

Uvod u programiranje znakovni i logički tip podataka

Prikaz slova i ostalih znakova

- Kombinacijom jedinica i nula – kôdom
- Koliko ima znakova?
 - 26 velikih slova engleske abecede A - Z
 - 26 malih slova engleske abecede a - z
 - 10 znamenaka 0 - 9
 - operatori, interpunkcije, upravljački znakovi
- Za prikaz je dovoljan 1 oktet
- ASCII (ISO-7 standard): 7 bita za informaciju + 1 bit za *paritet*
 - ⇒ $2^7 = 128$ različitih znakova
 - ASCII - American Standard Code for Information Interchange
 - ISO - International Organization for Standardization.
- *Paritet*: ako je u informaciji neparan broj bita, bit pariteta postavlja se na 1, inače na 0 (može i obratno: *odd/even parity*). Omogućuje otkrivanje jednostruke pogreške pri prijenosu informacija

2

7-bitni ASCII kod – kontrolni znakovi

- Znakovi za upravljanje ulazno-izlaznim jedinicama računala (*kontrolni znakovi, nonprintable*) nalaze se na pozicijama 0-31

Dec	Binarno	Hex	C-konstanta	Znak	Komentar
0	00000000	0x00	'\0'	NULL	
7	00000111	0x07	'\a'	BELL	Zvučni signal
8	00001000	0x08	'\b'	BS	Brisanje prethodnog znaka
9	00001001	0x09	'\t'	HT	Horizontalni tabulator
10	00001010	0x0A	'\n'	LF	Novi redak
11	00001011	0x0B	'\v'	VT	Vertikalni tabulator
12	00001100	0x0C	'\f'	FF	Nova stranica
13	00001101	0x0D	'\r'	CR	Na početak retka
27	00011011	0x1B		ESC	Escape

3

7 - bitni ASCII kod – ispisljivi znakovi

- Znakovi koji se mogu tiskati (*printable*) nalaze se na pozicijama 32-126

Dec	Binarno	Hex	C-konstanta	Znak	Komentar
32	00100000	0x20	' '		Praznina
slijede posebni znakovi i interpunkcije kao !, ", +, - itd.					
48	00110000	0x30	'0'	0	
49	00110001	0x31	'1'	1	
Slijede ostali brojevi do uključivo 57 ('9'), te još posebnih znakova (do 64)					
65	01000001	0x41	'A'	A	
66	01000010	0x42	'B'	B	
Slijede ostala velika slova engleske abecede do 90 ('Z'), te još posebnih znakova					
97	01100001	0x61	'a'		
Slijede ostala mala slova engleske abecede do 126 ('z'), te posebni znak 127 (DEL)					

- Na poziciji 127 nalazi se još jedan od kontrolnih znakova, znak DEL

4

char - znakovni tip podataka

- Pohrana malih cijelih brojeva sa ili bez predznaka
- Pohrana slova, interpunkcija, posebnih znakova
- Zauzima 1 oktet.
- Definicija varijabli u programskom jeziku C:
`char` [-128, 127]
`unsigned char` [0, 255]
- Za prikaz se koristi međunarodni standard: ASCII kôd (American Standard Code for Information Interchange)

5

Znakovne konstante

'A'	znak iz ASCII tablice
'0'	znamenka nula
'\x41'	heksadekadski zapis slova A
'\101'	oktalni zapis slova A
'\a'	simbolički zapis za znakove koji se ne mogu tiskati (Escape-sekvenca)
'\0'	kraj znakovnog niza (nul-karakter, ništični znak, znak prznoga)
'\n'	prijelaz u novi red
'\\'	znak \"backslash\"
'\''	jednostruki navodnik
'\"'	dvostruki navodnik

6

Primjeri sa znakovnim konstantama

- Primjer: Varijabli `c` tipa `char` pridružiti vrijednost slova A na različite načine:
`c = 'A';` pridružuje ASCII kôd znaka 'A' koji je $65_{10} = 41_{16} = 101_8$
`c = 65;` ili `c=0x41;` ili `c=0101;`
`c = '\x41';` /* heksadekadske konstante počinju s \x */
`c = '\101';` /* oktalne konstante počinju s \ */
- Primjer: Varijabli `c` tipa `char` pridružiti vrijednost jednostrukog navodnika ('), a zatim pridružiti vrijednost znaka \
`c = '\'';` /* specijalni znakovi unutar navodnika moraju imati ispred sebe znak \ */
`c = '\\';`
- Primjer pridruživanja prilikom definicije varijable:
`char r, x='a';`
`r = 'A';`

7

Znamenke 0 do 9 u ASCII tablici

- Kada se znakovni tip koristi za pohranjivanje znamenki 0 do 9 treba obratiti pozornost na to da se u varijablu znakovnog tipa ne pohranjuje brojčana vrijednost te znamenke, nego ASCII vrijednost te znamenke, odnosno redni broj u ASCII tablici:
`char a;`
`a = '1';` ekvivalentno izrazu: `a = 49;`
- U varijabli `a` nalazi se brojčana vrijednost 49, odnosno redni broj znaka '1' u ASCII tablici.
- Ukoliko se želi dobiti brojčana vrijednost znamenke, potrebno je od te vrijednosti oduzeti 48. Vrijednost 48 ustvari predstavlja ASCII vrijednost znaka '0' .

8

Pretvorba ASCII znamenke u cijeli broj

- Treba uočiti da pojedine znamenke prikazane kao znak ne odgovaraju po binarnom prikazu odgovarajućem cijelom broju. Npr. znamenka 7 prikazana kao cijeli broj u 1 oktetu iznosi $0000\ 0111_2$, a prikazana kao ASCII znak $0011\ 0111_2$, odnosno 55_{10} .
- Pretvorba:
 $\text{broj} = \text{znak} - 48; \quad 0011\ 0111\ (55_{10})$
 $\text{broj} = \text{znak} - '0';$
- 8-bitni ASCII kod
0-127 kao u 7-bitnom kodu, 128-255 prijeglasi i grafički znakovi, ovisno o tzv. kodnoj stranici

9

Primjer: Ispisati neke znakove ASCII tablice kao broj i kao znak

```
IspisAsciiZnakova
#include <stdio.h>
int main( ) {
    char x, y, p, q;
    x = 'A';
    y = x + 32;
    p = '\n';
    q = ' ';
    printf("%d %c %d %c %d %c\n", x, x, y, y, '0', '0'+1);
    printf("%d %c %d %c\n", p, p, q, q);
    return 0;
}
```

```
65 A 97 a 48 1
10
32 |
```

10

Primjeri s ASCII vrijednostima znamenaka

- Primjer:
`char c = 'A';`
- Što će se ispisati naredbama:

```
printf ("%c", c);           /* A */
printf ("%d", c);          /* 65 */
printf ("%c", c + 32);     /* a */
printf ("%d", 'B' - 'A');  /* 1 */
```
- Primjer: Zadane su dvije varijable a i b tipa char koje sadrže znamenke ('0'-'9'). Napisati izraz koji će izračunati broj koji odgovara zbroju tih znamenki (npr. za '5' i '6' rezultat treba biti 11.)

```
char a,b;
int i;
i = a - '0' + b - '0';
```

ili

```
i = a + b - 2 * '0';
```

ili

```
i = a + b - 96;           /* 2 * 48 */
```

11

Niz znakova (string)

- Navodi se unutar dvostrukih navodnika npr.
"Fakultet"
"Znak \\ nazivamo \"backslash\""
- Niz znakova uvijek završava NULL znakom ('0'). Na primjer, konstantni znakovni niz "ZAGREB" u memoriji računala zauzima 7 okteta:

Z	A	G	R	E	B	\0
---	---	---	---	---	---	----

- Definicija varijable u programskom jeziku C:

```
char ime_niza[duljina_niza+1];
```

(kao polje znakova, paziti da se rezervira mjesto za '\0')

12

Matematička logika

- Osnovni pojam: logički sud
- Može biti istinit ili lažan
- Primjeri:
 - 1 < 2 je istinit
 - 3 = 4 je lažan
- Osnovni ili atomni sud (atom): istinitost ili lažnost utvrđuje se neposrednim zaključivanjem
- Složeniji sud tvori se formulama koje se sastoje od:
 - atoma
 - logičkih operatora
 - zagrada

17

Matematička logika: osnovni operatori

- Negacija \neg

A	$\neg A$
1	0
0	1
- Konjunkcija \wedge

A	B	$A \wedge B$
0	0	0
1	0	0
0	1	0
1	1	1
- Disjunkcija \vee

A	B	$A \vee B$
0	0	0
1	0	1
0	1	1
1	1	1
- Ekskluzivna disjunkcija \otimes

A	B	$A \otimes B$
0	0	0
1	0	1
0	1	1
1	1	0
- Implikacija \rightarrow
- Dvostrana implikacija \leftrightarrow

18

Matematička logika: korisne ekvivalencije

- S jednom varijablom i konstantama

$\neg(\neg A)$	= A	zakon dvostruke negacije
$A \wedge A$	= A	idempotentnost konjunkcije
$A \wedge 1$	= A	
$A \wedge 0$	= 0	
$A \wedge \neg A$	= 0	
$A \vee A$	= A	idempotentnost disjunkcije
$A \vee 1$	= 1	
$A \vee 0$	= A	
$A \vee \neg A$	= 1	

19

Matematička logika: korisne ekvivalencije

- S dvije varijable

$A \vee B$	= $B \vee A$	komutativnost disjunkcije
$A \wedge B$	= $B \wedge A$	komutativnost konjunkcije
$\neg(A \wedge B)$	= $\neg B \vee \neg A$	de Morganov zakon
$\neg(A \vee B)$	= $\neg B \wedge \neg A$	de Morganov zakon
$A \wedge (A \vee B)$	= A	apsorpcija
$A \vee (A \wedge B)$	= A	apsorpcija
- S tri varijable

$A \wedge (B \wedge C)$	= $(A \wedge B) \wedge C$	asocijativnost konjunkcije
$A \vee (B \vee C)$	= $(A \vee B) \vee C$	asocijativnost disjunkcije
$A \wedge (B \vee C)$	= $(A \wedge B) \vee (A \wedge C)$	distributivnost konjunkcije
$A \vee (B \wedge C)$	= $(A \vee B) \wedge (A \vee C)$	distributivnost disjunkcije

20

Usporedbeni (relacijski) operatori

- Tvore atomne sudove

<i>Operator</i>	<i>Značenje</i>	<i>Logički izraz</i>	<i>Rezultat</i>
==	jednako	1 == 1	1 (Istina)
!=	različito	2 != 2 + 2	1 (Istina)
>	veće	5 > 6	0 (Laž)
>=	veće ili jednako	6 >= 6	1 (Istina)
<	manje	7 < 10	1 (Istina)
<=	manje ili jednako	7 <= 6	0 (Laž)

21

Logički operatori

- Jednostavni relacijski izrazi mogu se kombinirati u složene pomoću logičkih operatora. C uključuje 3 logička operatora:

Operator	Značenje
&&	logičko I
	logičko ILI
!	logičko NE

- U programskom jeziku postoje i logički operatori nad bitovima, obrađeni u jednom od narednih predavanja!

Primjeri:

```
if ((x > 20) && (x < 100))
    printf("x se nalazi u otvorenom intervalu 20-100");
if ((x < 5) || (x > 20))
    printf("x se ne nalazi u zatvorenom intervalu 5-20");
if (!(x > 20))
    printf("x je manji ili jednak 20");
```

22