

2. Műveletek és operátorok

2.1 Aritmetikai operátorok

Öt fajta létezik a C nyelvben : +, -, *, /, %.

Ha két egész számot osztunk, akkor az eredmény a hányados egész része lesz. A % operátor két egész osztása esetén a maradékot adja eredményül, float ill. double típusra nem alkalmazható !

Pl.: $5/3 = 1$ és $5\%3 = 2$.

Ezen operátorok közül az előjel (-) precedenciája a legmagasabb, utána következik a /, *, % majd a +, -. Az aritmetikai operátorok balról jobbra kötnek. Ennek a ténynek főleg azonos precedenciájú műveleteknél van jelentősége. Pl.: $4/2*2 = 4$ és nem 1. A műveletek végrehajtási sorrendjét zárójelezéssel befolyásolhatjuk : '(' ill. ')'. Ebben az esetben : $4/(2*2) = 1$. A zárójelezés precedenciája ugyanis megelőzi a többi operátorét !

Fontos : Ha valamely összetett művelet eredménye int típusú és a műveletben résztvevő változók is int típusúak, ettől függetlenül valamely rész-művelet eredménye túllépheti az int ábrázolási tartományát (túlcsordulás !). Ebben az esetben célszerű típusmódosítást előírni.

példaprogram :

```
#include <stdio.h>

int main(int argc, int *argv[])
{
    int i,j,k,q;

    printf("\nKérem J értékét : ") ; scanf("%d",&j);
    printf("\nKérem K értékét : ") ; scanf("%d",&k);
    printf("\nKérem Q értékét : ") ; scanf("%d",&q);

    i = (j*k)/q ; printf("\nSimán -> I = %d",i);

        // Típusmódosítással :

    i = (long) (j)*k/q ; printf("\n(LONG) -> I = %d",i);
}
```

2.2 Relációs operátorok

Azonos precedenciájúak : >, >=, <, <=.

Eggyel alacsonyabb szinten azonos precedenciájúak : ==, != .

Az == operátor két operandus egyenlőségét vizsgálja és ezzel a jelölésmóddal

