

Operatori logici pe biți

1 Noțiuni teoretice

Operatorii logici pe biți se aplică bit cu bit numai datelor de tip întreg, mai precis reprezentărilor interne ale acestora. În continuare ne referim la modul în care se comportă acești operatori în C++.

Operatorii logici pe biți sunt: negația (\sim), conjuncția ($\&$), disjuncția (\mid), disjuncția exclusivă (\wedge), deplasarea spre stânga $<<$ și deplasare spre dreapta ($>>$).

Operatorul de negație este un operator unar care aplicat unui număr întreg x returnează un număr întreg a cărui reprezentare internă se obține din reprezentarea internă a lui x prin complementarea față de unu a fiecărui bit ($\sim 1 = 0$ și $\sim 0 = 1$).

Exemplu:

Considerând $int\ x = 25$, reprezentarea internă a lui x (pe 4 octeți) este: 000000000000000000000000000011001.

Atunci $\sim x$ returnează 1111111111111111111111111100110.

Aceasta reprezentare corespunde unui număr negativ. Pentru a determina valoarea absolută a acestuia urmăm pașii inversi operației de complementare:
 $11111111111111111111111100110-1 = 111111111111111111111111111100101$
 și aplicând fiecărui bit operatorul \sim obținem $0000000000000000000000000011010$
 care corespunde numărului 26.

Deci, dacă $x = 25$ atunci $\sim x = -26$.

În general pentru x număr întreg (cu semn) $\sim x = -x - 1$.

Observatie:

Pentru *unsigned int* $x = 25$, reprezentarea internă a lui x pe 4 octeți este tot 000000000000000000000000011001, iar $\sim x$ returnează 1111111111111111111111111100110. De această dată însă numărul este interpretat ca unul natural și valoarea corespunzătoare este $2^{32} - 1 - (2^4 + 2^3 + 1) = 4294967270$.

($1\&1 = 1$, iar $0\&0 = 1\&0 = 0\&1 = 0$)
 Exemplu: Dacă $x = 5$ și $y = 12$ atunci reprezentările interne ale celor două numere sunt:

Exemplu: Dacă $x = 5$ și $y = 12$ atunci reprezentările interne ale celor două numere sunt:

care are corespondentul zecimal 4.

$$(1|0 = 0|1 = 1|1 = 1, \text{ iar } 0|0 = 0)$$
[illegible]

Operatorul de disjuncție exclusivă (\wedge) este un operator binar care returnează numărul întreg a cărui reprezentare internă se obține prin disjuncția exclusivă a biților situați pe aceeași poziție în reprezentările interne ale operanzilor. ($1 \wedge 1 = 0 \wedge 0 = 0$, iar $1 \wedge 0 = 0 \wedge 1 = 1$)

[illegible]

Operatorul de deplasare spre stânga (shift left) << este un operator binar care returnează numărul întreg a cărui reprezentare este obținută din reprezentarea internă a primului operand prin deplasarea la stânga cu un număr de biți egal cu al doilea operand. Biții rămași liberi se completează cu 0.

[illegible]

Operatorul de deplasare spre dreapta (shift right) >> este un operator binar care returnează numărul întreg a cărui reprezentare este obținută din reprezentarea internă a primului operand prin deplasarea la dreapta cu un număr de biți egal cu al doilea operand. Biții rămași liberi se completează cu bitul de semn al numărului inițial.

Observație: Expresia $x \gg n$ are ca efect împărțirea întreagă a operandului x la 2^n .

2 Aplicații

- Indicație:** Dacă n este par cel mai din dreapta bit este 0, altfel este 1. Astfel, se poate folosi conjuncția dintre n și 1.

- Indicație: Folosiți operatorul de deplasare la stânga.

- Indicație:** Câtul se obține folosind deplasarea la dreapta, iar restul folosind conjuncția dintre n și 7.

- Indicație:** n este o putere a lui 2 dacă și numai dacă reprezentarea sa internă conține un sigur bit egal cu 1. Dacă presupunem că acest bit ocupă poziția k , atunci pentru $n - 1$ toți biții din dreapta lui k devin 1, iar bitul k și toți biții din stânga sa devin 0. În consecință n este o putere a lui 2 dacă și numai dacă $n \& (n - 1) == 0$.

5. Fiind dat un număr natural n să se afișeze reprezentarea lui n în baza 2.
Indicație: Bitul de pe poziția i este dat de expresia logică $(n \gg i) \& 1$.
6. Scrieți un program care afișează numărul cifrelor de 1 din reprezentarea internă a unui număr natural n .
7. Scrieți un program care afișează cea mai mare putere a lui 2 care divide un număr natural dat n .