

Grootste gemene deler (zonder deling)

Iedereen kent wel de code voor de grootste gemene deler van twee getallen:

```
: GGD ( x y -- ggd ) begin tuck mod ?dup 0= until ;
```

In noForth gaat dit altijd goed, ook met negatieve getallen of nul als input, maar dat geldt niet voor iedere forth.

De volgende code werkt met **unsigned** getallen en handelt nullen correct af.

UGGD

```
: UGGD ( u1 u2 -- uggd )  
  begin                                cr .s  
    2dup umin  
    while tuck 0 swap um/mod drop  
    repeat or ;
```

CR .S staat erin om te laten zien hoe vaak UGGD de lus uitvoert, maar het is ook handig voor het opsporen van fouten, als .S tenminste niet te veel rommel afdrukt.

Zijn u1 en u2 beide deelbaar door n? Dan is hun verschil ook deelbaar door n.

Dit betekent dat we MOD zouden moeten kunnen vervangen door een aftrekking. Om underflow te voorkomen wordt steeds de **grootste** van de twee unsigned getallen vervangen door de **grootste minus de kleinste**.

UGGD- met aftrekken i.p.v. delen

```
: UGGD- ( u1 u2 -- uggd )  
  begin key? ?abort                    cr .s  
    2dup umin dup  
    while >r umax r> tuck -  
    repeat drop or ;
```

KEY? ?ABORT geeft de mogelijkheid om het programma af te breken, want aftrekken is simpel, maar in sommige gevallen wordt de lus wel heel vaak uitgevoerd en dan is het handig om eruit te kunnen stappen.

Kortom, de winst is dat we de deling kwijt zijn, maar de prijs daarvan is dat de lus soms **hopeloos** vaak uitgevoerd wordt.

Wat je ook als winst zou kunnen beschouwen is:

- ▶ De code is te vertalen naar machinecode. (Zonder ?ABORT en .S natuurlijk.)
- ▶ De code is aan te passen voor unsigned dubbel-getallen.

Opgave: UGGD--

```
: UGGD-- ( u1 u2 -- uggd ) ... ; \ een verbeterde versie van UGGD-
```

Met **het kappen van nullen** is het mogelijk om het aantal doorlopen van de lus drastisch te beperken. Wat je daarvoor moet doen is:

- ▶ Bepaal en bewaar het aantal gemeenschappelijke factoren 2 van u1 en u2.
- ▶ Verwijder alle factoren 2 uit u1 en uit u2.
- ▶ In de lus: verwijder na iedere aftrekking alle factoren 2 uit het verschil.
- ▶ Voeg aan het einde het bewaarde aantal factoren 2 weer toe aan het resultaat.

ZEROS

Maak hiervoor het hulpwoord ZEROS

```
: ZEROS ( u -- n ) ... ;  
256 zeros . 8  
8 zeros . 3  
3 zeros . 0  
0 zeros . 0 \ N.B.
```

n is het aantal factoren 2 van u (het aantal binaire eind-nullen).

Uitzondering: 0 geeft 0 als resultaat.

Omdat alle getallen oneven worden gemaakt is het resultaat van de aftrekking altijd even en dat wordt ook weer oneven gemaakt. Dit begrenst het aantal lusdoorlopen, dat nu in de orde van grootte komt te liggen als bij de MOD-versie.

Succes!

A.N. augustus 2026

Voorbeelden op de volgende pagina

:)15 1400 uggd .
(15 1400)
(1400 15)
(15 5)
(5 0)5 OK.0

:)15 1400 uggd- .
(15 1400)
(15 1385)
(15 1370)
(15 1355)
(15 1340)
(15 1325)
(15 1310)
(15 1295)
(15 1280)
(15 1265)
(15 1250)
(15 1235)
(15 1220)
(15 1205)
(15 1190)
(15 1175)
(15 1160)
(15 1145)
(15 1130)
(15 1115)
(15 1100)
(15 1085)
(15 1070)
(15 1055)
(15 1040)
(15 1025)
(15 1010)
(15 995)
(15 980)
(15 965)
(15 950)
(15 935)

(15 920)
(15 905)
(15 890)
(15 875)
(15 860)
(15 845)
(15 830)
(15 815)
(15 800)
(15 785)
(15 770)
(15 755)
(15 740)
(15 725)
(15 710)
(15 695)
(15 680)
(15 665)
(15 650)
(15 635)
(15 620)
(15 605)
(15 590)
(15 575)
(15 560)
(15 545)
(15 530)
(15 515)
(15 500)
(15 485)
(15 470)
(15 455)
(15 440)

(15 425)
(15 410)
(15 395)
(15 380)
(15 365)
(15 350)
(15 335)
(15 320)
(15 305)
(15 290)
(15 275)
(15 260)
(15 245)
(15 230)
(15 215)
(15 200)
(15 185)
(15 170)
(15 155)
(15 140)
(15 125)
(15 110)
(15 95)
(15 80)
(15 65)
(15 50)
(15 35)
(15 20)
(15 5)
(5 10)
(5 5)
(5 0)5 OK.0

:)15 1400 uggd-- .
(175 15)
(15 5)
(5 5)
(5 0)5 OK.0

:)-1 dup u. 4294967295 OK.1
:) 1234567890 OK.2
:)uggd .
(-1 1234567890)
(1234567890 591263625)
(591263625 52040640)
(52040640 18816585)
(18816585 14407470)
(14407470 4409115)
(4409115 1180125)
(1180125 868740)
(868740 311385)
(311385 245970)
(245970 65415)
(65415 49725)
(49725 15690)
(15690 2655)
(2655 2415)
(2415 240)
(240 15)
(15 0)15 OK.0

:)-1 dup u. 4294967295 OK.1
:) 1234567890 OK.2
:)uggd-- .
(617283945 -1)
(617283945 1838841675)
(617283945 610778865)
(610778865 813135)
(813135 304982865)
(813135 152084865)
(813135 75635865)
(813135 37411365)
(813135 18299115)
(813135 4371495)
(813135 444795)
(444795 92085)
(92085 176355)
(92085 42135)
(42135 24975)
(24975 2145)
(2145 11415)
(2145 4635)
(2145 1245)
(1245 225)
(225 255)
(225 15)
(15 105)
(15 45)
(15 15)
(15 0)15 OK.0

:)-1. 2dup du. 18446744073709551615 OK.2
:) 12345678901234567890. OK.4
:)duggd du.
(6172839450617283945 18446744073709551615)
(6172839450617283945 6136952311546133835)
(6136952311546133835 17943569535575055)
(17943569535575055 1529752185502639695)
(17943569535575055 23622009624485385)
(17943569535575055 2839220044455165)
(2839220044455165 7552174745559945)
(2839220044455165 1178238675276195)
(1178238675276195 830490684589485)
(830490684589485 173873995343355)
(173873995343355 328308344623065)
(173873995343355 77217174639855)
(77217174639855 24164205175875)
(24164205175875 13263242365995)
(13263242365995 1362620351235)
(1362620351235 1487577751845)
(1362620351235 62478700305)
(62478700305 650070825465)
(62478700305 73449015645)
(62478700305 2742578835)
(2742578835 29868060735)
(2742578835 6781370475)
(2742578835 504848955)
(504848955 279716235)
(279716235 14070795)
(14070795 2075355)
(2075355 749715)
(749715 165705)
(165705 292005)
(165705 31575)
(31575 67065)
(31575 17745)
(17745 6915)
(6915 5415)
(5415 375)
(375 315)
(315 15)
(15 75)
(15 15)
(15 0) 15 OK.0