

PRIKAZ PODATAKA U RAČUNALU

CJELOBROJNI TIP PODATAKA U RAČUNALU

Cjelobrojni tip podataka (integer) s obzirom na preciznost može se deklarirati kao short ili long. Npr. u programskom jeziku C ne postoji ograničenje na duljinu short-a ili long-a ali postoje pravila:

short ne može biti dulji od int
int ne može biti dulji od long

odnosno za duljinu može se napisati $\text{short} \leq \text{int} \leq \text{long}$

PRIKAZ CIJELIH BROJEVA S PREDZNAKOM I BEZ PREDZNAKA

Cijele brojeve deklariramo s kvalifikatorom **int** ispred kojeg mogu biti kvalifikatori **signed** i **unsigned**.

Signed kvalifikator označava da se u varijabli može pohraniti i negativna vrijednost. Varijabla deklarirana kao **unsigned** (bez predznaka) može pohraniti samo pozitivne vrijednosti što udvostručuje maksimalnu vrijednost koja u njoj može biti pohranjena.

Primjer 4.1:

Varijabla a je deklarirana kao cijeli broj s predznakom:

signed short int a

Ako smo varijabli a pridružili vrijednost -5:

a = -5, prikažite sadržaj 16-bitnog registra u kojem je varijabla pohranjena.

Rješenje:

U registru smještamo broj -5 tako da koristimo pravilo dvojnog komplementa.

Broj 5 je:

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

Primjenimo dvojni komplement.

Broj -5 prikazan metodom dvojnog komplementa izgleda ovako:

1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

U tom slučaju vodeća jedinica može se smatrati predznakom broja (1 → negativan broj).

Primjer 4.2:

Varijabla a je deklarirana kao cijeli broj bez predznaka:

unsigned short int a

U registru imamo zapisan slijedeći sadržaj:

1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Koja je decimalna vrijednost je pohranjena u registru ?

Rješenje:

Budući je varijabla deklarirana kao unsigned (bez predznaka), slijedi da vodeća znamenka 1 ne znači da baratamo s negativnim brojem. Pretvorimo zapisani binarni broj u decimalni i dobivamo:

a=65531

Zadatak 4.1:

Varijabla a je na 32-bitnom operativnom sustavu deklarirana kao:

signed short int a, a u registru imamo zapisan sadržaj:

| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

Koja je decimalna vrijednost je pohranjena u registru ?

Rješenje:

Budući je varijabla deklarirana kao signed (s predznakom), slijedi da vodeća znamenka 1 znači da baratamo s negativnim brojem. Pretvorimo zapisani binarni broj u decimalni metodom dvojnog komplementa i dobivamo:

| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

dvojni komplement:

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

+

1

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

Pretvoreno u decimalni broj:

a=-5

Zadatak 4.2:

Varijabla a je na 16-bitnom operativnom sustavu deklarirana kao:

signed int a, a u registru imamo zapisan sadržaj:

| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

Koja je decimalna vrijednost je pohranjena u registru ?

Rješenje:

| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

Dvojni komplement:

| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

+ 1

| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

Decimalna vrijednost: - **18565**

Zadatak 4.3:

Varijabla a je na 16-bitnom operativnom sustavu deklarirana kao:

unsigned int a, a u registru imamo zapisan sadržaj:

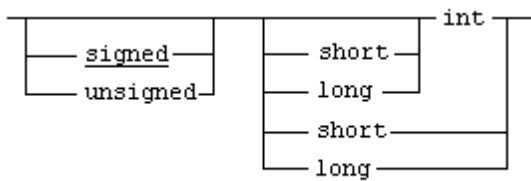
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

Koja je decimalna vrijednost pohranjena u registru ?

Rješenje:

46971

DEKLARACIJA CJELOBROJNOG (INT) TIPa:PRETVORBA TIPOVA PODATAKA



Tip rezultata aritmetičkih izraza određen je tipovima operanada u njemu. Kada se u izrazu nađe više različitih tipova podataka, tip rezultata ovisi o definiranim pravilima za pretvorbu. Pravila pretvorbe različitih tipova podataka orijentirana su prema višem tipu podataka.

Dvije su vrste pretvorbi podataka :

- automatska (implicitna)
- zadana (eksplicitna)

Za automatsku pretvorbu vrijedi:

short int -> unsigned short int -> long int -> unsigned long int -> float -> double -> long double

Zadana pretvorba podataka ima viši prioritet od automatske. Opći oblik deklaracije zadane (eksplicitne) pretvorbe (eng. cast operator) glasi

(tip_podatka) operand

Operand može biti varijabla ili izraz.

Primjer 4.3:

(int) c;

Primjer 4.4:

Riješiti sljedeće primjere za sustav koji za pohranu varijable tipa *short int* koristi 16 bita, a za pohranu varijable tipa *long int* 32 bita.

```
long int l;  
unsigned short int ui1=40000;  
unsigned short int ui2=40000;  
signed short int si1=30000;  
signed short int si2=30000;  
long int li1=30000;
```

Primjer 4.5:

```
l=ui1 + ui2;
```

Rješenje:

Najprije se obavlja zbrajanje na desnoj strani. Rezultat binarnog zbrajanja je short int bez predznaka (jer su tako deklarirane varijable u1 i u2). To znači da dobiveni rezultat u registru nakon zbrajanja odmah pretvaramo u decimalni broj.

```
ui1 =
```

```
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
```

```
ui2 =
```

```
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
```

```
ui1 + ui2=
```

```
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
```

1 - preliv

U registru ostaje binarno: 11100010000000

Decimalno je to 14464.

```
l=14464
```

Primjer 4.6:

```
l =(long)ui1+(long)ui2;
```

Cijeli brojevi bez predznaka su pomoću tzv. cast operatora pretvoreni u long i njihov zbroj je long int.

Rješenje: l=80000

Primjer 4.7:

```
l=si1 +si2;
```

Rješenje:

si1 =30000

| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

si2 =30000

| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

si1+si2

| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

Primjetimo da se rezultat bilježi kao cijeli broj s predznakom, tj short signed int.

1110 1010 0110 0000 prvi bit=1 -> broj je negativan -> dvojni komplement

0001 0101 1001 1111
+ 1
0001 0101 1010 0000

U registru je zapisan decimalni broj: -5536

Primjer 4.8:

l=li1 + si2;

Izraz uključuje long int i signed int. Rezultat je long.

Rješenje: l=60000

CJELOBROJNO DIJELJENJE

Potrebno je obratiti pozornost i na 'neželjene' rezultate zbog cjelobrojnog dijeljenja. Ukoliko se na primjer u realnu varijablu *a* želi pridružiti vrijednost od $\frac{1}{2}$, sljedeća naredba pridruživanja **neće** 'dobro' raditi:

```
a = 1 / 2;
```

U izrazu s desne strane jednakosti postoje dva operanda cjelobrojnog tipa i operacija dijeljenja. U tom slučaju obavlja se cjelobrojno dijeljenje. Rezultat tog izraza je vrijednost 0 (uz ostatak 1). Za izbjegavanje takvih neželjenih rezultata potrebno je koristiti zadanu pretovrbu tipa (dovoljno samo na jednom operandu) ili zadati konstante tako da je barem jedna realna:

```
a = (float) 1 / 2; ili  
a = 1. / 2;          ili  
a = 1 / 2.;
```

Napomena: U prvom slučaju korištena je zadana pretvorba tipa (mogla se uporabiti i nad drugim operandom); u drugom odnosno trećem slučaju, dodavanjem točke konstanta je predstavljena kao realna te se dijeljenje obavlja u realnoj domeni.

Primjer 4.9:

Zadan je izraz:

$a = 5 / 2$

Koja je vrijednost varijable *a*?

Rješenje:

Budući s oba argumenta s desne strane cijeli brojevi, rezultat je cijeli broj 2.

Zadatak 4.4:

Zadan je izraz:

$a = 5 / 2.$

Koja je vrijednost varijable *a*?

Rješenje:

Budući je jedan argumenta s lijeve strane cijeli broj a drugi decimalni, rezultat je decimalni broj 2.5

PRIORITETI OPERATORA

1. * / %

2. + -

Ako i izrazu ima više operatora jednakog prioriteta, izračunavaju se s lijeva nadesno. Izrazi u okruglim zagradama imaju najveći prioritet.

Primjer 4.10:

Koliki je rezultat izraza:

```
5 + 10 / 3 * ( 8 - 6 )
5 + 10 / 3 * 2
5 + 3 * 2
5 + 6
11
```

Primjer 4.11:

Koliko iznosi:

- a) $9 / 4$ Rješenje: 2 (cjelobrojno dijeljenje)
b) $9 \% 4$ Rješenje: 1 (ostatak cjelobrojnog dijeljenja $9:4=2$ i ostatak 1)

Primjer 4.12:

Koju će vrijednost poprimiti varijable i,x,c

```
int i;
double x,c,d;
```

nakon naredbi:

```
d=6.0;
i=(int) (d+1.73);
x=i/2;
c=(float) i/2;
```

Rješenje:

$i=7, x=3, c=3.$

Primjer 4.13:

Prikazati heksadecimalno sadržaj varijable tipa short u koju je pohranjena vrijednost -7.

```
710 = 1112           0000000000000111
(komplement) 1111111111111000
              +                       1
                          111111111111001
                          F     F     F     9
```


OPERATORI S BITOVIMA

&	AND
	OR
^	XOR
<<	SHIFT LEFT
>>	SHIFT RIGHT
~	NOT

Operatori & , |, ^ su binarni (definirani nad dva bita), a operator ~ unarni (definiran nad jednim bitom).

Primjer 4.14:

Izračunati izraze $3 \& 5$, $3 | 5$, $3 \wedge 5$, ~ 3

```

    0000 0011 (3)
& 0000 0101 (5)
-----
    0000 0001 (1)

    0000 0011 (3)
| 0000 0101 (5)
-----
    0000 0111 (7)

    0000 0011 (3)
^ 0000 0101 (5)
-----
    0000 0110 (6)

~ 0000 0011 (3)
-----
   1111 1100 (252 ili -4)
```

Operatori << i >> služe za pomak svih bitova vrijednosti varijable u lijevo ili u desno. Pomak bitova u varijabli za jedno mjesto u lijevo odgovara množenju vrijednosti varijable sa 2, dok pomak za jedno mjesto u desno rezultira dijeljenjem vrijednosti varijable sa 2. Elektronička računala u skupu svojih strojnih naredbi u pravilu imaju naredbe za pomak vrijednosti u registru, i tako izvršeno množenje ili dijeljenje sa višekratnikom broja 2 je bitno brže u odnosu na klasično množenje i dijeljenje. Broj pomaka je određen parametrom.

Primjer 4.15:

Izračunati izraze $2 \ll 1$, $37 \gg 2$

0000 0010 (2), nakon pomaka u lijevo za jedno mjesto,
0000 0100 (4) (rezultat je umnožak vrijednosti sa 2)

0010 0101 (37), nakon pomaka u desno za dva mjesta,
0000 1001 (9) (cjelobrojno dijeljenje vrijednosti sa 4)

Primjer 4.16:

U slučaju da je podatak spremljen u jedan oktet (8 bita) bez bita za predznak, izračunati $128 \ll 1$

1000 0000 (128), nakon pomaka za jedno mjesto,
0000 0000 (0)

Primjer 4.17:

U slučaju da je podatak spremljen u jedan oktet (8bita) sa bitom za predznak, izračunati $64 \ll 1$

0100 0000 (64), nakon pomaka za jedno mjesto,
1000 0000 (-128)