

# Programski jezik C

Osnovni podatkovni tipi

Tomaž Dobravec

# Podatkovni tipi in konstante

- C pozna 5 osnovnih podatkovnih tipov:

Tip	Pomen	Predvideno število bajtov
void	nedoločeno	0
char	znak	1
int	celo število	4
float	realno število (enojna natančnost)	4
double	realno število (dvojna natančnost)	8

- Število bajtov, ki jih zasede posamezen tip, je odvisno od računalnika in prevajalnika

```
|| printf("Velikost tipa int je: %d\n", sizeof(int));
```

# Osnovni številski podatkovni tipi

Vsi številski podatkovni tipi:

- `int`, `unsigned int`, `signed int`, `short int`, `unsigned short int`, `signed short int`, `long int`, `signed long int`, `unsigned long int`,
- `float`, `double`, `long double`.

Osnovna pravila celoštevilskih tipov:

- tip `short int` mora biti vsaj 16-biten,
- tip `long int` mora biti vsaj 32-biten,
- $\text{sizeof}(\text{short int}) \leq \text{sizeof}(\text{int}) \leq \text{sizeof}(\text{long int})$

# Osnovni številski podatkovni tipi

- Najverjetnejše vrednoti na sodobnih računalnikih:

tip	bajti	obseg		
short int	2	-32.768	+32.767	(32kb)
unsigned short int	2	0	+65.535	(64kb)
unsigned int	4	0	+4.294.967.295	(4Gb)
int	4	-2.147.483.648	+2.147.483.647	(2Gb)
long int	4	-2.147.483.648	+2.147.483.647	(2Gb)

Tipi float, double in long double zasedejo 4, 8 in 12 bajtov pomnilnika.

- Natančne podatke najdemo v `limits.h` in `float.h`

# Številске konstante

## Celoštevilске konstante

```
int a= 45;           // a = 45   (desetisko)
int c=012;           // c = 10   (osmisko)
int d=0xFF;          // d = 255  (sestnajstisko)
```

## Realne konstante

```
float x = 3.141593;   // 3.141593
float y = 3e4;         //  $3 \times 10^4 = 30000$ 
float z = 3141593e-6   //  $3141593 \times 10^{-6} = 3.141593$ 
```

# Velikost in obseg celoštevilskih tipov

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

int main(int argc, char **args) {
    char format[] = "%-28s%-10d%20d ... %-20d\n";
    char formatU[] = "%-28s%-10d%20u ... %-20u\n";
    char formatL[] = "%-28s%-10d%20lu ... %-20lu\n";

    printf("Tip                                Velikost                                Obseg\n");
    printf(format, "short int", sizeof(short int), SHRT_MIN, SHRT_MAX);
    printf(formatU, "unsigned short int", sizeof(unsigned short int), 0, USHRT_MAX);
    printf(format, "int", sizeof(int), INT_MIN, INT_MAX);
    printf(formatU, "unsigned int", sizeof(unsigned int), 0, UINT_MAX);
    printf(formatL, "long int", sizeof(long int), LONG_MIN, LONG_MAX);
    printf(formatL, "unsigned long int", sizeof(unsigned long int), 0, ULONG_MAX);
}
```

Tip	Velikost	Obseg
short int	2	-32768 ... 32767
unsigned short int	2	0 ... 65535
int	4	-2147483648 ... 2147483647
unsigned int	4	0 ... 4294967295
long int	8	9223372036854775808 ... 9223372036854775807
unsigned long int	8	0 ... 18446744073709551615

# Znaki

Tip `char` je običajno 8-biten.

ASCII tabela:

0 do 31: kontrolni znaki

32: presledek

48 do 57: števke (od '0' do '9')

65 do 90: velike črke (od 'A' do 'Z')

97 do 122: male črke (od 'a' do 'z')



# ASCII tabela

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	<b>Space</b>	64	40	100	&#64;	<b>@</b>	96	60	140	&#96;	<b>`</b>
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	<b>!</b>	65	41	101	&#65;	<b>A</b>	97	61	141	&#97;	<b>a</b>
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	<b>"</b>	66	42	102	&#66;	<b>B</b>	98	62	142	&#98;	<b>b</b>
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	<b>#</b>	67	43	103	&#67;	<b>C</b>	99	63	143	&#99;	<b>c</b>
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	<b>\$</b>	68	44	104	&#68;	<b>D</b>	100	64	144	&#100;	<b>d</b>
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	<b>%</b>	69	45	105	&#69;	<b>E</b>	101	65	145	&#101;	<b>e</b>
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	<b>&amp;</b>	70	46	106	&#70;	<b>F</b>	102	66	146	&#102;	<b>f</b>
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	<b>'</b>	71	47	107	&#71;	<b>G</b>	103	67	147	&#103;	<b>g</b>
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	<b>(</b>	72	48	110	&#72;	<b>H</b>	104	68	150	&#104;	<b>h</b>
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	<b>)</b>	73	49	111	&#73;	<b>I</b>	105	69	151	&#105;	<b>i</b>
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	<b>*</b>	74	4A	112	&#74;	<b>J</b>	106	6A	152	&#106;	<b>j</b>
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	<b>+</b>	75	4B	113	&#75;	<b>K</b>	107	6B	153	&#107;	<b>k</b>
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	<b>,</b>	76	4C	114	&#76;	<b>L</b>	108	6C	154	&#108;	<b>l</b>
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	<b>-</b>	77	4D	115	&#77;	<b>M</b>	109	6D	155	&#109;	<b>m</b>
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	<b>.</b>	78	4E	116	&#78;	<b>N</b>	110	6E	156	&#110;	<b>n</b>
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	<b>/</b>	79	4F	117	&#79;	<b>O</b>	111	6F	157	&#111;	<b>o</b>
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	<b>0</b>	80	50	120	&#80;	<b>P</b>	112	70	160	&#112;	<b>p</b>
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	<b>1</b>	81	51	121	&#81;	<b>Q</b>	113	71	161	&#113;	<b>q</b>
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	<b>2</b>	82	52	122	&#82;	<b>R</b>	114	72	162	&#114;	<b>r</b>
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	<b>3</b>	83	53	123	&#83;	<b>S</b>	115	73	163	&#115;	<b>s</b>
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	<b>4</b>	84	54	124	&#84;	<b>T</b>	116	74	164	&#116;	<b>t</b>
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	<b>5</b>	85	55	125	&#85;	<b>U</b>	117	75	165	&#117;	<b>u</b>
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	<b>6</b>	86	56	126	&#86;	<b>V</b>	118	76	166	&#118;	<b>v</b>
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	<b>7</b>	87	57	127	&#87;	<b>W</b>	119	77	167	&#119;	<b>w</b>
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	<b>8</b>	88	58	130	&#88;	<b>X</b>	120	78	170	&#120;	<b>x</b>
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	<b>9</b>	89	59	131	&#89;	<b>Y</b>	121	79	171	&#121;	<b>y</b>
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	<b>:</b>	90	5A	132	&#90;	<b>Z</b>	122	7A	172	&#122;	<b>z</b>
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	<b>:</b>	91	5B	133	&#91;	<b>[</b>	123	7B	173	&#123;	<b>{</b>
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<b>&lt;</b>	92	5C	134	&#92;	<b>\</b>	124	7C	174	&#124;	<b> </b>
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	<b>=</b>	93	5D	135	&#93;	<b>]</b>	125	7D	175	&#125;	<b>}</b>
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	<b>&gt;</b>	94	5E	136	&#94;	<b>^</b>	126	7E	176	&#126;	<b>~</b>
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	<b>?</b>	95	5F	137	&#95;	<b>_</b>	127	7F	177	&#127;	<b>DEL</b>



# Znaki

Ker sta število in znak tesno povezana, lahko z znaki tudi "računamo":

```
char a = 65;  // a = 'A' (ASCII tabela)

char b = 'A', c = 'D';
int i = c - b;  // i = 3 ('D' - 'A' = 3)

char p = ' ';
printf("ASCII koda presledka je: %d", p);
```

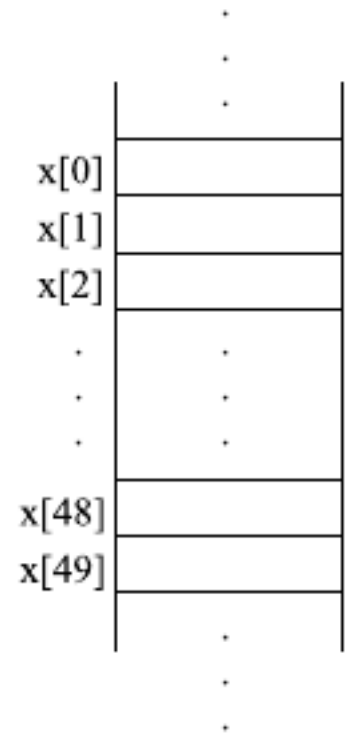
# Znakovne konstante

znak	pomen	znak	pomen
<code>\n</code>	prehod v novo vrstico	<code>\t</code>	tabulator
<code>\b</code>	pomik nazaj	<code>\r</code>	pomik na začetek vrstice
<code>\f</code>	nova stran	<code>\a</code>	zvočni signal
<code>\\</code>	znak <code>\</code>	<code>\?</code>	znak <code>?</code>
<code>\'</code>	znak <code>'</code>	<code>\"</code>	znak <code>"</code>
<code>\0xx</code>	znak v osmiškem zapisu	<code>\xhh</code>	znak v šestnajstiškem zapisu

```
char a = 'A', // znak 'A'
      b = '\n', // nova vrsta
      c = '\a', // zvonček
      d = '\040', // ' ' - 40(OCT) = 32(DEC)
      e = '\x30'; // '0' - 30(HEX) = 48(DEC)
```

# Tabela

```
float x[50];
```



- tabela ima fiksno, vnaprej določeno velikost
- pri deklaraciji tabele povemo njeno velikost: `int x[50]`
- tabele NE moremo deklarirati z `int x[];`

# Tabelarične konstante

Tabelarične konstante podamo v zavutih oklepajih:

```
short meseci[] =  
    {31,28,31,30,31,30,31,31,30,31,30,31};
```

Inicializacija na vrednost 0:

```
int tab[10] = {0}; // vse elemente postavi na 0
```

# Večdimenzionalne tabele

```
int a[3][3] = {{42, 13, 7}, {15, 8, 3}, {1, 17, 5}};
```

	0	1	2	$\leftarrow j$
0	42	13	7	
1	15	8	3	
2	1	17	5	
$i \uparrow$				

dvodimenzionalna tabela  
z elementi  $a[i][j]$

	.
	.
	.
$a[0][0]$	42
$a[0][1]$	13
$a[0][2]$	7
$a[1][0]$	15
$a[1][1]$	8
$a[1][2]$	3
$a[2][0]$	1
$a[2][1]$	17
$a[2][2]$	5
	.
	.
	.

zapis elementov dvodimenzionalne  
tabele v pomnilniku