

Programski jezik C

Operatorji in izrazi

Tomaž Dobravec

Operatorji in izrazi

- Aritmetični operatorji

unarni operator (en sam operand): $-$ in $+$

```
int a=-1;  
int b=+15;    // isto kot int b=15;
```

binarni operator (dva operanda) $+$, $-$, $*$, $/$, $\%$

```
int a=8, b=2, c;  
  
c = (a + b) * 3;    // c = 30;  
c = a / b;          // 4 (celostevilsko deljenje)  
c = 33 % 5;         // 3 (ostanek po deljenju s 5)  
  
float x = 3, y = 2, z;  
z = x / y;          // z = 1.5 (realno deljenje)
```

Relacijski in logični operatorji

- Relacijski operatorji
 - < manjše
 - <= manjše ali enako
 - > večje
 - >= večje ali enako
 - == enako
 - != različno
- Logični operatorji
 - ! negacija
 - && konjunkcija (logični in)
 - || disjunkcija (logični ali)

Relacijski in logični operatorji

- Logični izrazi se preverjajo od leve proti desni. Izračun vrednosti logičnih izrazov se konča takoj, ko je znana končna vrednost.
- Pravilo ARL: Aritmetični operatorji imajo prednost pred relacijskimi, relacijski pa pred logičnimi

`i < a - 1`

isto kot

`i < (a-1)`

`a < 5 && b > 3`

isto kot

`(a<5) && (b>3)`

Relacijski in logični operatorji

- Če je logični izraz neresničen, je njegova vrednost enaka nič.
- Če je logični izraz resničen, je njegova vrednost različna od nič.

```
if (a)      ... isto kot ...   if (a != 0)
if (!a)     ... isto kot ...   if (a == 0)

printf("%d, %d", (4 < 5), (13 == 42)); // 1, 0
```

Operatorja ++ in --

... isto kot ...

++i;	i = i + 1;
--i;	i = i - 1;
x = ++n;	n = n + 1; x = n;
y = --i;	i = i - 1; y = i;
t[++i]=a;	i = i + 1; t[i] = a;

... isto kot ...

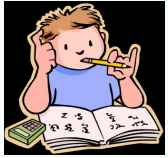
i++;	i = i + 1;
i--;	i = i - 1;
x = n++;	x = n; n = n + 1;
y = i--;	y = i; i = i - 1;
t[i++]=a;	t[i] = a; i = i + 1;

Operatorji +=, -=, *=, /= in %=

... isto kot ...

i += 5;	i = i + 5;
i -= 2;	i = i - 2;
i /= 2;	i = i / 2;
i *= 2;	i = i * 2;
i %= 2;	i = i % 2;

Kaj izpiše program?



Vprašanje

(razno/operatorji.c)

- Kaj izpiše spodnji program?

```
main() {  
    int a=7, b=-1, c=2, d=5;  
    if (a+a && ++b && c--) d=1;  
    printf("%d %d %d %d \n", a, b, c, d);  
}
```


Bitni operatorji

- C pozna naslednje bitne operatorje:

&	in	AND
	ali	OR
^	ekskluzivni ali	XOR
<<	pomik bitov v levo	LSHIFT
>>	pomik bitov v desno	RSHIFT

- ~ eniški komplement NOT

01011	01011	01011	~01011
&00101	00101	^00101	
=00001	=01111	=01110	=10100

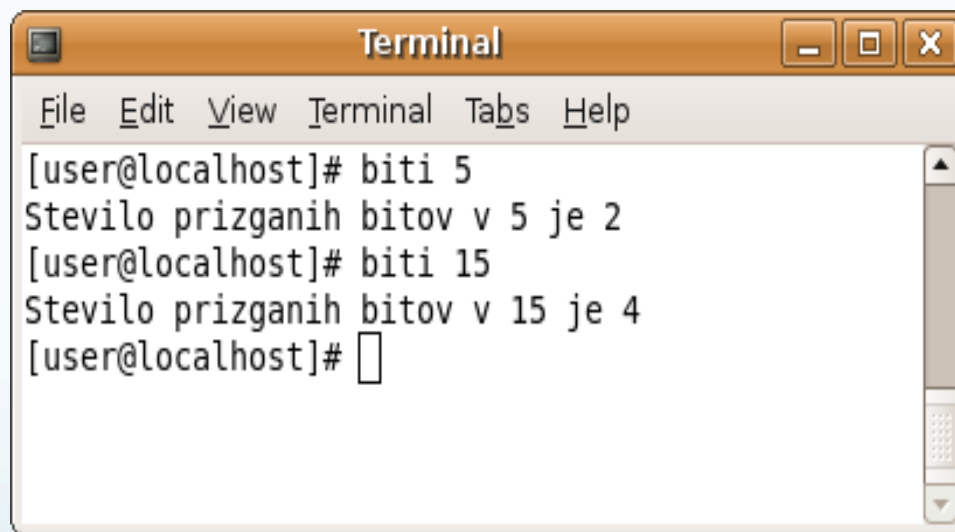
Bitni operatorji



Naloga 3-I.

(operatorji/bit1.c)

Napiši program, ki izpiše število prižganih bitov v prvem argumentu programa.



```
Terminal
File Edit View Terminal Tabs Help
[user@localhost]# biti 5
Stevilo prizganih bitov v 5 je 2
[user@localhost]# biti 15
Stevilo prizganih bitov v 15 je 4
[user@localhost]#
```

Operator ?

Namesto

```
if (pogoj)
    rezultat = izraz1;
else
    rezultat = izraz2;
```

lahko pišemo tudi

```
rezultat = pogoj ? izraz1 : izraz2;
```

Primer:

```
printf("%s", x < 37 ? "OK" : "VROCINA!");
```

Prireditveni stavek

Prireditveni stavek vrne vrednost.

Primer: rezultat prirejanja `b = a` je `a`.

Uporaba:

- `c = b = a;`
- ```
while ((c=getchar()) != '\n') {
 ...
}
```

# Prioriteta in asociativnost operatorjev

Vprašanje 1: Se bo v

$$d = a + b * c;$$

najprej izračunal seštevek ali zmnožek?

Odgovor: najprej zmnožek, saj ima \* večjo prioriteto kot +.

# Prioriteta in asociativnost operatorjev

Vprašanje 2: Se bo v

$$d = 8 / 4 / 2;$$

najprej delilo z 2 ali s 4?

Odgovor: najprej s 4, saj je / levo-asociativen.

# Prioriteta in asociativnost operatorjev

| Operatorji                                          | Asociativnost |
|-----------------------------------------------------|---------------|
| () [] -> .                                          | L             |
| ! ~ ++ -- + - * & sizeof                            | D             |
| * / %                                               | L             |
| + -                                                 | L             |
| << >>                                               | L             |
| <, <=, >=, >                                        | L             |
| == !=                                               | L             |
| &                                                   | L             |
| ~                                                   | L             |
|                                                     | L             |
| &&                                                  | L             |
|                                                     | L             |
| ?:                                                  | D             |
| =, + =, - =, * =, / =, % =, & =, =,   =, << =, >> = | D             |
| ,                                                   | L             |

Tabela: Prioriteta in asociativnost operatorjev