, ne celog niza**

# ***Nizovi***

## **Primeri:**

```
vek[1] = 155;
```

```
vek[i+(j-1)*n] = a+b*c;
```

```
mat[i][j] = vek[i+j];
```

```
p[7] = q[0][9];
```

# ***Logical***

- **od C99 postoji tip `_Bool` sa vrednostima 0 za logičku neistinu i 1 za logičku istinu (može `bool`, `false` i `true` ako se uključi `stdbool.h`)**
- **u praksi se često koristi tip `int`, kao navika iz prethodnih standarda**
- **ako se proizvoljan numerički podatak koristi na mestu gde se očekuje logički, onda:**
  - ✓ **0 predstavlja logičku neistinu**
  - ✓ **sve ostalo (osim 0) predstavlja logičku istinu**

# ***String***

- **nizovi znakova promenljive dužine**  
**tip char[]**
- **obavezno se završava sa \0**