

Strukture podataka i algoritmi 1

Školska 2023/24

dr Tatjana Stojanović

Organizacija predmeta

- Predavanja i vežbe
 - Programski jezik C
- Literatura
 - B.W. Kernighan, D.M. Ritchie, *Programski jezik C*,
- Polaganje ispita
 - Predispitne obaveze – 70 poena
 - Dolasci 4 poena
 - 2 testa i 1 kolokvijuma (2·13+40) – **po 4 poena sa svakog testa uslov za izlazak na kolokvijum**
 - 36 poena uslov za izlazak na završni deo ispita
 - Završni deo ispita – 30 poena
 - Zadatak 30 poena – obavezni delovi
 - Bonus poeni sa domaćeg za studente koji imaju preko 51 poen

Šta je C?

- Razvijen na UNIX operativnom sistemu
- Programski jezik opšte namene
- Jezik za sistemsko programiranje, ali i za druge domene
- BCPL, Martin Richards, 1967
- B, Ken Thompson, 1970

Šta sadrži C?

- Tipovi podataka
 - Znakovi
 - Brojevi (celi i sa pokretnim zarezom)
 - **Pokazivači**
 - Nizovi
 - Strukture
 - Unije
- Konstrukcije za kontrolu toka
 - Grupisanje naredbi
 - Naredbe grananja
 - Naredbe višestrukog grananja
 - Naredbe ponavljanja
 - Prevremeni prekid petlji

Šta sadrži C?

- Funkcije
 - Vraćaju vrednosti osnovnih tipova, strukture, unije, pokazivače
 - Rekurzija
 - Automatske lokalne promenljive
 - Odvojene izvorne datoteke
 - Različiti opsezi promenljivih
- Pretprocesiranje

Još nešto o C-u

- Jezik niskog nivoa
- Ne sadrži operacije za rad sa složenim objektima
- Nema poseban mehanizam za rezervisanje memorije
- Ne obezbeđuje ulazno/izlazne mehanizme
- Svi mehanizmi višeg nivoa moraju biti obezbeđeni pomoću funkcija koje se pozivaju
- Ne podržava konkurentno programiranje
- ANSI standard 1988 godine

Promenljive i aritmetički izrazi

Program koji prikazuje tabelu temperatura u Farenhajtima i Celzijusima po formuli

$$C = \frac{5}{9}(F - 32)$$

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

```
/* print Fahrenheit-Celsius table
```

```
   for fahr = 0, 20, ..., 300 */
```

```
main()
```

```
{
```

```
    int fahr, celsius;
```

```
    int lower, upper, step;
```

```
    lower = 0;    /* lower limit of temperature scale */
```

```
    upper = 300; /* upper limit */
```

```
    step = 20;   /* step size */
```

```
    fahr = lower;
```

```
    while (fahr <= upper)
```

```
    {
```

```
        celsius = 5 * (fahr-32) / 9;
```

```
        printf("%d\t%d\n", fahr, celsius);
```

```
        fahr = fahr + step;
```

```
    }
```

```
}
```

Zašto ne `celsius = 5 / 9 * (fahr-32)`?

Rezultat

```
printf("%d\t%d\n", fahr, celsius);
```

0	-17
20	-6
40	4
60	15
80	26
100	37
120	48
140	60
160	71
180	82
200	93
220	104
240	115
260	126
280	137
300	148

Rezultat

```
printf("%d\t%d\n", fahr, celsius);
```

```
0      -17
20     -6
40     4
60     15
80     26
100    37
120    48
140    60
160    71
180    82
200    93
220    104
240    115
260    126
280    137
300    148
```

```
printf("%3d\t%6d\n", fahr, celsius);
```

```
  0      -17
 20     -6
 40     4
 60     15
 80     26
100     37
120     48
140     60
160     71
180     82
200     93
220    104
240    115
260    126
280    137
300    148
```

Simboličke konstante

```
#define ime tekst_zamene
```

Primer

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

```
#define LOWER 0 /* lower limit of table */
```

```
#define UPPER 300 /* upper limit */
```

```
#define STEP 20 /* step size */
```

```
/* print Fahrenheit-Celsius table */
```

```
main()
```

```
{
```

```
    int fahr;
```

```
    for (fahr = LOWER; fahr <= UPPER; fahr = fahr + STEP)
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32));
```

```
}
```

Rezultat

```
printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32));
```

```
0      -17.8  
20     -6.7  
40      4.4  
...
```

Ulaz i izlaz znakova

- Tekstualni tok (stream)
- getchar
 - `c=getchar()`
- putchar
 - `putchar(c)`

Primer

Kopiranje ulaznog toka na izlaz.

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

```
/* copy input to output; 1st version */  
main()  
{  
    int c;  
  
    c = getchar();  
    while (c != EOF)  
    {  
        putchar(c);  
        c = getchar();  
    }  
}
```

- EOF (*end of file*) je ceo broj definisan u `stdio` i njegova vrednost se razlikuje od svih char vrednosti
CTRL+Z – Windows CTRL+D – Linux (Unix)

Kraće pisanje

```
#include <stdio.h>

/* copy input to output; 2nd version */
main()
{
    int c;

    while ((c = getchar()) != EOF)
        putchar(c);
}
```

- Prioritet operatora != je viši od operatora dodele =
- Rezultat komande

```
c = getchar() != EOF
```

Brojanje znakova

```
#include <stdio.h>

/* count characters in input;
1st version */

main()
{
    long nc;

    nc = 0;
    while (getchar() != EOF)
        ++nc;

    printf("%ld\n", nc);
}
```

```
#include <stdio.h>

/* count characters in input;
2nd version */

main()
{
    double nc;

    for (nc = 0; getchar() != EOF; ++nc);

    printf("%.0f\n", nc);
}
```

Šta ako ulaz ne sadrži ni jedan karakter?

Brojanje redova

```
#include <stdio.h>

/* count lines in input */
main()
{
    int c, nl;

    nl = 0;
    while ((c = getchar()) != EOF)
        if (c == '\n')
            ++nl;

    printf("%d\n", nl);
}
```

Brojanje reči

```
#include <stdio.h>
#define IN 1 /* inside a word */
#define OUT 0 /* outside a word */

/* count lines, words, and characters in input */
main()
{
    int c, nl, nw, nc, state;

    state = OUT;
    nl = nw = nc = 0;
    while ((c = getchar()) != EOF)
    {
        ++nc;
        if (c == '\n') ++nl;
        if (c == ' ' || c == '\n' || c == '\t') state = OUT;
        else if (state == OUT)
        {
            state = IN;
            ++nw;
        }
    }
    printf("%d %d %d\n", nl, nw, nc);
}
```

ekvivalentno sa `nl = (nw = (nc = 0));`

Nizovi

```
#include <stdio.h>

/* count digits, white space, others */
main()
{
    int c, i, nwhite, nother;
    int ndigit[10];

    nwhite = nother = 0;
    for (i = 0; i < 10; ++i) ndigit[i] = 0;

    while ((c = getchar()) != EOF)
        if (c >= '0' && c <= '9')
            ++ndigit[c-'0'];
        else if (c == ' ' || c == '\n' || c == '\t')
            ++nwhite;
        else
            ++nother;

    printf("digits =");
    for (i = 0; i < 10; ++i) printf(" %d", ndigit[i]);
    printf(", white space = %d, other = %d\n", nwhite, nother);
}
```

TIPOVI, OPERATORI I IZRAZI

Nazivi promenljivih

- Nazivi se sastoje od slova i brojeva
- Znak `_` se smatra slovom i koristi se za poboljšanje čitljivosti dugačkih naziva promenljivih
- Ne počinjite naziv promenljive znakom `_` zato što biblioteke često koriste takve nazive
- C razlikuje velika i mala slova (`x` i `X` su različite promenljive)
- Najčešće se promenljive pišu malim slovima, a konstante sa svim velikim slovima
- Interna imena – prvih 31 znakova značajni; Spoljašnja imena – 6 znakova, bez razlikovanja velikih i malih slova
- **if**, **else**, **int**, **float** itd. su rezervisane reči i ne mogu se koristiti kao promenljive. Pišu se isključivo malim slovima

Tipovi i veličina podataka

Osnovni tipovi

- **char** - 1 bajt, čuva jedan karakter (znak)
- **int** - ceo broj, veličina zavisi od mašine (16 ili 32)
- **float** - broj sa pokretnim zarezom jednostruke tačnosti
- **double** - broj sa pokretnim zarezom dvostruketačnosti

Dodatni kvalifikatori

- **short, long**
 - short **int** (16), long **int** (32)
 - long double
- **signed, unsigned**
 - signed char - od -127 do 128
 - unsigned char - od 0 do 255

Tipovi i veličina podataka

- `<limits.h>`, `<float.h>` - standardne datoteke zaglavlja sa konstantama koje predstavljaju vrednosti limita
- Neke od konstanti u biblioteci `limits.h`

Makro	Vrednost	Opis
<code>SCHAR_MIN</code>	-128	Minimalna vrednost za <code>signed char</code>
<code>SCHAR_MAX</code>	+127	Maksimalna vrednost za <code>signed char</code>
<code>UCHAR_MAX</code>	255	Maksimalna vrednost za <code>unsigned char</code>
<code>SHRT_MIN</code>	-32768	Minimalna vrednost za <code>short int</code>
<code>SHRT_MAX</code>	+32767	Maksimalna vrednost za <code>short int</code>
<code>USHRT_MAX</code>	65535	Maksimalna vrednost za <code>unsigned short int</code>
<code>INT_MIN</code>	-2147483648	Minimalna vrednost za <code>int</code>
<code>INT_MAX</code>	+2147483647	Maksimalna vrednost za <code>int</code>
<code>UINT_MAX</code>	4294967295	Maksimalna vrednost za <code>unsigned int</code>
<code>LONG_MIN</code>	-9223372036854775808	Maksimalna vrednost za <code>long int</code>

Float ili double?

- Za komande

```
printf("%f  %.3f\n",a,a);
```

```
printf("%f  %.3f\n",b,b);
```

Vrednosti	Izlaz
<pre>float a=50.3; double b=50.3;</pre>	<pre>50.299999 50.300 50.300000 50.300</pre>
<pre>float a=5000.3; double b=5000.3;</pre>	<pre>5000.299805 5000.300 5000.300000 5000.300</pre>

Konstante

- **int** `1234`
- **long** `123456789l` ili `123456789L`
- **unsigned long** `123456ul` ili `123456UL`
- **double** `123.4` ili `1e-2`
- **float** `123.4f` ili `123.4F`
- **long double** `12356.78l` ili `12356.78L`

- **dekadni** `31`
 - **oktalni** `037`
 - **heksadekadni** `0x1f` ili `0X1F`

Šta predstavlja `0xFUL`?

Znakovne konstante

- 'a'
- Specijalne sekvence
 - '\n' – novi red
 - '\t' – tabulator
 - '\r' – povratak na pocetak reda
 - '\\' – \
 - '\?' – ?
 - '\0' – nula
- Pisati '0' ili 48?

<code>#define VTAB '\013'</code>	ili	<code>#define VTAB '\xb'</code>	ASCII vertikalni tabulator
<code>#define BELL '\007'</code>	ili	<code>#define BELL '\x7'</code>	ASCII zvučni znak

Konstantni izrazi

- Izrazi u kojima učestvuju samo konstante **izračunavaju se prilikom kompajliranja**

```
#define MAXLINE 1000  
char line[MAXLINE+1];
```

ili

```
#define PRESTUPNA 1 /* u prestupnoj godini */  
int dani[31+28+PRESTUPNA+31+30+31+30+31+31+30+31+30+31];
```