**OPERACIJE S BITOVIMA**

* **operatori** za rad s bitovima:

|  |  |
| --- | --- |
| Simbol | Značenja |
| | | ILI |
| & | I |
| ^ | Ekskluzivno ILI |
| >> | Desni pomak |
| << | Lijevi pomak |
| ~ | Jedinični komplement |

* navedeni operatori mogu se primjenjivati samo za podatke cjelobrojnog tipa:
  + char - najmanje 8 bitova
  + short - najmanje 16 bitova
  + int - najmanje 16 bitova
  + long - najmanje 32 bita
* bitovne operacije I, ILI te ekskluzivno ILI koriste se za „maksiranje“ tj. prekrivanje određenih bitova u operandima -> služe za transformacije u binarnom zapisu broja
* **operatori pridruživanja:**

|  |  |
| --- | --- |
| Operatori pridruživanja | Kraći zapis |
| x = x >> y | x >>= y |
| x = x << y | x <<= y |
| x = x & y | x &=y |
| x = x | y | x |=y |
| x = x ^ y | x ^= y |

* **prioriteti** bitovnih operatora:
  1. I (&)
  2. ekskluzivno ILI (^)
  3. ILI (|)
* u C-u nije moguće raditi s binarnim konstantama pa se služimo heksadekadskim konstantama prilikom zapisa maske
* ispred heksadekadske konstante pišemo **0x**, a ako ispisujemo ili učitavamo broj u heksadekadskom obliku, format je **%x**
* **binarna vrijednost 0101 0101 0101 0101 => heksadekadska vrijednost 0x555**

**OPERATOR BITOVNO ILI**

|  |  |  |
| --- | --- | --- |
| X | Y | X | Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

* izvodi operaciju nad bitovima dvaju operanda (disjunkcija - OR)
* **X | Y**
* želimo li postaviti određene bitove operanda x na 1, napravit ćemo to pomoću operatora ILI te pomoćnog operanda y -> u masku y postavimo 1 na one pozicije gdje želimo dobiti 1 u operandu x

**OPERATOR BITOVNO I**

|  |  |  |
| --- | --- | --- |
| X | Y | X & Y |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

* izvodi operaciju nad bitovima dvaju operanda (konjunkcija- AND)
* **X & Y**
* želimo li postaviti određene bitove operanda x na 0, koristit ćemo operator I te pomoćni operator y -> u masku y postavit ćemo 0 na one pozicije gdje želimo dobiti 0 u operandu x

|  |  |  |
| --- | --- | --- |
| X | Y | X ^ Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**OPERATOR BITOVNO EKSKLUZIVNO ILI**

* izvodi operaciju nad bitovima dvaju operanda (binarni operator)
* **X ^ Y**
* koristimo kada želimo odrediti koji su bitovi operanda x i y različiti te je moguće maskom y promijeniti bitove operandu x s 1 na 0 ili obrnuto

**DESNI I LIJEVI POMAK**

* izvode operaciju nad bitovima dvaju operanda (binarni operatori)
* **X >> Y ; X << y**
* prvi operand je podatak cjelobrojnog tipa nad kojim se izvršava operacija, a drugi predstavlja broj bitova za koje će se izvršiti pomak
* drugi operand mora biti tipa unsigned int i ne može biti negativan broj te mora biti manji od broj bitova koje sadrži prvi operand jer je u suprotnom rezultat 0

**JEDINIČNI KOMPLEMENT**

* unarni operator (operacija koja se izvodi samo na jednoj varijabli) - (negacija - NOT)
* **~ X**
* jedinični kompliment koristimo za pretvaranje jedinice u nule i obratno
* dvojni kompliment dobivamo tako da jedinični kompliment zbrojimo s 1