

Heltallsdivisjon og rest – div og mod

La a og b være to heltall med $a \neq 0$. Vi sier at a går opp i b (eng. a divides b) hvis det finnes et heltall c slik at $b = ac$.

I så fall kalles a for en faktor i b og b kalles et multiplum av a .

(eng. b is a multiple of a).

Symboler:

$a \mid b$ betyr at a går opp i b .

$a \nmid b$ betyr at a ikke går opp i b .

Eksempler

3 går opp i 12 fordi $12 = 3 \cdot 4$, dvs. $3 \mid 12$

4 går opp i 28 fordi $28 = 4 \cdot 7$, dvs. $4 \mid 28$

5 går ikke opp i 28 fordi tallet 28 kan ikke skrives med 5 som faktor, dvs. $5 \nmid 28$

Observasjoner

- Tallet 1 går opp i alle hele tall. ($x = 1 \cdot x$)
- Alle hele tall $\neq 0$ går opp i 0. ($0 = x \cdot 0$)
- Alle hele tall $\neq 0$ går opp i seg selv.

Når vi faktoreriserer et tall vil alle faktorene i tallet gå opp i tallet, f. eks:

$$12 = 4 \cdot 3 = 2 \cdot 6 = 2 \cdot 2 \cdot 3.$$

I tillegg vet vi at $12 = 1 \cdot 12$, og at alle tall går opp i seg selv.

Følgelig går både 1, 2, 3, 4, 6 og 12 opp i 12.

Regneregler

1. Hvis $a \mid b$ og $a \mid c$, så $a \mid (b+c)$

Bevis:

$$b = a \cdot x$$

$$c = a \cdot y$$

$$b + c = a \cdot x + a \cdot y = a(x + y)$$

Siden a er faktor i $(b+c)$ gjelder $a \mid (b+c)$

2. Hvis $a \mid b$ og $a \mid c$, så $a \mid (mb+nc)$

Bevis:

$$b = a \cdot x$$

$$c = a \cdot y$$

$$mb + nc = m \cdot a \cdot x + n \cdot a \cdot y = a(m \cdot x + n \cdot y)$$

Siden a er faktor i $(mb+nc)$ gjelder $a \mid (mb+nc)$

3. Hvis $a \mid b$ og $b \mid c$, der $b \neq 0$, så $a \mid c$

Bevis:

$$b = a \cdot x$$

$$c = b \cdot y = a \cdot x \cdot y.$$

Siden a er faktor i c gjelder $a \mid c$

4. Hvis $a \mid b$, så $a \mid b \cdot c$ for alle c

Bevis:

$$b = a \cdot x$$

$$b \cdot c = a \cdot x \cdot c$$

Siden a er faktor i $b \cdot c$ gjelder $a \mid b \cdot c$

5. Hvis $a \mid b$ og $a \nmid c$, så $a \nmid (b+c)$

Bevis:

$$b = a \cdot x$$

$a \nmid c$ betyr at a ikke er faktor i c . Følgelig kan vi ikke sette a utenfor parentesen slik vi gjorde det under regel 1. Dermed blir a heller ikke faktor i $(b+c)$.

Eksempel på bruk av regel 1 og 5

Går 7 opp i 101?

Regel 1 sier at hvis $a \mid b$ og $a \mid c$, så $a \mid (b+c)$.

$101 = 77 + 24$. Hvis 7 går opp i begge tallene vet vi at 7 går opp i summen av dem, dvs. i 101.

Vi vet at 7 går opp i 77 (fordi $77 = 7 \cdot 11$), men 7 går ikke opp i $101 - 77 = 24$. Derfor kan ikke 7 gå opp i 101 (regel 5).

Går 3 opp i 1001?

$1001 = 999 + 2$. Hvis 3 går opp i begge tallene vet vi at 3 går opp i summen av dem, dvs. i 1001.

Vi vet at 3 går opp i 999 ($3 \cdot 333$), men 3 går ikke opp i $1001 - 999 = 2$. Dermed vet vi at 3 ikke går opp i 1001 (regel 5).

Divisjonsalgoritmen

La a og d være hele tall der $d > 0$.

Da finnes det entydige, hele tall q og r slik at

$$a = d \cdot q + r, \text{ der } 0 \leq r < d$$

Her kalles a for *dividend*, d for *divisor*, q for *kvotient* og r for *rest*.

$$\begin{array}{r}
 \text{dividend } a \\
 16 : 3 = 5 \leftarrow \text{kvotient } q \\
 \underline{-15} \\
 = 1 \\
 \uparrow \\
 \text{rest } r
 \end{array}$$

Generelt har vi:

$$\begin{array}{r} a : d = q \\ - d \cdot q \\ \hline = r \end{array}$$

$$a - d \cdot q = r$$

$$a = d \cdot q + r$$

Eksempler

$a = 33, d = 17$. Da blir $q = 1$ og $r = 16$.

$$\begin{array}{r} 33 : 17 = 1 \\ - 17 \\ \hline = 16 \text{ i rest} \end{array}$$

$a = 123, d = 7$. Da blir $q = 17$ og $r = 4$.

$$\begin{array}{r} 123 : 7 = 17 \\ - 7 \\ \hline 53 \\ - 49 \\ \hline = 4 \text{ i rest} \end{array}$$

$a = 7, d = 11$. Da blir $q = 0$ og $r = 7$

$$\begin{array}{r} 7 : 11 = 0 \\ - 0 \\ \hline = 7 \text{ i rest} \end{array}$$

$a = 0$ og $d = 3$. Da blir $q = 0$ og $r = 0$.

$$\begin{array}{r} 0 : 3 = 0 \\ \underline{0} \\ 0 \text{ i rest} \end{array}$$

Definisjon av div og mod (modulus)

$$a \text{ div } d = q$$

(tilsvarer heltallsdivisjon i Java: $a/d == q$, der a , d og q er variabler av typen integer)

$$a \text{ mod } d = r$$

(tilsvarer modulus i Java: $a \% d == r$, der a , d og q er variabler av typen integer)

Div og mod ved gjentatt subtraksjon

Vi starter med a og d . Vi trekker fortløpende d fra a inntil vi får et resultat som er mindre enn d . Resultatet blir resten r og antallet ganger vi trakk fra blir g .

I Java:

```

public static void divisjon(int a, int d)
{
    int r = a < 0 ? -a : a; // skifter fortegn hvis a er negativ
    int q = 0;

    while (r >= d)
    {
        r -= d;
        q++;
    }

    if (a < 0)
    {
        q = -(q + 1);
        r = d - r;
    }

    System.out.println("Kvotient: " + q + " Rest: " + r);
}

```

OBS! Den matematiske definisjonen av kvotient q og rest r sier at

$0 \leq r < d$, der d er divisor.

Dette betyr at r **aldri** kan være negativ!

Eksempel

Hvis $a = -123$ og $d = 7$.

Siden a er negativ finner algoritmen først q og r for det tilsvarende positive tallet, dvs. 123. (her blir $- - 123 = 123$).

Når vi går ut av *while*-løkka er q lik 17 og r lik 4. Men dette er ikke de riktige verdiene!

$$a = d \cdot q + r, \text{ men } 17 \cdot 7 + 4 = 123 \neq -123$$

Regnestykket stemmer ikke!

Med utgangspunkt i de midlertidige verdiene for q og r kan de korrigeres på følgende måte ved hjelp av tilordningssetningene (Java):

$q = -(q + 1);$ // dvs. q får verdien $-(17 + 1) = -18$.

$r = d - r;$ // dvs. r får verdien $7 - 4 = 3$.

Når vi nå sjekker disse verdiene ser vi at de stemmer:

$$a = d \cdot q + r = (-18) \cdot 7 + 3 = -126 + 3 = -123$$