## Modulo

Modulo is niks anders dan uitrekenen van het rest getal. Het wordt soms ook wel eens klokrekenen genoemd. Als je een digitale klok hebt gebruik je modulo om te berekenen hoe laat het is. Stel er staat 20 op de digitale klok, dan weet je dat het 8 uur s’avonds is: 20 mod 12 = 8. Andere voorbeelden:

26 (mod 8) = 2  
-13 (mod 8) = 5  
257 (mod 8) = 1

## Relatief priem.

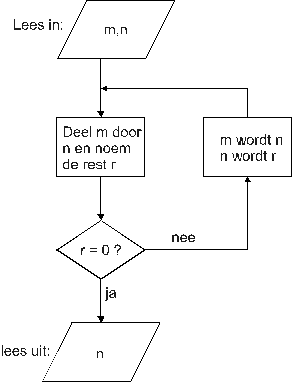
Twee gehele getallen worden ten opzichte van elkaar relatief priem (ook wel copriem) of onderling ondeelbaar genoemd, wanneer er geen positief geheel getal groter dan [1](http://nl.wikipedia.org/wiki/1_(getal)) bestaat dat beide getallen deelt.

### grootste gemene deler (ggd);

De grootste gemene deler, afgekort tot ggd, van een aantal gehele getallen (waarvan er ten minste één ongelijk is aan 0) is het grootste positieve gehele getal, waar al deze gehele getallen door gedeeld kunnen worden zonder dat er een rest overblijft. De grootste gemene deler van de getallen 8 en 12 is bijvoorbeeld 4.

Het algoritme van Euclides wordt gebruikt om bij grotere getallen toch snel de grootste gemene deler te vinden.

1. Noem het grootste van de beide getallen *m*, het andere *n*.
2. Deel m door *n*, bereken hoeveel je overhoudt en noem dat *r*.
3. Wanneer er 0 over blijft zijn we klaar, en is *n* de ggd.
4. Zo niet, herhaal dan het algoritme met *n* en *r*.



1. Kies twee grote priemgetallen *p* ≠ *q* willekeurig en onafhankelijk van elkaar.
2. Bereken *N* = *p \* q*.
3. Kies een geheel getal *e* dat tussen de 1 en de *N* in ligt en relatief priem is t.o.v. *(p-1)\*(q-1).* Dit is *e*
4. Bereken de modulo inverse *d* van *e mod (p-1)\*(q-1)*. Er moet dus gelden dat  *(e \* d) mod (p-1)\*(q-1) = 1*.
5. Vernietig alle sporen van *p* en *q*.

### Versleutelen en ontcijferen

Veronderstel dat Bob een bericht *m* naar Alice wil zenden. Hij kent *N* en *e* (publieke sleutel), want die heeft Alice hem gezonden. Hij zet de klare tekst *m* om in een getal *n* < *N*, gebruik makend van een eerder afgesproken, omkeerbaar protocol. Bijvoorbeeld, elk teken in een bericht kan worden omgezet in zijn 8 bit lange ASCII-code, wat vertaald kan worden naar een getal. Dan berekent hij de cijfertekst *c*:

 c = n^e\ \mathrm{mod}\ N

Alice ontvangt *c* van Bob, en kent haar geheime sleutel *d*. Ze kan *n* te weten komen uit *c* op de volgende manier:

 c^d\  \mathrm{mod}\ N = n 

Alice kan dan *n* vinden, aangezien *n* < *N*, en uit *n* kan ze dan het oorspronkelijke bericht *m* vinden.

1. Alice wil graag een bericht krijgen van Bob. Ze gebruikt het RSA systeem om het bericht van Bob veilig over te laten komen. Ze gebruikt de priem getallen 7 en 17 en rekent uit dat *N* gelijk is aan 119. Maak gebruik van de rekenmachine op de computer.
   1. Laat zien met een berekening dat we *e = 5* kunnen gebruiken.
   2. Laat zien met een berekening dat als *e = 5*  dat *d* gelijk wordt aan 77.
   3. Bob ontvangt dus *e = 5* en *N = 119.* Bob wil graag ‘n’ (ASCII-code 110) terugsturen. Laat zien met een berekening dat hij het getal 94 naar Alice stuurt.
   4. Laat de berekening zien dat Alice kan gebruiken om het ontvangen bericht te ontcijferen.
2. Bob krijg N = 12317 en e = 7 binnen. Hij wil ‘y’ (ASCII-code 121) naar Alice sturen. Bereken welk getal hij naar Alice stuurt.
3. Alice krijgt het getal 95 binnen en heeft als sleutel: *N = 91*, *e = 5* en *d =29.* Bereken welk getal Bob heeft gestuurd in check in een ACSII-tabel welke letter dit is.