PROGRAMIRANJE

PROGRAMSKI JEZIK C

Tipovi Podataka
Konverzije tipova podataka

Tipovi podataka

- Tip podataka određuje
  - Skup vrednosti
  - Skup operacija nad vrednostima datog tipa
  - Veličinu memorijskog prostora za smeštanje vrednosti datog tipa
  - Način reprezentacije vrednosti datog
- C jezik podržava dva osnovna tipa numeričkih podataka
  - Celobrojni
    - Uključuje celobrojne vrednosti
  - Razlomljeni tip podataka
    - Uključuje razlomljene vrednosti

Celobrojni tip podataka

- Celobrojni tip podataka int
  - obično podrazumeva (ne obavezno) da se vrednosti reprezentuju sa 32 bita
  - ovo je moguce izmeniti kalifikatorima long odnosno short
- Celobrojni tip u C jeziku se konstruiše i kvalifikatorima long, short odnosno signed i unsigned

short int unsigned short int	int unsigned int	long int unsigned long

  - Podrazumeva se da se ako se ne navede neki od kvalifikatora signed odnosno unsigned radi o označenom (signed) celobrojnom tipu: short int ⟶ signed short int
  - Redosled kvalifikatora nije bitan: long unsigned int ⟶ unsigned long int
  - C jezik dopušta i sledeće skraćenje, može se izostaviti ključna reč int ako se u konstrukciji celobrojnog tipa koristi neki od kvalifikatora long, short odnosno signed, unsigned
- Standard C jezika ne propisuje veličinu memorijskog prostora koji će se koristiti za reprezentaciju vrednosti datog tipa, vec samo sledeća ograničenja

short ≤ int ≤ long

  - Konkretni raspon vrednosti pojedinog tipa sadržan je u datoteci zaglavlja limits.h

Konstante celobrojnog tipa

- Celobrojna konstanta
  - Niz cifara koji ne sadrži decimalnu tačku
  - C jezik dopušta zapis celobrojne konstante u oktalnom, dekadnom ili u heksadekadnom brojnom sistemu
    - Niz cifara oktalnog brojnog sistema 0 1 … 7 koji počinje cifrom 0 je oktalna konstanta
    - Niz cifara dekadnog brojnog sistema 0 1 … 9 koji ne počinje cifrom 0 je dekadna konstanta
    - Niz cifara heksadekadnog brojnog sistema 0 1 … 9 a A b B … f F koji počinje sa 0x ili 0X je heksadekadna konstanta
    - Dakle radi se samo o alternativvnim zapisima jedne iste vrednosti

      47 = 057 = 0x2f
- Tip celobrojne konstante je podrazumevano int, sem ako se drugačije ne specificira
  - Ako se zapis celobrojne konstante završava slovom
    - l odnosno L: odgovarajuća konstanta će biti tipa long int
    - u odnosno U : odgovarajuća konstanta će biti tipa unsigned int
    - ul (redosled i veličina slova nije od značaja) : odgovarajuća konstanta će biti tipa unsigned long int

Ulaz/izlaz celobrojnih vrednosti

- Specifikator konverzije
  - %d: učitavanje odnosno ispisivanje označenih celih brojeva
  - Za učitavanje odnosno ispisivanje neoznačenih celobrojnih vrednosti koriste se sledeći specifikatori
    - %u: učitavanje odnosno ispisivanje decimalnog zapisa neoznačene celobrojne vrednosti
    - %o: učitavanje odnosno ispisivanje oktalnog zapisa neoznačene celobrojne vrednosti
    - %x: učitavanje odnosno ispisivanje heksadekadnog zapisa neoznačene celobrojne vrednosti
  - Ukoliko specifikatoru konverzije d, u, o, x prethodi znak
    - h: učitava se odnosno ispisuje se short vrednost
    - l: učitava se odnosno ispisuje se long vrednost

Razlomljeni tip podataka

- Ako su vrednosti podataka
  - Razlomljene
  - Izrazito velike
  - Izrazito male
- Koritiće se sledeći tipovi podataka
  - float: jednostruka preciznost
  - double: dvostruka preciznost
  - long double: proširena preciznost
- Kao u slučaju celobrojnog tipa podataka standard ne propisuje veličinu memorijskog prostora koji će se koristiti za reprezentaciju vrednosti
  - Utvrđuje se sledeće ograničenje

  float ≤ double ≤ long double
- Konkretan raspon vrednosti za svaki pojedini razlomljeni tip podataka sadržan je u datoteci zaglavlja float.h

Konstante razlomljenog tipa

- Niz cifara koji sadrži
  - decimalnu tačku
  - i/ili eksponent (stepen broja 10)
  - Ako je prisutan eksponent
    - prethodi mu slovo e odnosno E
    - za kojim sledi opcioni znak
    - i na kraju celobrojna konstanta

  57.0 57. 57.0e0 57E0 5.7e1 .57e+2
- Razlomljena konstanta se podrazumevano tretira kao double vrednost
  - U slučaju da se zapis razlomljene konstante završava slovom
    - f odnosno F: odgovarajuća konstanta će biti tipa float
    - l odnosno L: odgovarajuća konstanta će biti tipa long double

Ulaz/izlaz razlomljenih vrednosti

- Specifikatori konverzije za ispis razlomljenih vrednosti
  - %f: ispis razlomljene vrednosti u fiksnom dekadnom formatu (bez eksponenta)
  - %e: ispisuje se razlomljena vrednost u eskponencijalnom formatu
  - %g: format ispisa se bira automatski
    - u slučaju vrlo malih, odnosno vrlo velikih vrednosti koristi se eksponencijalni format (%e)
    - inače se koristi uobičajeni format (%f)
- Specifikatori konverzije za učitavanje razlomljenih vrednosti
  - %f, %e, %g: učitava se razlomljena vrednost tipa float koja opciono može sadržati eksponent
- Ako specifikatoru konverzije e, f, g prethodi znak
  - l: učitava se double vrednost, u slučaju ispisa nema efekta
  - L: učitava se odnosno ispisuje se long double vrednost

Znakovni tip podataka

- char
- Reprezentacija raspoloživih karaktera
  - Vrednosti karakter tipa su karakteri odgovarajućeg karakter set-a (mogu se razlikovati na različitim mašinama)
    - ASCII karakter set …
- Konstante znakovnog tipa
  - Znak između jednostrukih navodnika

```
char ch;
ch='a';
ch='A';
ch='0';
ch=' ';
```

ASCII karakter set

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Operacije na karakter tipu podataka

- Vrednosti karakter tipa se u C jeziku tretiraju kao mali celi brojevi (short int vrednosti)
  - U slučaju ASCII karakter set-a to su celobrojne vrednosti iz zatvorenog intervala [0, 127]

    ‘a’→ 97 ‘A’→ 65 ‘0’→ 48 ‘ ’→ 32
- celobrojne operacije nad odgovarajućim kodovima karakter set-a

Operacije na karakter tipu podataka

```
int i;
char ch;
i='a'; /* i ima vrednost 97 */
ch=65; /* ch ima vrednost 'A' */
ch+=1; /* ch ima vrednost 'B' */
ch++;  /* ch ima vrednost 'C' */
```

```
if('a'<=ch && ch<='z')
    ch=ch-'a'+'A';
```

prepretpostavlja se da su karakteri 'a', 'b', …, 'z' odnosno 'A', 'B', …, 'Z' kodirani uzastopnim vrednostima

- C standard ne propisuje da li su vrednosti karakter tipa podataka označeni ili neoznačeni celi brojevi
  - kontroliše se kvalifikatorima signed odnosno unsigned

escape sekvence

- Neki specijalni karakteri kao (new-line symbol, …)
  - ne mogu se reprezentovati kao znak između jednostrukih navodnika
  - Da bi se obezbedila puna podrška datom karakter set-u utvrđena je sledeća notacija
    - karakter reprezentacija escape sekvence

```
\a  \b  \f  \n  \r  \t  \v  \\  \?  \'  \"
```

  - Karakter \ kojim počinju escape sekvence označava da se izbegava (escape) uobičajena interpretacija karaktera koji sledi
  - karakter reprezentacija escape sekvence ne uključuje sve neprintabilne karaktere već samo one najcešće
    - numerička reprezentacija escape sekvence (zadaje se odgovarajuci kod karakter set-a)
      - oktalna reprezentacija escape sekvence: (najviše) tri oktalne cifre kojima prethodi znak \
      - heksadekadna reprezentacija escape sekvence: (najvise) dve heksadekadne cifre kojima prethodi \x

```
#define ESC '\33'
```

Učitavanje i ispisivanje karakter vrednosti

- Odgovarajući specifikator konverzije je %c

```
char ch;
scanf("%c", &ch);
…
printf("%c", ch);
```

  - scanf funkcija u slučaju %c konverzije ne preskače beline (kao kod numericke konverzije)
  - Ako želimo da scanf funkcija preskoči sve beline do prvog karaktera koji nije belina koristiće se sledeći poziv scanf funkcije

```
scanf("□%c", &ch);
```

- Za učitavanje i ispisivanje karaktera obično se koriste efikasnije funkcije getchar i putchar

```
putchar(ch);
ch=getchar();
```

```
do {
    ch=getchar();
} while(ch!='\n');
```

→

```
while((ch=getchar())!='\n');
```

→

```
while(getchar()!='\n');
```

Konverzije tipova

- Za korektno izvršavanje aritmetičke operacije na računaru pretpostavlja se izmedju ostalog istovetnost reprezentacije vrednosti operanada

```
 00000001
+00000110
 00000111
```

- Implicitna konverzija
  - C jezik dopušta kombinaciju tipova operanada u izrazima
  - Kompajler generiše instrukcije za konverziju tipova
  - Sprovodi se u sledećim situacijama (između ostalog):
    - operandi u aritmetičkim ili logičkim izrazima nisu istog tipa
    - tip vrednosti izraza sa desne strane operatora dodeljivanja nije isti kao tip promenljive sa leve strane

Standardna aritmeticka konverzija

```
int i;
float f;
f+i;
```

Bezbednije je celobrojnu vrednost promenljive i prevesti u float nego obratno

- Promocija
  - Uži tip podataka se konvertuje u širi tip podataka
  - Celobrojna promocija
    - konverzija char, short vrednosti u int odnosno unsigned int vrednost
- Standardna aritmetička konverzija
  - Ako je neki od operanada razlomljenog tipa
    - ako je neki operand long double i drugi operand se konvertuje u long double tip
    - ako je neki operand double i drugi operand se konvertuje u double tip
    - ako je neki operand float i drugi operand se konvertuje u float tip
  - Nema operanada razlomljenog tipa
    - prvo se vrsi celobrojna promocija, a zatim
    - konverzija sledecim redom
    - int → unsigned int → long int → unsigned long int
  - Dakle u slučaju kombinacije označenih i neoznačenih celobrojnih operanada
    - prvo se vrši konverzija signed → unsigned, a zatim
    - uži tip u širi tip

Standardna aritmeticka konverzija

```
int i=-10;
unsigned int u=10;
i<u
```

- Konverzija označene vrednosti (-10) u gornjem slučaju je neuspešna
  - ne postoji odgovarajuća neoznačena reprezentacija negativne vrednosti
  - vrednost poredjenja je neočekivana
    - Obicno kompajler prijavljuje upozorenje

```
comparison between signed and unsigned
```

- Konverzija u naredbi dodeljivanja
  - Nije problematična ako je tip promenljive sa leve strane izraza dodeljivanja najmanje širok kao tip vrednost izraza sa desne strane
  - Problematično, ako to nije slučaj

```
char c;
int i;
float f;
i=c;
f=i;
```

```
int i;
i=842.97;
c=100000;
i=1e20;
f=1e100;
```

Ekplicitna konverzija, cast operator

- Nekada podrazumevana (implicitna) konverzija nije dovoljna. C jezik dopušta eksplicitnu konverziju vrednosti izraza u zahtevani tip podataka

```
(tip) izraz
```

```
float f; frac_part;
frac_part=f-(int)f;
```

  - (tip): unarni operator u C jeziku i kao takav je višeg prioriteta od binarnih

```
int i, j;
float f;
f=i/j;
f=(float)i/j;
```

i/j je celobrojni izraz

nakon konverzije (float)i izvršiće se implicitna konverzija j → float(j)
(nisu neophodne obe eksplicitne konverzije)

Eksplicitna konverzija, cast operator

- Nekada je eksplictna konverzija neophodna

```
long i;
int j=1000;
```

```
i=j*j;
```

promenljiva i tipa long može prihvatiti vrednost 1000*1000 ali kako je izraz j*j tipa int odgovarajuća vrednost se možda ne može izracunati u okiru int tipa podataka

```
i=(long)(j*j);
```

ne rešava problem

```
i=(long)j*j;
```

bezbedno

sizeof operator

- Standard C jezika ne utvrđuje veličinu memorijskog prostora za reprezentaciju vrednosti pojedinih tipova
- sizeof operator kao vrednost vraća tu veličinu (neoznačen ceo broj)

  sizeof(tip)
  - unarni operator i kao takav višeg prioriteta od binarnih
  - sizeof(char) je uvek 1
  - vrednost sizeof operatora za ostale tipove zavisi od implementacije
  - Argument sizeof operatora može biti ime tipa ili izraz
    - ako je argument izraz zagrade nisu potrebne

    sizeof i+j → (sizeof i)+j