



Van PSD naar Small Basic

Vertalen van structuren

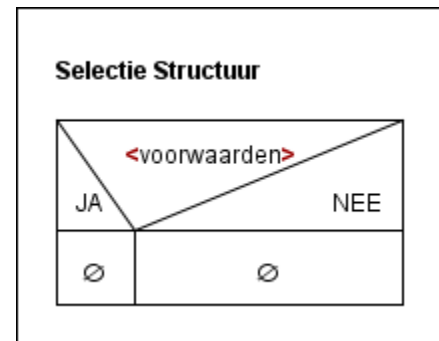
Voor de sequentie structuur zijn er geen speciale opdrachten in een programmeertaal. Programma instructies worden in volgorde van uitvoer opgeschreven.

Voor de **selectie structuur** zijn er meerdere mogelijkheden. De basis opdracht waarmee in deze les gewerkt zal worden is de If-instructie.

Small Basic

```
IF <voorwaarden> Then  
    <opdrachten>  
ELSE  
    <opdrachten>  
EndIF
```

PSD

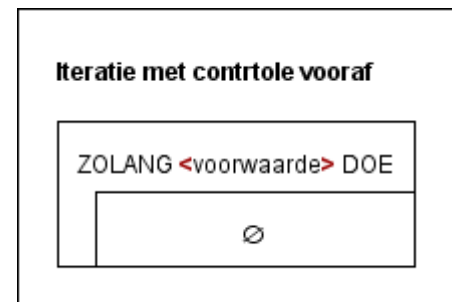


Voor de iteratie (herhaal) structuur zijn er ook meerdere mogelijkheden. Voor deze lessen zullen we voornamelijk gebruik maken van twee.

Small Basic

```
While <voorwaarde>  
    <opdrachten>  
EndWhile
```

PSD



```
For <variabele> = 1 To <getal> Step <getal>  
    <opdrachten>  
EndFor
```





Van PSD naar Small Basic

Nesten

Structuren kunnen genest worden, dit wil zeggen in elkaar gezet worden. Je kunt bijvoorbeeld meerdere selectie structuren in elkaar hebben en er is ook één variant.

Variant selectie structuur:

```
If <voorwaarde> Then
    <opdrachten>
ElseIf <voorwaarde> then
    <opdrachten>
Endif
```

In de PSD hiernaast is te zien hoe dit vertaald is.

Dit had echter ook met de normale selectie geschreven kunnen worden als volgt:

```
If <voorwaarde> Then
    <opdrachten>
Else
    If <voorwaarde> then
        <opdrachten>
    Endif
Endif
```

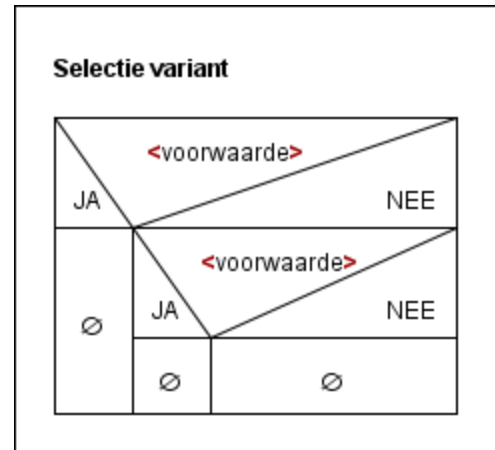
Dit is een voorbeeld waar de If-structuur genest wordt gebruikt. Het zelfde is ook mogelijk met de twee herhaal-structuren.

Bijvoorbeeld:

```
For x = 1 to 10
    For y = 1 to 10
        <opdrachten>
    EndFor
EndFor
```

```
While <voorwaarde>
    While <voorwaarde>
        <opdrachten>
    Endwhile
EndWhile
```

De herhaal-structuren bestaan uit een begin opdracht en een eind opdracht. **Het is heel belangrijk deze goed aan te geven anders zal het programma niet werken zoals verwacht.**





Van PSD naar Small Basic

De PSD opdrachten SCHRIJF en LEES

SCHRIJF

De PSD opdracht **SCHRIJF** vertaal je naar `Write` of `WriteLine`. Deze zijn beide methodes van het object `TextWindow`.

Het verschil tussen `Write` en `WriteLine` is dat bij `Write` de computer nieuwe tekst meteen achter de vorige tekst plakt. Terwijl bij `WriteLine` er eerst naar een nieuwe regel gegaan wordt.

Voorbeelden:

PSD

```
SCHRIJF <variabele>  
SCHRIJF "Hallo Wereld!"  
SCHRIJF "Aantal: ";Aantal  
SCHRIJF "Prijs is ";Prijs;" AWG."
```

Small Basic

```
TextWindow.Write(<variabele>)  
TextWindow.Write("Hallo Wereld!")  
TextWindow.Write("Aantal:" + Aantal)  
TextWindow.Write("Prijs is " + Prijs +  
" AWG.")
```

De zelfde **syntax** (*schrijfwijze*) wordt gebruikt voor de `WriteLine` opdracht.

LEES

Voor de LEES opdracht van PSD's zijn er in Small Basic ook weer twee opdrachten. Namelijk `Read` en `ReadLine`. Net als bij `Write` en `WriteLine` is hier het verschil dat na de `ReadLine` opdracht de computer naar een nieuwe regel gaat op het beeldscherm.

Ook de `Read` en `ReadLine` opdrachten zijn methoden van had object `TextWindow`.

Hieronder zie je enkele voorbeelden van het gebruik van de `Read/ReadLine` opdrachten.

```
LEES <variabele>          TextWindow.Read(<variabele>)  
LEES "Voer een getal in: ";Getal  TextWindow.Write("Voer een getal in:")  
                                  TextWindow.Read(Getal)
```

De tweede LEES opdracht moet in Small Basic vertaald worden naar twee instructies. Eerst een `Write` opdracht en dan een `Read` of `ReadLine` opdracht.



Van PSD naar Small Basic

Vraag:

Wat gebeurt er al je in plaats van Write een WriteLine opdracht gebruikt?

Antwoord:

Invoer van de gebruiker op de volgende regel komen en niet direct achter de tekst dan zal de

RANDOM

Tijdens de eerdere lessen van PSD's is er een opdracht gebruikt om willekeurige getallen te genereren (*PSD voor het spel Getal Raden in les 2*).

De functie om een willekeurig getal te genereren bevindt zich bij het object **Math**. De functie zelf is [GetRandomNumber](#)

PSD:

```
Getal := RANDOM(<getalwaarde>)
```

Small Basic:

```
Getal = Math.GetRandomNumber(<max getal>)
```

De functie genereert een willekeurig getal tussen nul en het getal dat voor max getal is in gevuld.

Bijvoorbeeld:

```
Getal = Math.GetRandomNumber(10)
```

Zorgt ervoor dat de variabele Getal een waarde tussen 1 en 10 krijgt.

Criteria of wel voorwaarden

In de voorgaande lessen zijn deze twee woorden vaker voor gekomen, maar is echter tot nu toe niet echt besproken hoe de expressies voor een voor waarde tot stand komen.

Bij een voorwaarde wordt een expressie vergeleken. Er wordt gekeken of dat de expressie waar of niet waar is. Bijvoorbeeld:

```
If A > 10 Then  
    <Ja Opdrachten>  
Else  
    <Nee Opdrachten>  
EndIf
```



Van PSD naar Small Basic

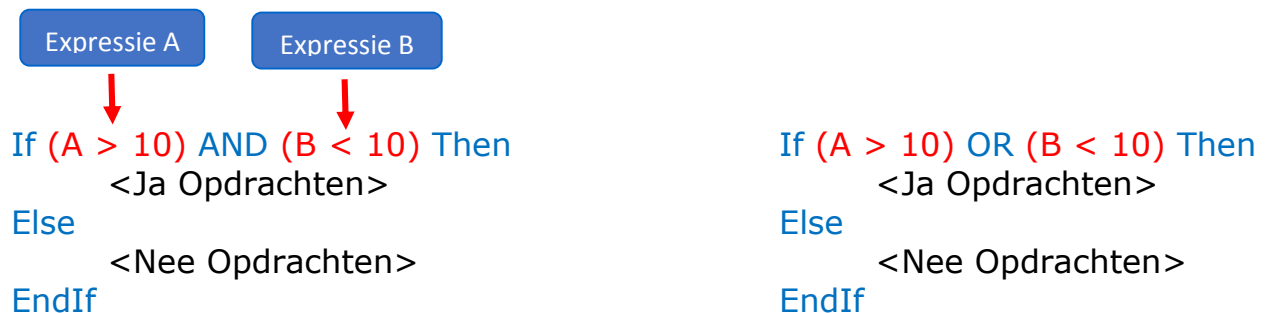
De expressie $A > 10$ wordt gecontroleerd als de waarde die de variabele A heeft groter is dan 10, dan worden de <Ja Opdrachten> uitgevoerd.

Mogelijke vergelijkingen:

- $A = B$, *gelijk aan*
- $A <> B$ of $A \neq B$, *ongelijk aan*
- $A > B$, *groter dan*
- $A < B$, *kleiner dan*
- $A \geq B$, *groter of gelijk aan*
- $A \leq B$, *kleiner of gelijk aan*

Soms moeten er meerdere expressies vergeleken worden, daarvoor zijn er de instructies **AND** en **OR**.

<Expressie A> **AND** <Expressie B> of <Expressie A> **OR** <Expressie B>



De volgende tabellen laten zien wanneer de <Ja Opdrachten> of de <Nee Opdrachten> worden uitgevoerd.

	Expressie A		
	AND	W	N
Expressie B	W	W	N
	N	N	N

	Expressie A		
	OR	W	N
Expressie B	W	W	W
	N	W	N

Bij de **AND** instructie moeten beide expressies *waar* zijn wil het resultaat van de totale expressie waar te laten zijn. Bij de **OR** instructie moeten beide expressies *niet waar* zijn om het resultaat niet waar te laten zijn.



Van PSD naar Small Basic

Opdracht 1

De PSD hieronder vraagt de gebruiker om drie getallen en berekend het gemiddelde van die getallen en drukt dat af op het beeldscherm.

Opdracht 1 PSD
SCHRIJF "Dit programma berekend het gemiddelde van drie in te voeren getallen."
LEES "Voer getal 1 in."; GETAL1
LEES "Voer getal 2 in."; GETAL2
LEES "Voer getal 3 in."; GETAL3
GEM := (GETAL1+GETAL2+GETAL3)/3
SCHRIJF "Het gemiddelde van de drie getallen is "; GEM

Schrijf de bron code in Small Basic voor de bovenstaande PSD en test of het programma werkt zoals het hoort.

Opdracht 2

De PSD hieronder berekend met behulp van de stelling van Pythagoras de schuine zijde (C) als de zijden A en B door de gebruiker van het programma worden ingevuld. En de waarde van C afdrukt op het beeldscherm.

Opdracht 2 PSD
SCHRIJF "Dit programma berekend de schuine zijde van een drie hoek met de stelling van Pthagoras."
LEES "Voer een waarde in voor zijde A."; ZijdeA
LEES "Voer een waarde in voor zijde B."; ZijdeB
ZijdeC := (ZijdeA ² + ZijdeB ²) ^{0.5}
SCHRIJF "De lengte van de schuine zijde C is."; ZijdeC



Van PSD naar Small Basic

Schrijf de bron code in Small Basic voor de bovenstaande PSD en test of het programma werkt zoals het hoort.

Opdracht 3

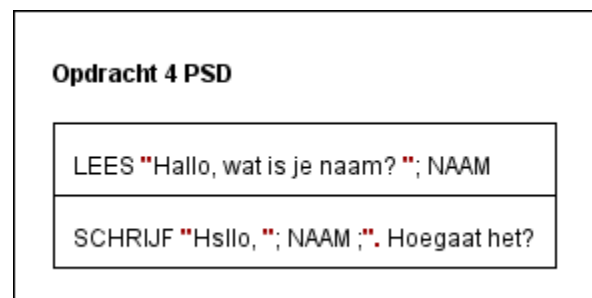
De PSD hieronder berekend de eerste 10 waarden van de tafel van 6 en drukt deze af op het beeld scherm.



Schrijf de bron code in Small Basic voor de bovenstaande PSD en test of het programma werkt zoals het hoort.

Opdracht 4

Schrijf de bron code in Small Basic voor de PSD hiernaast en test of het programma werkt zoals het hoort.

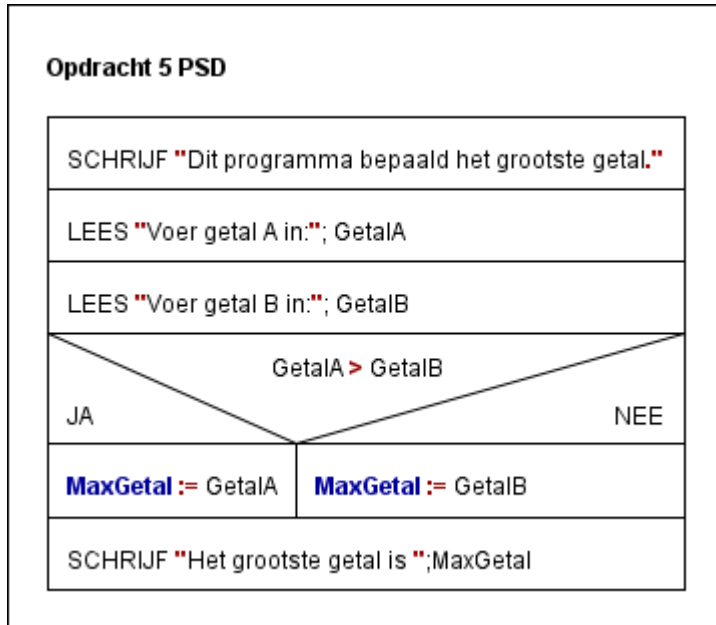




Van PSD naar Small Basic

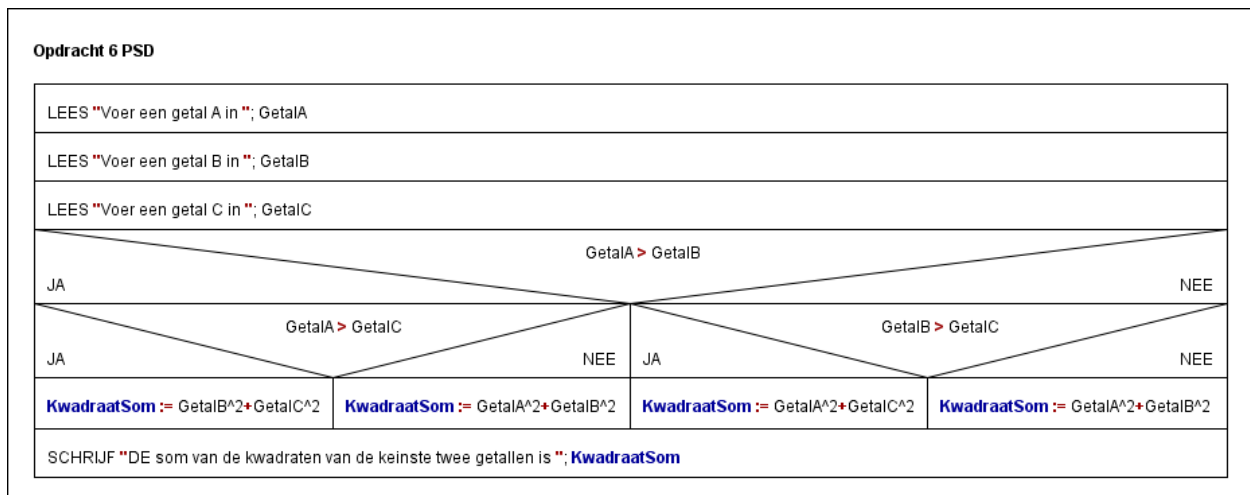
Opdracht 5

De PSD hieronder laat de gebruiker van het programma twee getallen invoeren en drukt het grootste van de twee getallen af op het scherm.



Schrijf de bron code in Small Basic voor de bovenstaande PSD en test of het programma werkt zoals het hoort.

Opdracht 6

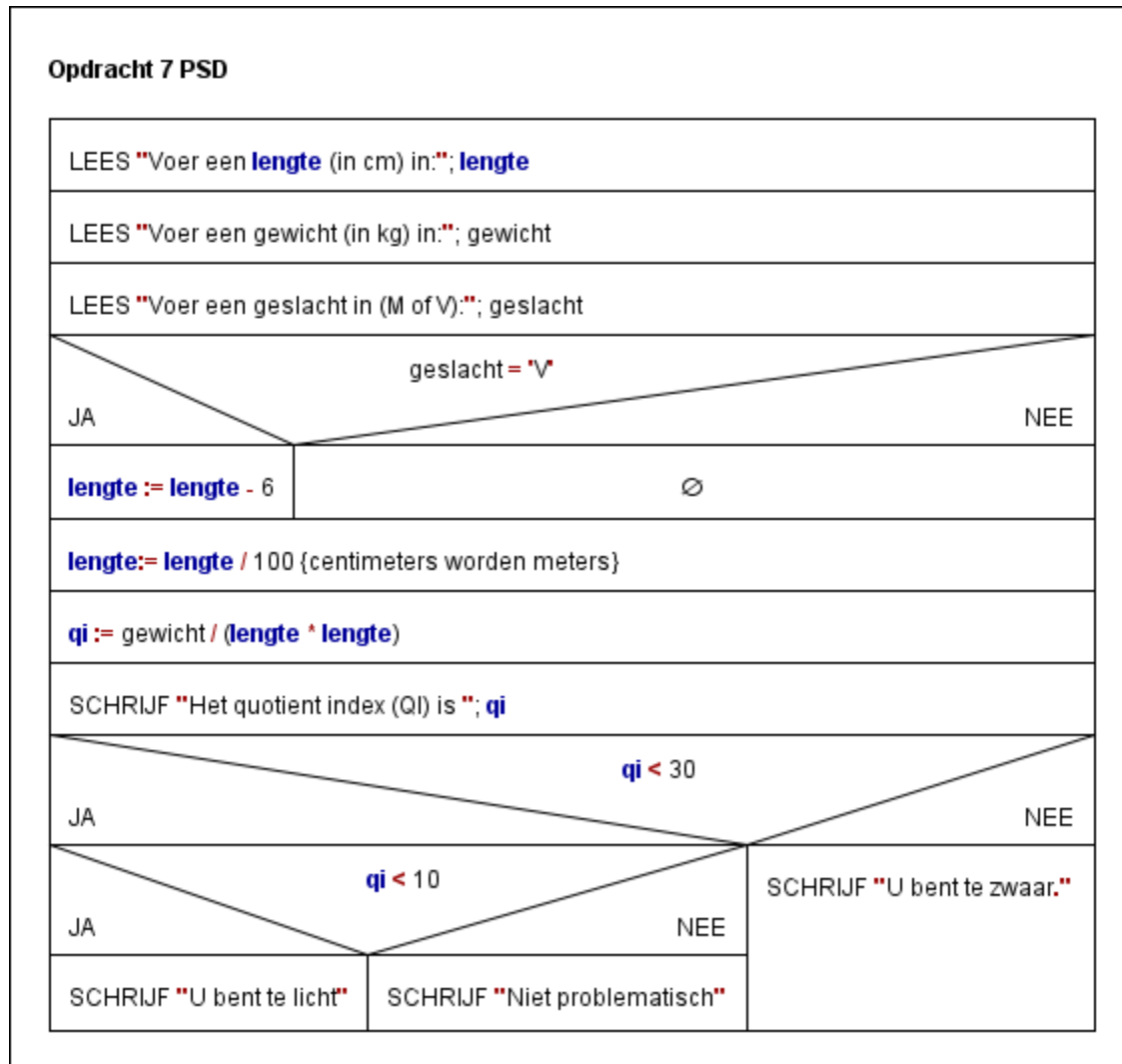


Schrijf de bron code in Small Basic voor de bovenstaande PSD en test of het programma werkt zoals het hoort.



Van PSD naar Small Basic

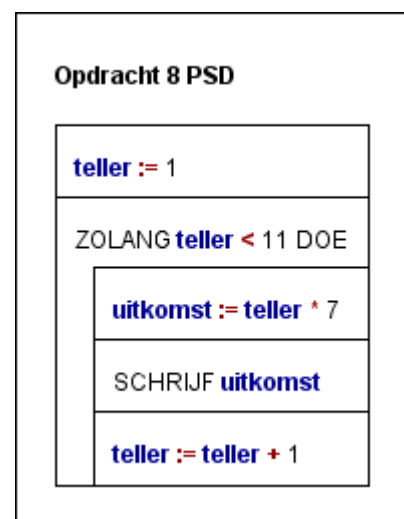
Opdracht 7



Schrijf de bron code in Small Basic voor de bovenstaande PSD en test of het programma werkt zoals het hoort.

Opdracht 8

Schrijf de bron code in Small Basic voor de PSD hiernaast en test of het programma werkt zoals het hoort.

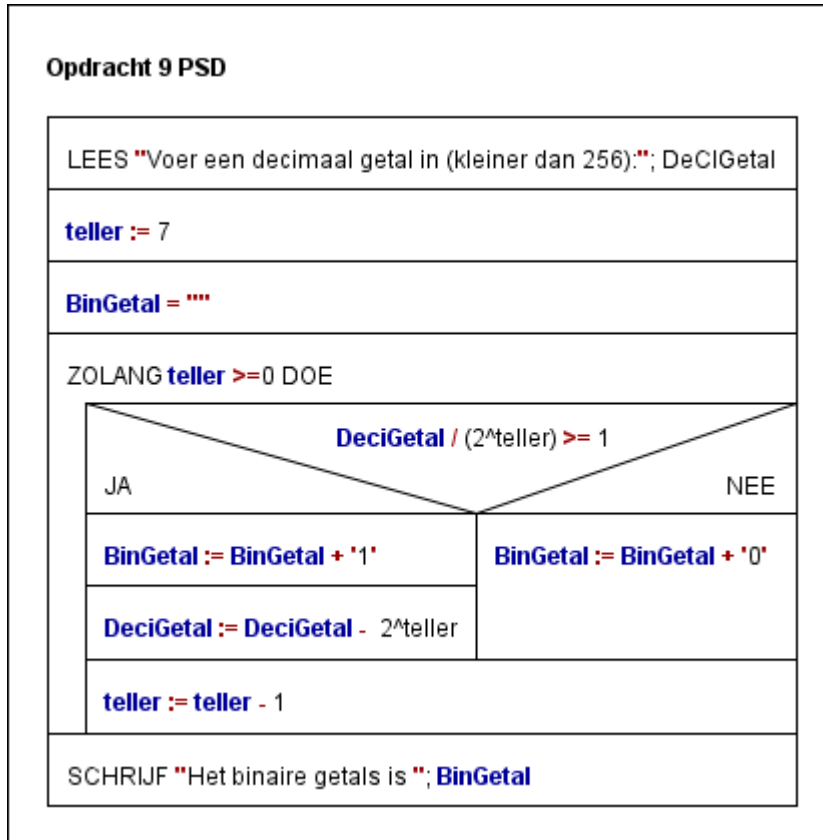




Van PSD naar Small Basic

Opdracht 9

De PSD hieronder zet een decimaal getal, kleiner dan 256, om in een binair getal van 8 bits.

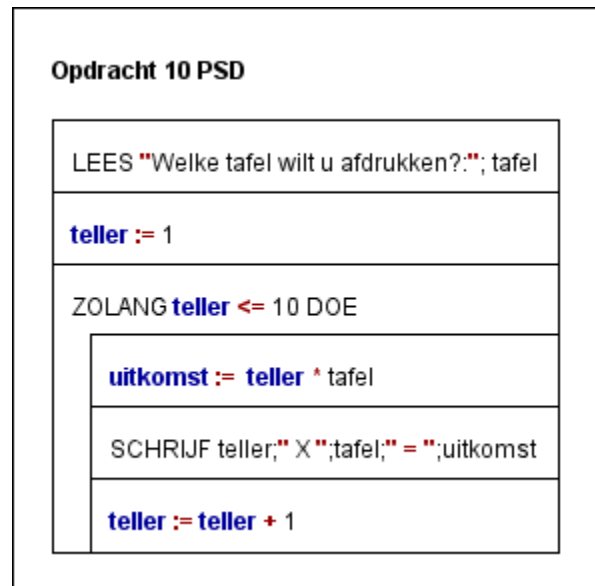


Schrijf de bron code in Small Basic voor de bovenstaande PSD en test of het programma werkt zoals het hoort.

Opdracht 10

De PSD hiernaast drukt de eerste 10 uitkomsten van een door de gebruiker ingevoerde tafel op het beeldscherm af.

Schrijf de bron code in Small Basic voor de bovenstaande PSD en test of het programma werkt zoals het hoort.





Van PSD naar Small Basic

Opdracht 11

Maak een PSD en schrijf het programma in Small Basic die de temperatuur van 0 graden Celsius tot 100 graden Celsius omrekent naar Fahrenheit en met tussen stappen van 5 graden Celsius een tabel met beide waarden afdrukt op het scherm.

De formule voor het omrekenen is al volgt:
Fahrenheit= Celsius x 9 / 5 + 32

Celsius	Fahrenheit
0	32
5	...

Opdracht 12

Het algoritme van Euclides. In de klassieke wiskunde speelde het begrip deelbaarheid een belangrijke rol. De Griekse wiskundige Euclides (325 – 265 voor Chr.) schreef een algoritme voor het bepalen van de grootste gemeenschappelijke deler (GGD) van twee getallen, dat is het grootste gehele getal waardoor je beide getallen kunt delen.

Voorbeeld:

De delers van 24 zijn 1, 2, 3, 4, 6, 8, 12 en 24

De delers van 15 zijn 1, 2, 3, 5 en 15

De grootste gemene deler van 24 en 15 is dus 3

Opmerking:

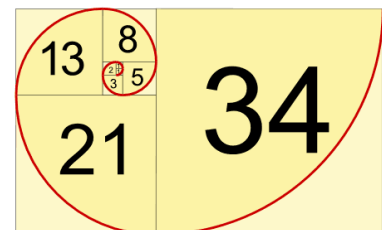
Om de rest waarde te berekenen gebruik % in de PSD en `Math.Remainder` in Small Basic.

PSD	Small Basic
Restwaarde = A % B	Restwaarde = <code>Math.Remainder(A, B)</code>

- A. Maak een PSD voor dit algoritme.
- B. Schrijf de bron code in Small Basic.

Opdracht 13

De **rij van Fibonacci** is genoemd naar Leonardo van Pisa, bijgenaamd Fibonacci.





Van PSD naar Small Basic

De rij begint met 0 en 1 en vervolgens is elk volgende element van de rij steeds de som van de twee voorgaande elementen.

De eerste elementen van de rij [1] zijn dan als volgt:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, ...

- A. Maak een PSD waarbij de gebruiker van het programma aangeeft hoeveel elementen van de reeks door het programma afgedrukt moeten worden op het beeldscherm. Voert de gebruiker het getal -1 in dan moet het programma beëindigen.
- B. Schrijf de bron code in Small Basic en test het programma.

Opdracht 14

Maak de PSD en schrijf de bron code in Small Basic voor een programma dat uit een onbepaalde reeks willekeurig door de gebruiker ingevoerde getallen het grootste getal afdruckt.

De gebruiker voert getallen in en wanneer het cijfer nul wordt ingevoerd stopt het programma en drukt het grootste getal af op het beeldscherm.

Opdracht 15

```
1 TextWindow.Write("Welke tafel moet ik afdrucken? ")
2 Tafel = TextWindow.ReadNumber()
3 Teller = 1
4 While Teller <= 10
5     Uitkomst = Teller * Tafel
6     TextWindow.WriteLine(Teller + " X " + Tafel + " = " + Uitkomst)
7     Teller = Teller + 1
8 EndWhile
9
```

- A. Maak de PSD voor de bovenstaande bron code.
- B. Schrijf in eigenwoorden op wat dit programma doet.