



Univerza v Ljubljani

Fakulteta *za računalništvo
in informatiko*

Programski jezik C

Tomaž Dobravec

Strukturiranje kode

Vejitveni stavek `if-else`

```
if (a==1)      |   if (x==5) {      |   if (q<5) {  
    b=1;        |       c=1;d=3;        |       d=3;  
                |   } else {            |   } else if (q<2) {  
                |       c=4;            |       d=2;  
                |   }                  |   } else {  
                |                       |       d=1;  
                |                       |   }
```

Strukturiranje kode

Vejitveni stavek `switch`:

```
switch (mesec) {  
    case 2:  
        steviloDni=28;  
    case 4:  
    case 6:  
    case 9:  
    case 11:  
        steviloDni=30;  
    default:  
        steviloDni=31;  
}
```

- Težava: zgornji program ne dela, kot bi pričakovali. Zakaj?
- Pomembno: uporaba ukaza `break` za končanje vsake veje.

Zanke

Zanke do, while, for

<code>int i=0;</code>		<code>int i=0;</code>		<code>int i;</code>
<code>do {</code>		<code>while (i<=10) {</code>		<code>for (i=1;i<=10;i++) {</code>
<code>i=i+1;</code>		<code>i=i+1;</code>		<code>....</code>
<code>....</code>		<code>....</code>		<code>}</code>
<code>} while (i<10)</code>		<code>}</code>		

Funkcije

- Program v C je sestavljen iz funkcij.
- Preprosti programi imajo samo eno funkcijo (`main`), zahtevnejši jih imajo več
- Funkcije predstavljajo ločene celote programa; s pomočjo funkcij program postane preglednejši

Funkcije

- Funkcija (lahko) prejme parametre in vrne rezultat

```
double sredina(double x, double y) {  
    return (x+y)/2;  
}
```

- Funkcijo *pokličemo* takole:

```
double rezultat = sredina(7, 10);
```

Program v več datotekah

- Program v C je lahko napisan v več datotekah.
- Pri prevajanju je potrebno vse datoteke povezati v celoto.
- Globalne spremenljivke lahko uporabljamo v vseh funkcijah programa.
- Funkcijo, ki je napisana v eni datoteki, lahko uporabljamo tudi v drugih datotekah.

5. 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 `floats.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   //  $3141593 \times 10^{-6} = 3.141593$ 
```

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

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', // zvoncek
      d = '\040', // ' ' - 40(OCT) = 32(DEC)
      e = '\x30'; // '0' - 30(HEX) = 48(DEC)
```

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 $a[i][j]$

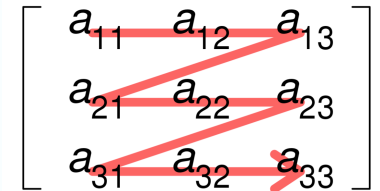
	.
	.
$a[0][0]$	42
$a[0][1]$	13
$a[0][2]$	7
$a[1][0]$	15
$a[1][1]$	8
$a[1][2]$	3
$a[2][0]$	1
$a[2][1]$	17
$a[2][2]$	5
	.
	.
	.

zapis elementov dvodimenzionalne
tabele v pomnilniku

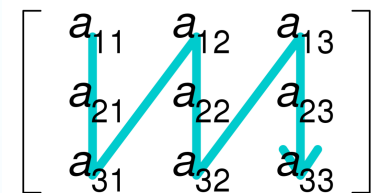
Večdimenzionalne tabele

- C uporablja row-major order za zapis podatkov dvodimenzionalne tabele
- ker so elementi zapisani eden za drugim, lahko sami izračunamo naslov (i,j)-tega elementa in sicer takole:
 - N ... naslov začetka tabele
 - V ... velikost enega elementa tabele
 - naslov elementa $a[i][j]$ je $N + (i * \text{sirina} + j) * V$

Row-major order



Column-major order



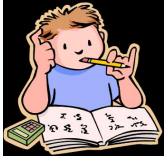
Primer za tabelo `int a[3][3]:`

`N = 0x33FEC4`

`V = 4 (int)`

naslov elementa `a[2][1]` je `0x33FEC4 + (2*3+1)*4 = 0x33FEE0`

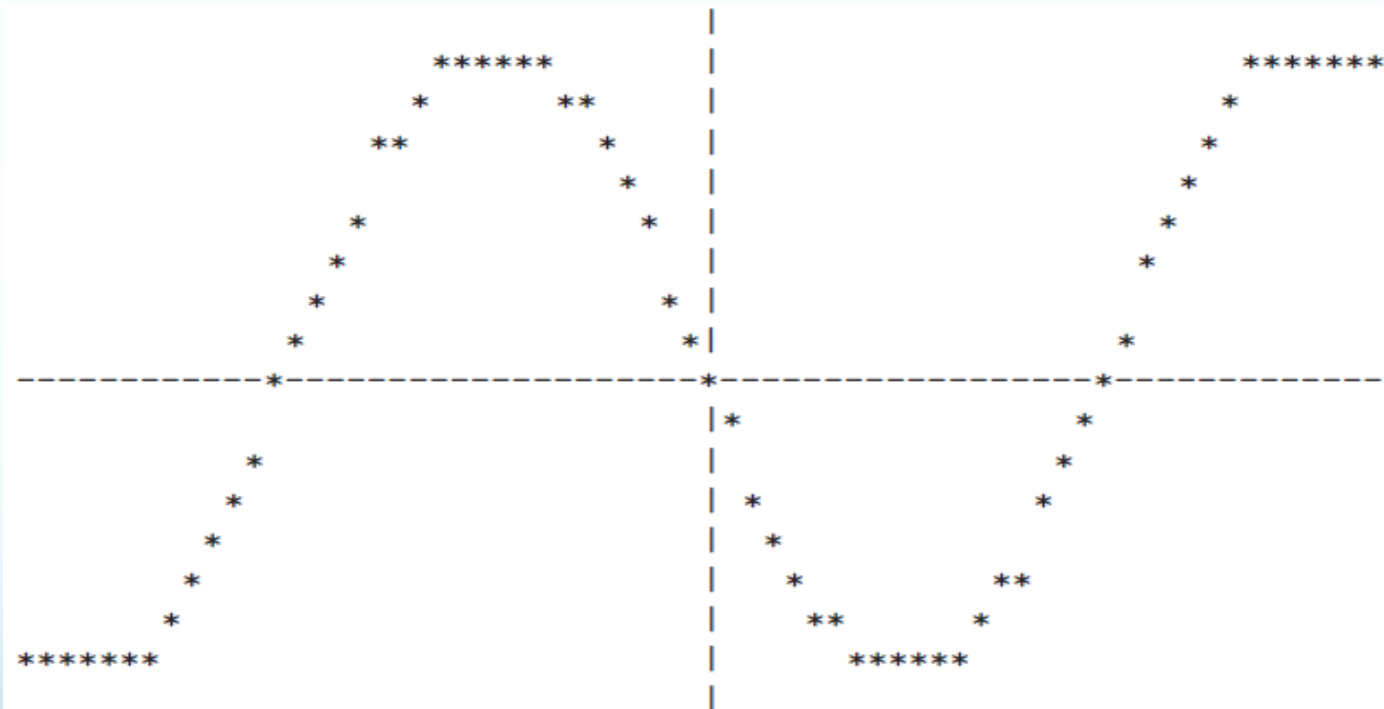
Risanje grafov s pomočjo večdimenzionalne tabele



Naloga 5-V.

(tabele/graf.c)

Napiši program za risanje grafov matematičnih funkcij na tekstoven zaslon.



Slika 5.2: Primer izhoda programa *tabele/graf.c* (funkcija: $-\sin(x)$)

Risanju grafov na tekstovni zaslon

- Tekstovni zaslon ima 25 vrstic in 80 stolpcev.
- Grafa ne rišemo direktno na zaslon ampak v tabelo velikosti 80 x 25.
- Če rišemo funkcijo $f(x): [x1..x2] \rightarrow [y1..y2]$, potem za preslikavo poskrbita funkciji:

$$x = \frac{x2 - x1}{79} * i + x1, \quad j = 24 * \frac{y - y1}{y2 - y1}$$