

# Kunnen computers programmeren?

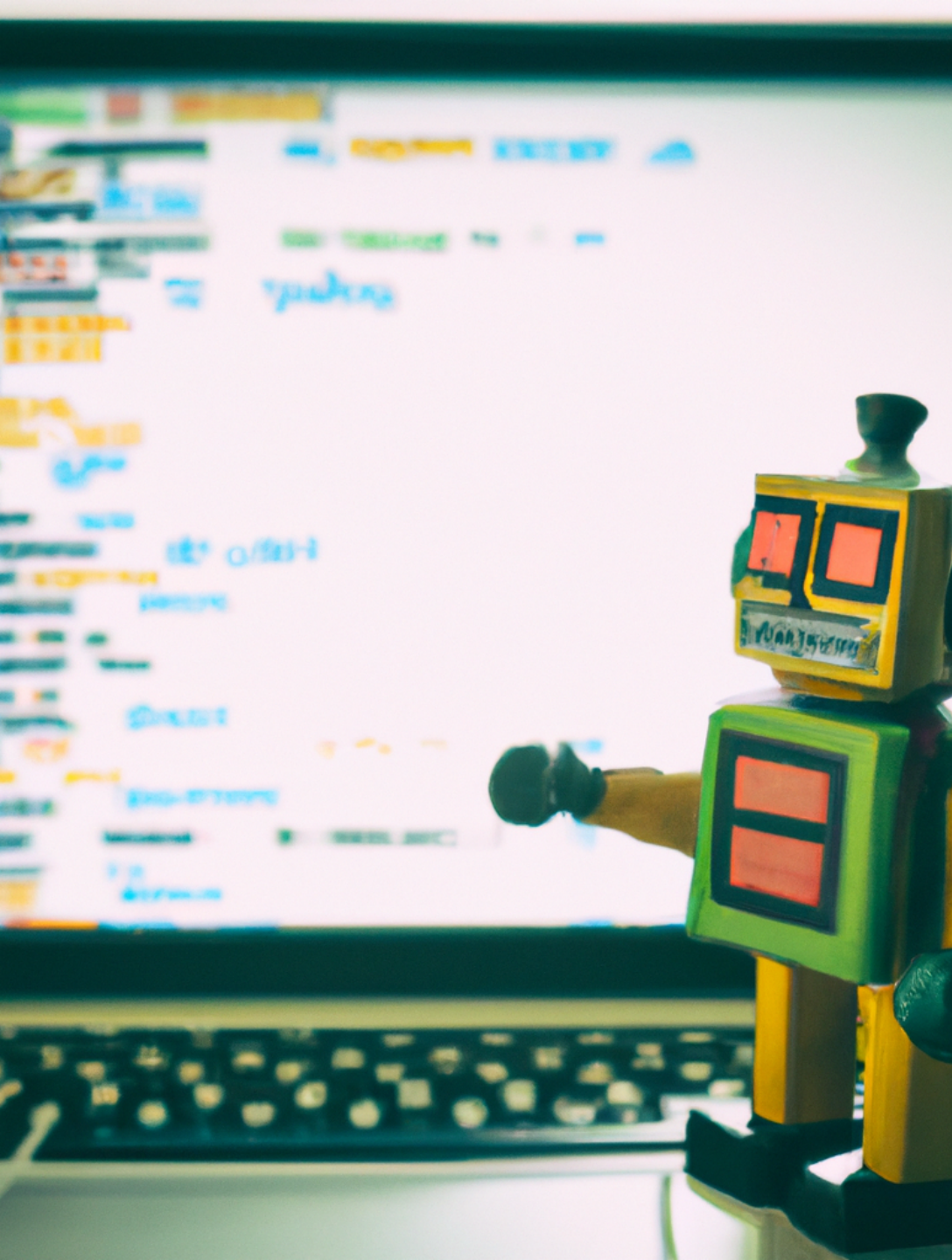
Prof. dr. Frank Neven  
frank.neven@uhasselt.be

Codefestival - EU Codeweek -  
09/10/2023

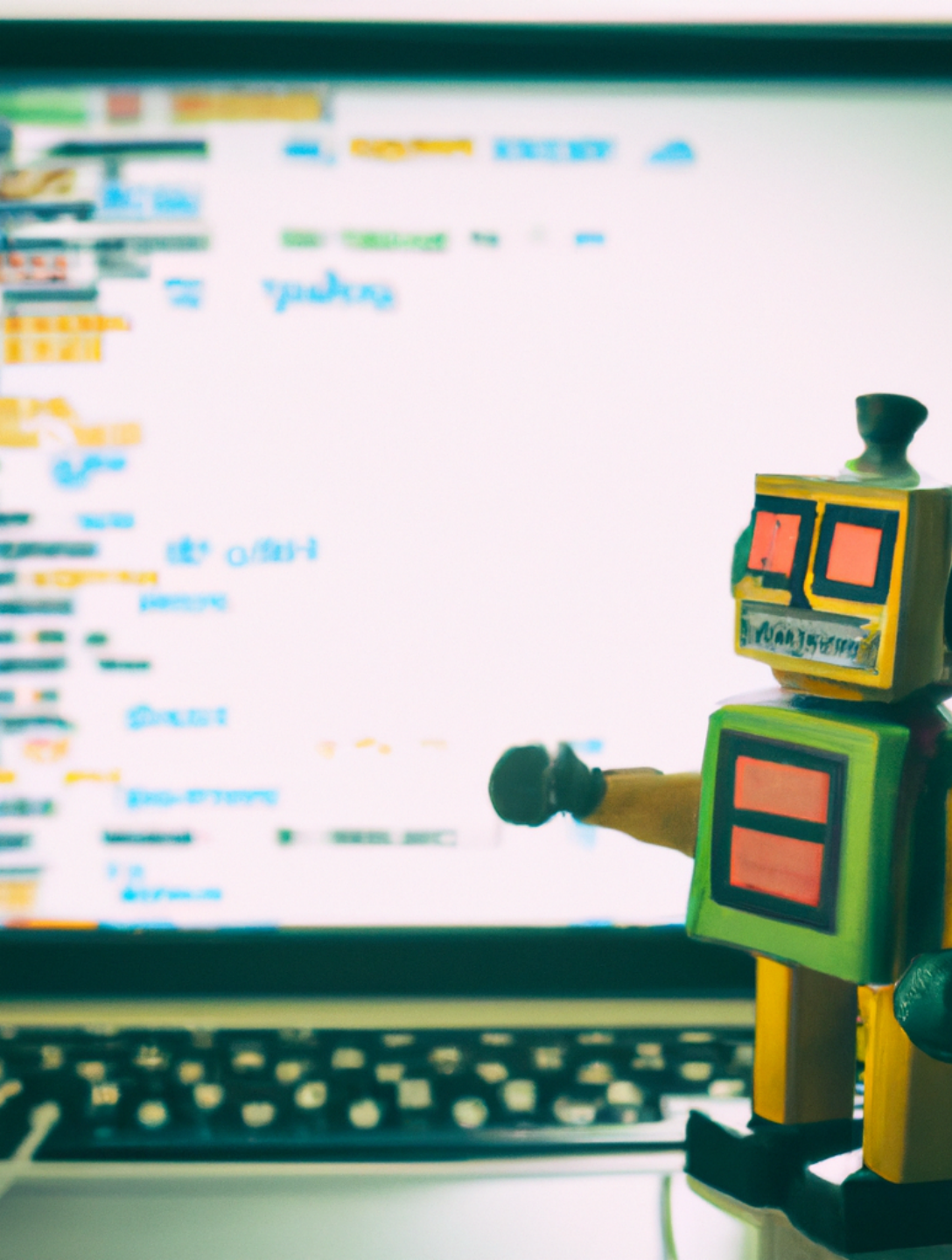


[WWW.UHASSELT.BE/DSI](http://WWW.UHASSELT.BE/DSI)





# Kunnen computers programmeren?



# Kunnen computers programmeren?

- ✓ Programmeren
- ✓ Algoritmen en datastructuren
- ✓ Informaticawetenschappen
- ✓ Artificiële Intelligentie

# Generatieve AI



# Generatieve AI



# Generatieve AI

Een schilderij  
in de stijl van Van Gogh  
van een  
Amerikaanse footballspeler

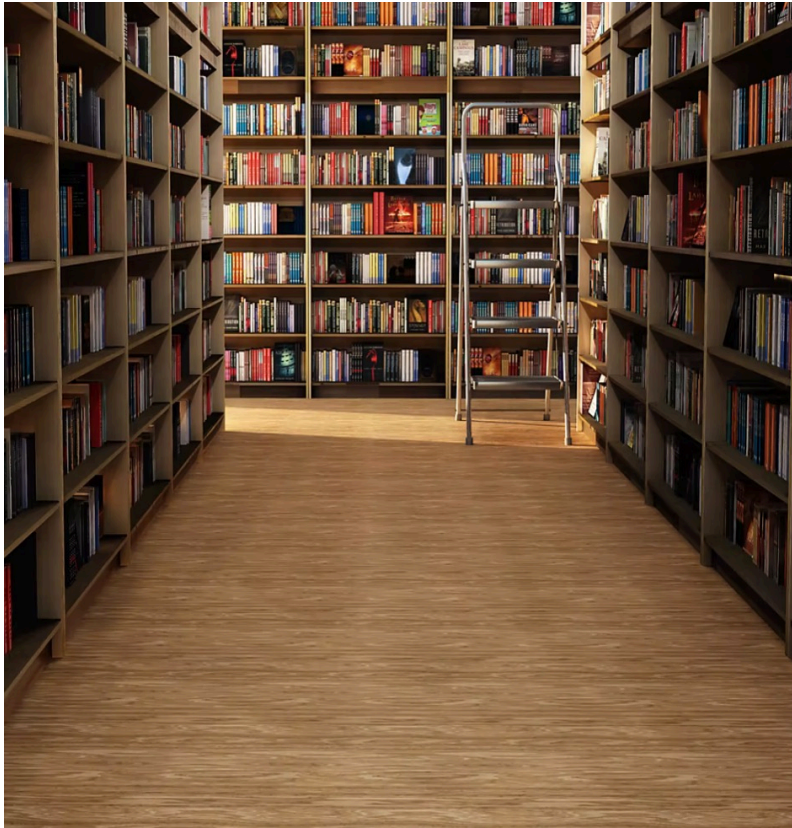


# Generatieve AI

Een foto  
van een teddybeer  
op een skateboard  
in Times Square

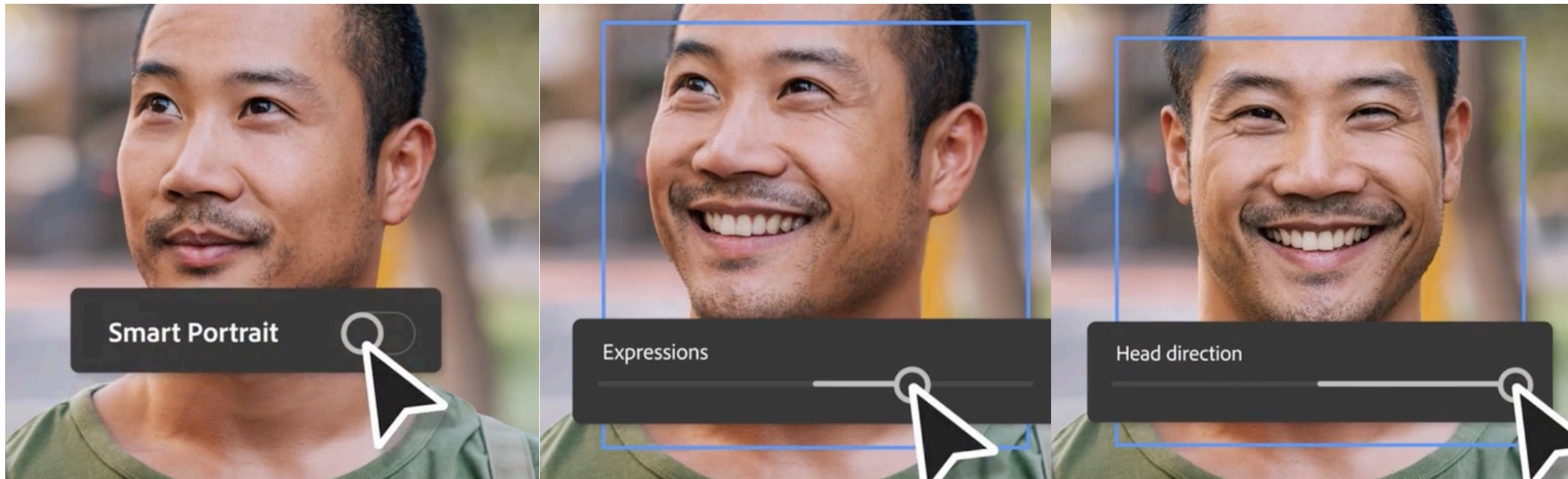


# Generatieve AI





# Generatieve AI



# Generatieve AI




chatGPT


# Generatieve AI



chatGPT

**F** Stel jezelf voor in 1 zin



 Ik ben ChatGPT, een door OpenAI ontwikkelde virtuele assistent, hier om je te helpen met informatie, advies en gesprekken over een breed scala aan onderwerpen.

# Generatieve AI



chatGPT

F

Schrijf een limerick over het plezier van programmeren



Een coder, heel blij met zijn scherm,  
Vond coderen vast, niet echt een term.  
Met een lach, heel sereen,  
Schreef hij code, heel clean,  
En de bugs overwon hij, standvastig en ferm.

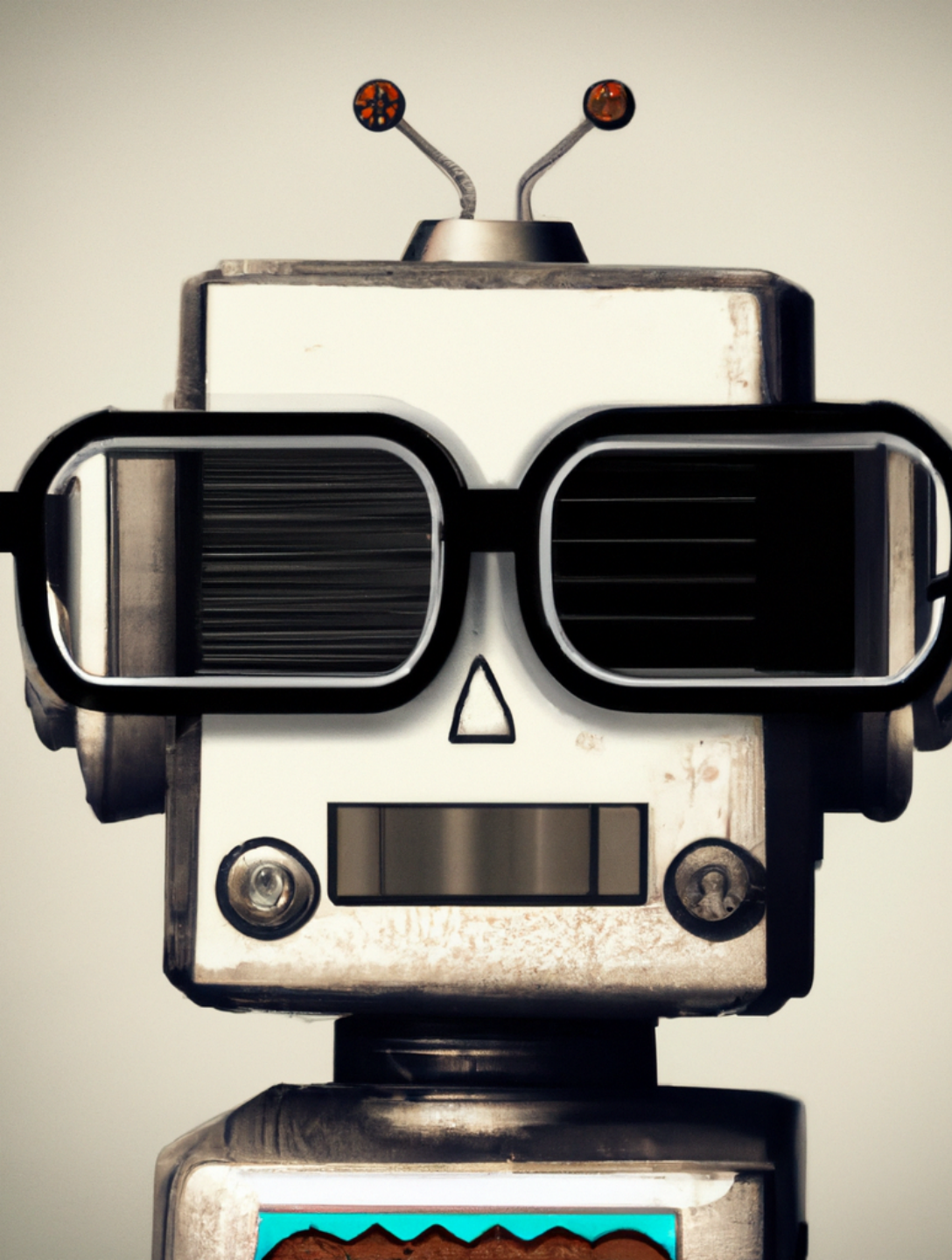
# Generatieve AI

F

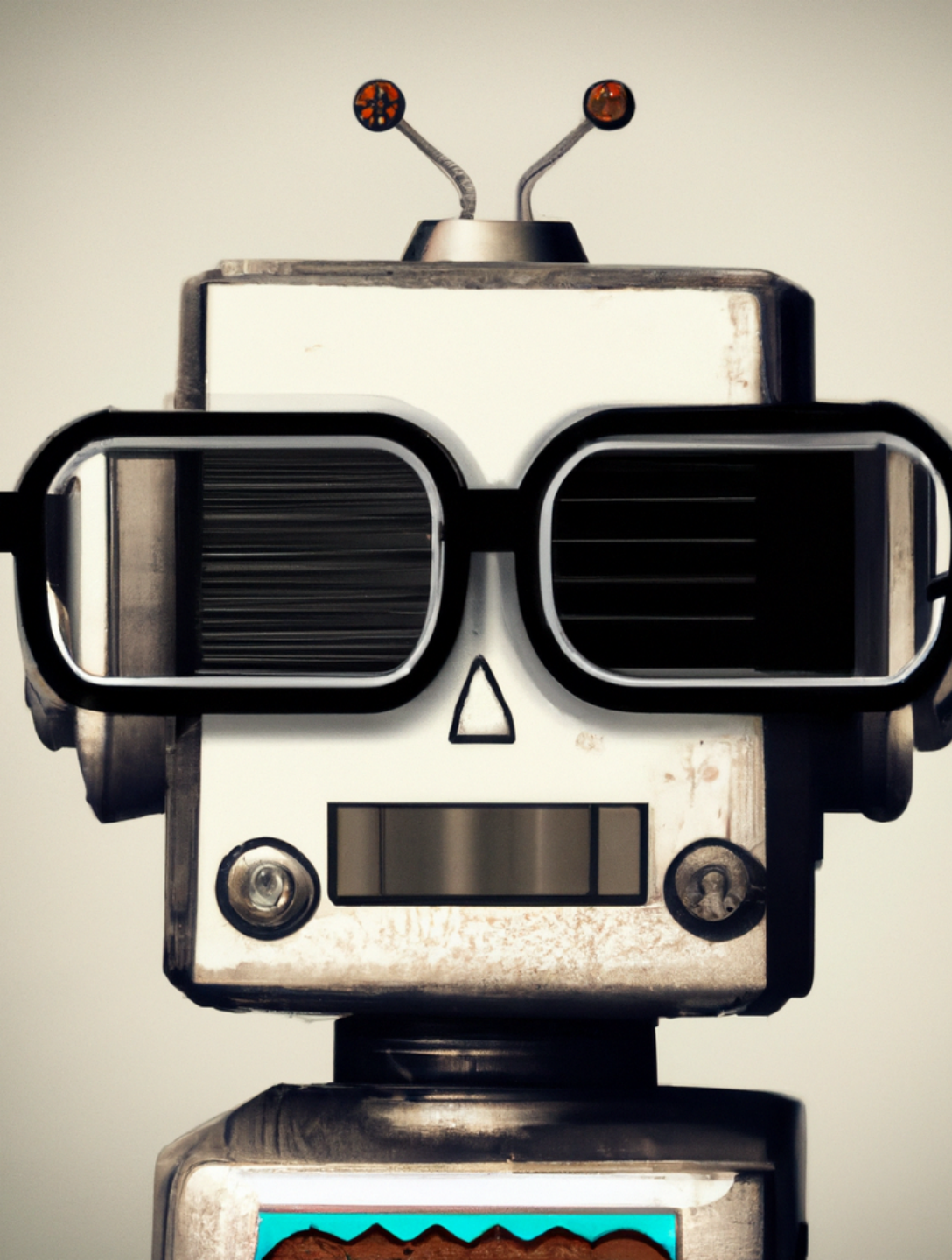
Schrijf een functie ggd die de grootste gemene deler van twee getallen berekent

```
def ggd(a, b):  
    while b != 0:  
        a, b = b, a % b  
    return a  
  
# Voorbeeldgebruik:  
x = 48  
y = 18  
print(f"De grootste gemene deler van {x} en {y} is {ggd(x, y)}")
```

# Generatieve AI democratiseert technologie



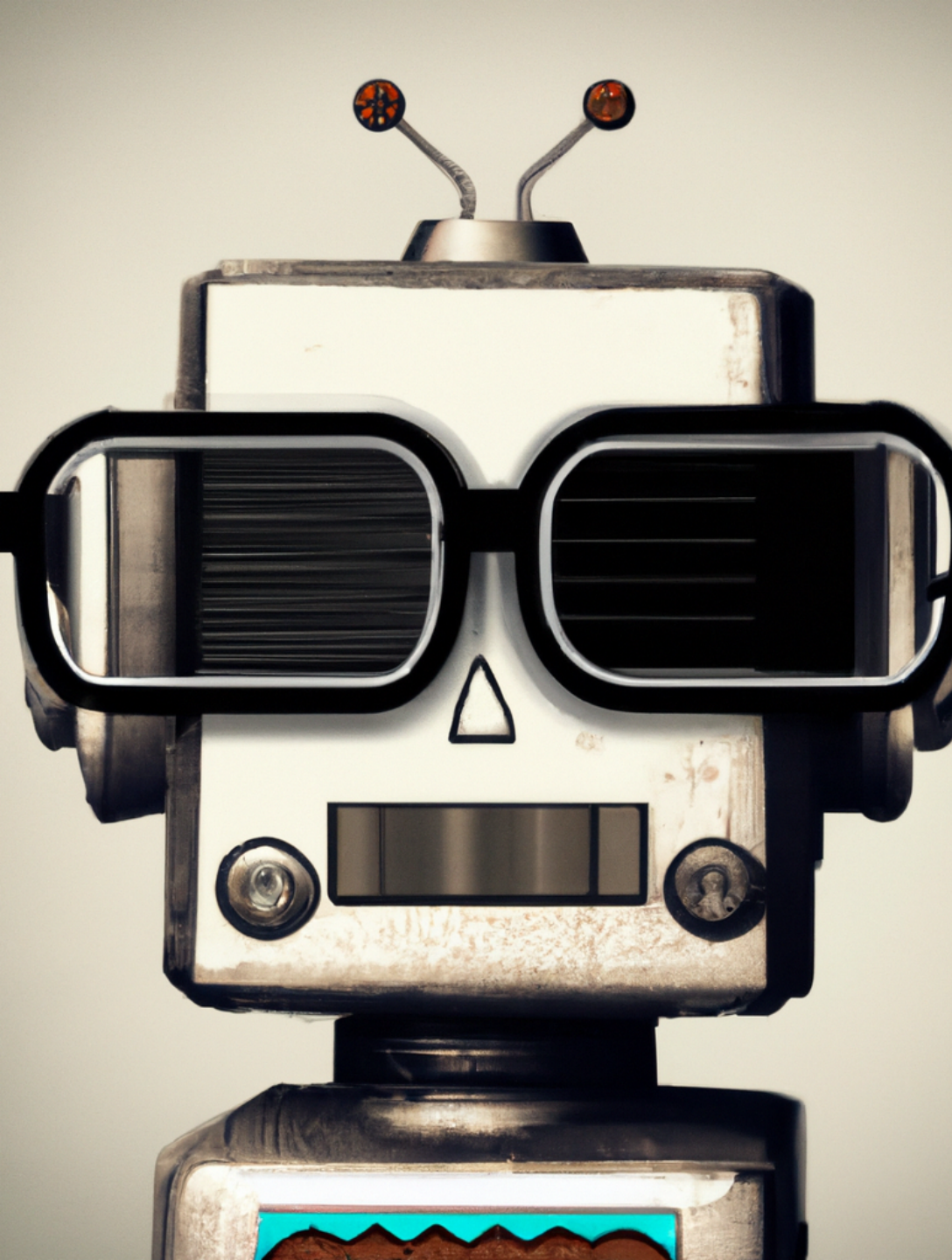
**Computers kunnen code genereren...**



**Computers kunnen code genereren...**

**...maar kunnen ze ook programmeren?**





**Computers kunnen code genereren...**

**...maar kunnen ze ook programmeren?**

**Programmeren bekijken door de  
bril van informaticawetenschappen**

# Informaticawetenschappen

# Informaticawetenschappen

- De wetenschap onderliggend aan de digitale wereld

# Informaticawetenschappen

- De wetenschap onderliggend aan de digitale wereld
- Nieuw vak in de 3de graad van het secundair onderwijs (voor een aantal richtingen)

# Informaticawetenschappen

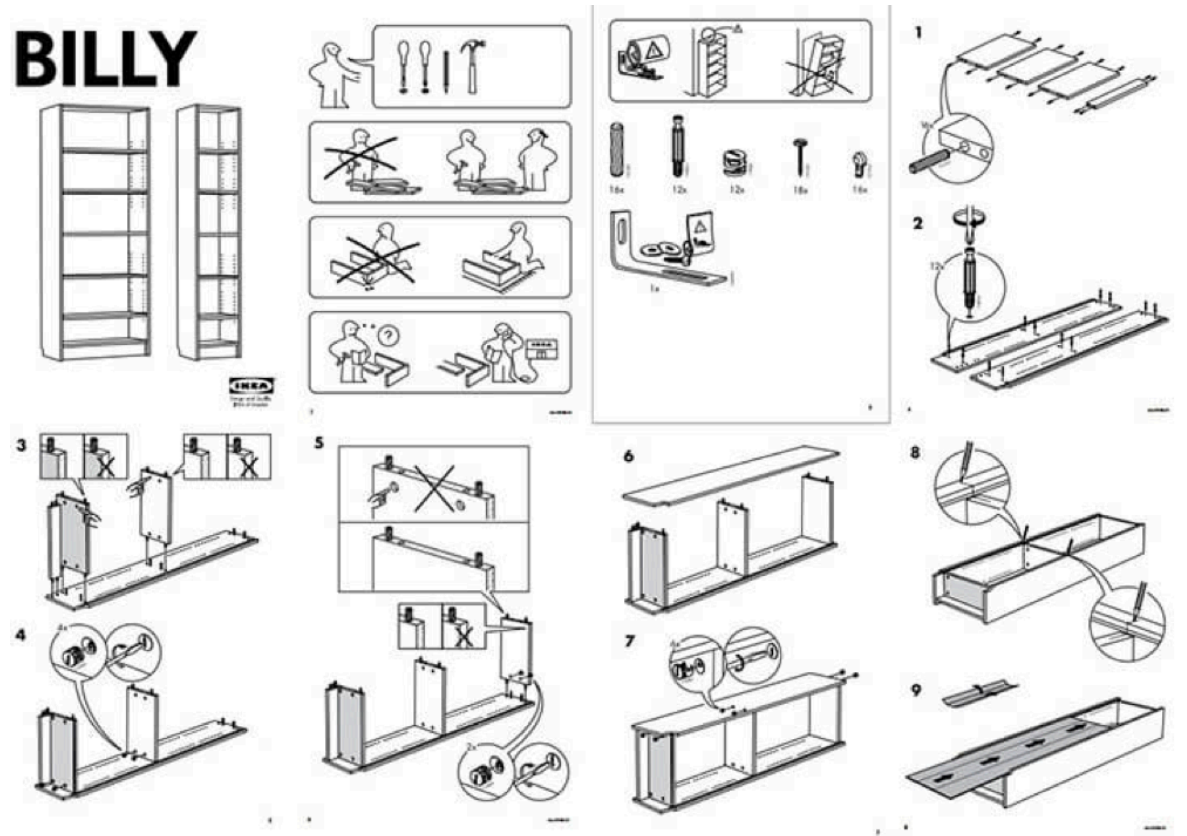
- De wetenschap onderliggend aan de digitale wereld
- Nieuw vak in de 3de graad van het secundair onderwijs (voor een aantal richtingen)
- Focus op **onderliggende principes** met lange levensduur:
  - *Algoritmen en programmeren*
  - *Bouwstenen van digitale systemen*
    - *Informatie- en databeheer*
    - *Datacommunicatie, computer- en netwerkachitectuur*
    - *Softwareontwikkeling*

# Algoritme

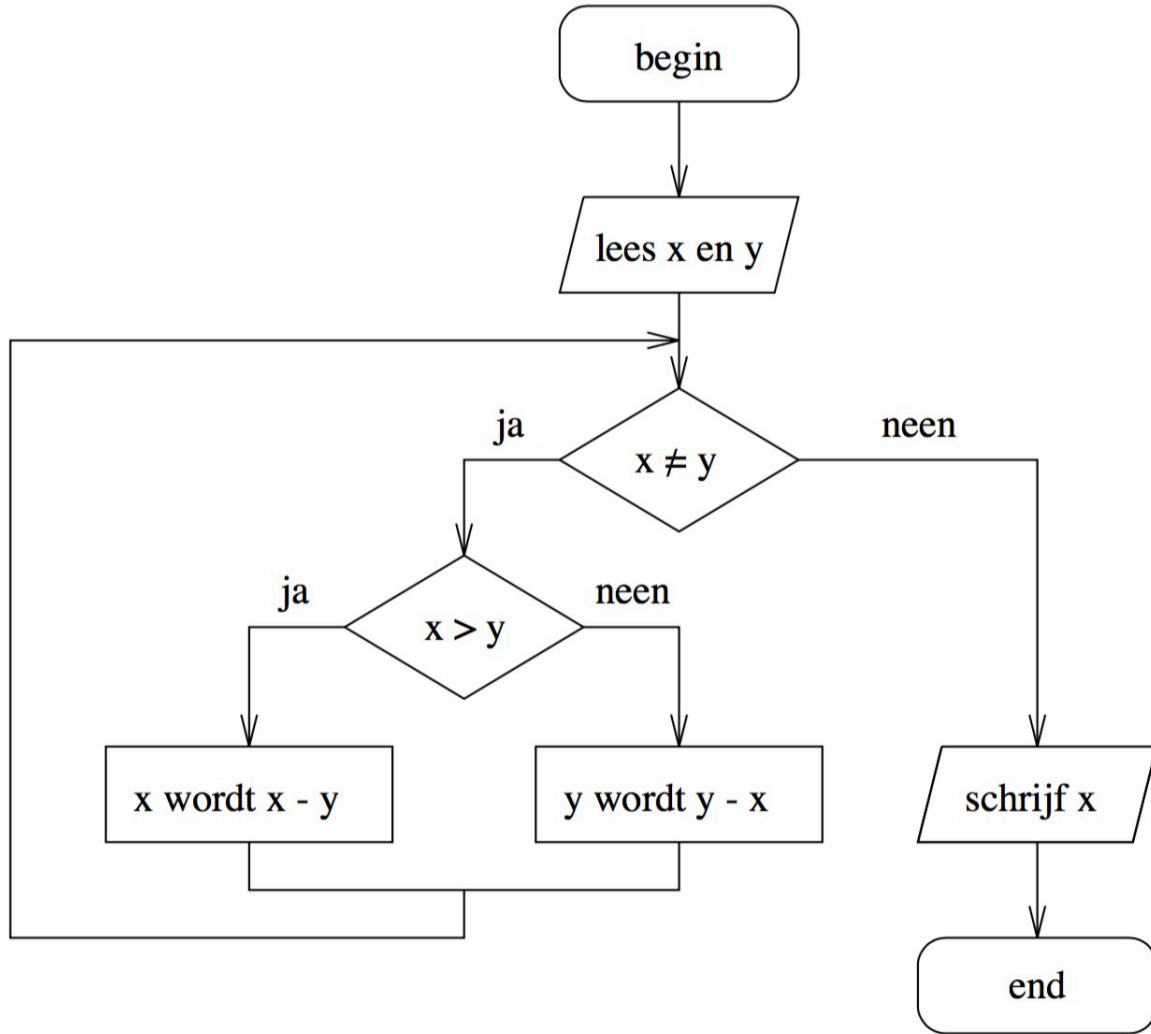
Een algoritme is een reeks instructies of regels die worden gevolgd om een taak of probleem op te lossen.

# Algoritme

Een algoritme is een reeks instructies of regels die worden gevolgd om een taak of probleem op te lossen.

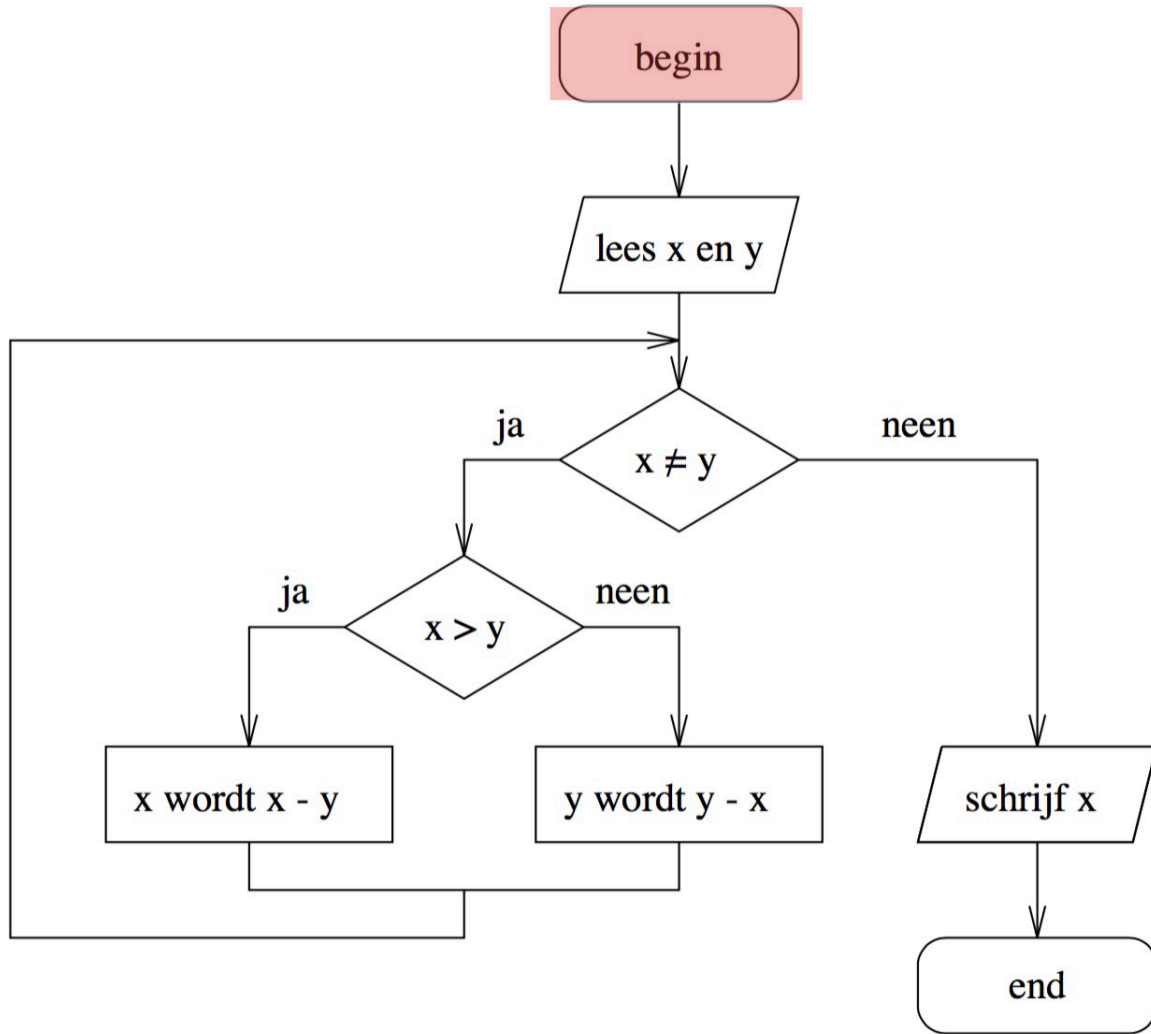


# Berekenen van ggd: Algoritme van Euclides

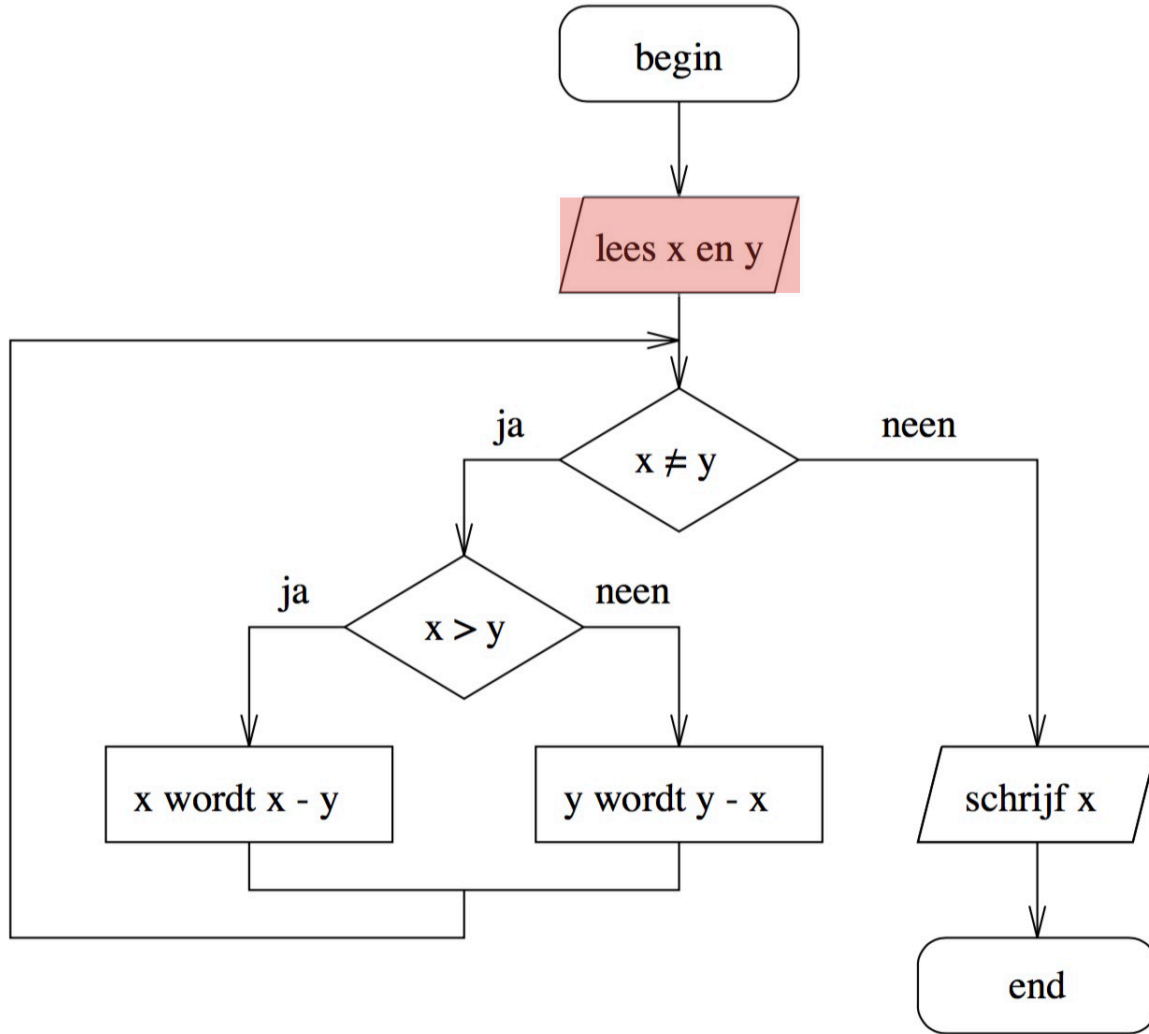




# Berekenen van ggd: Algoritme van Euclides

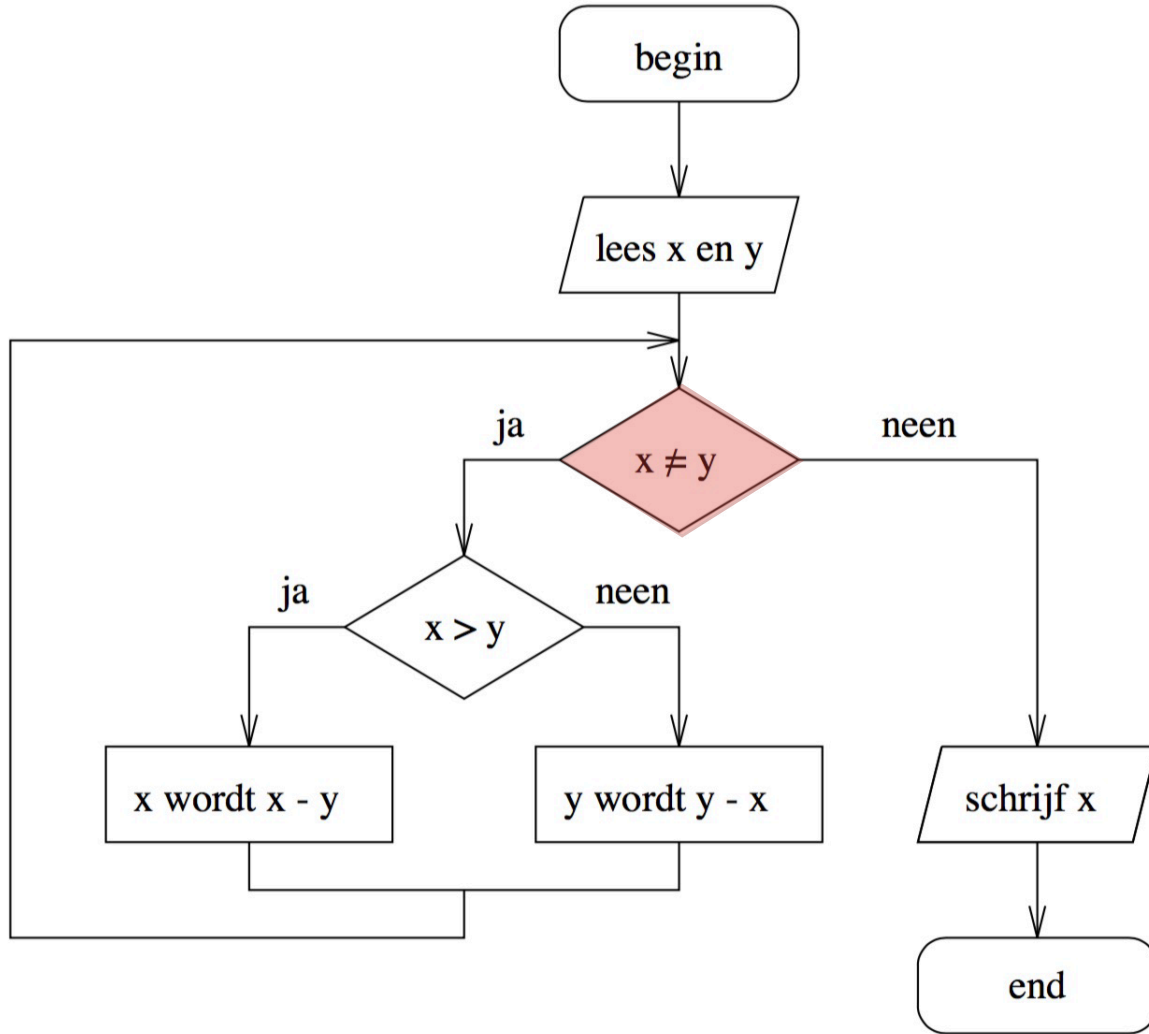


# Berekenen van ggd: Algoritme van Euclides



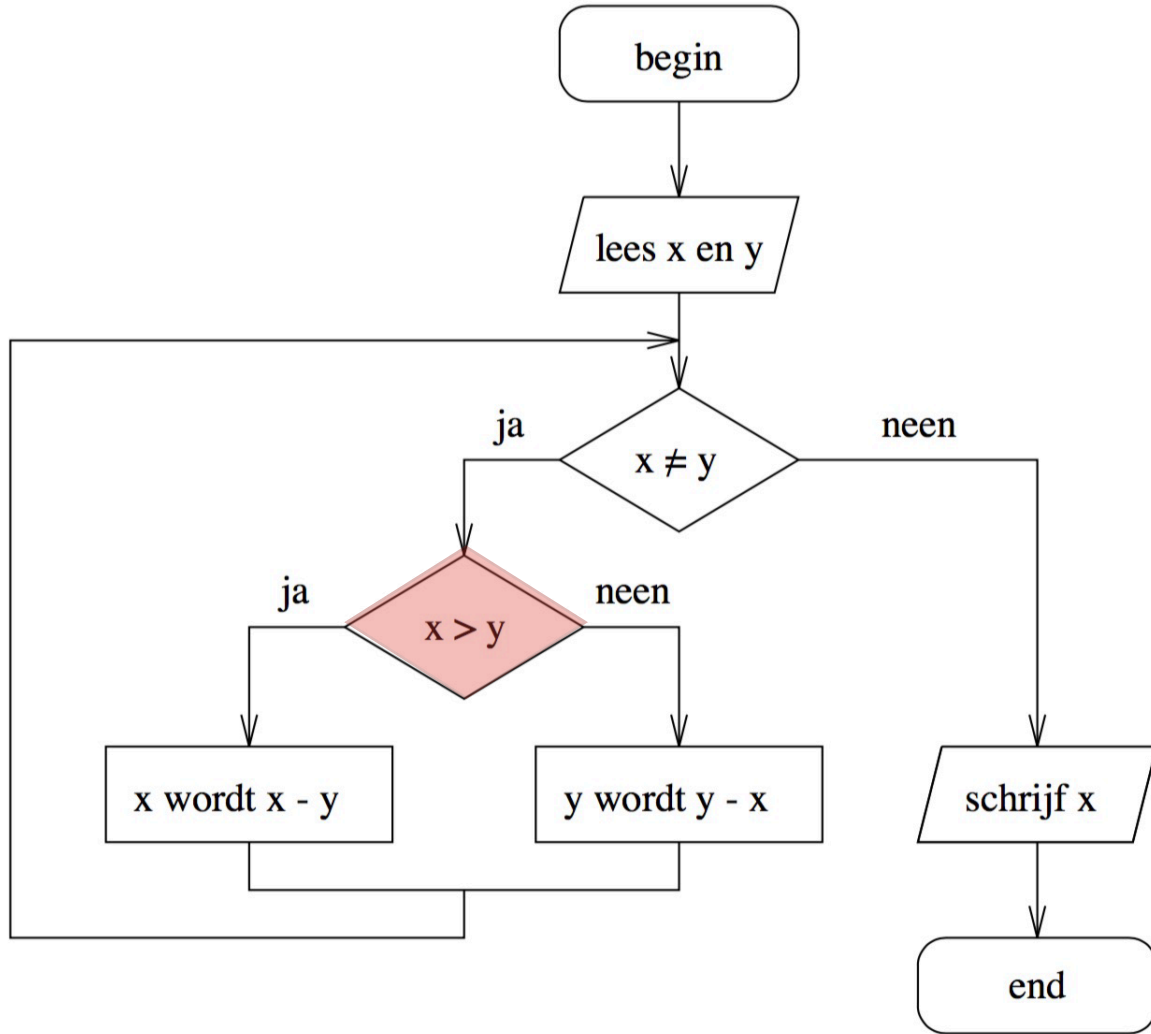
x	y
12	8

# Berekenen van ggd: Algoritme van Euclides



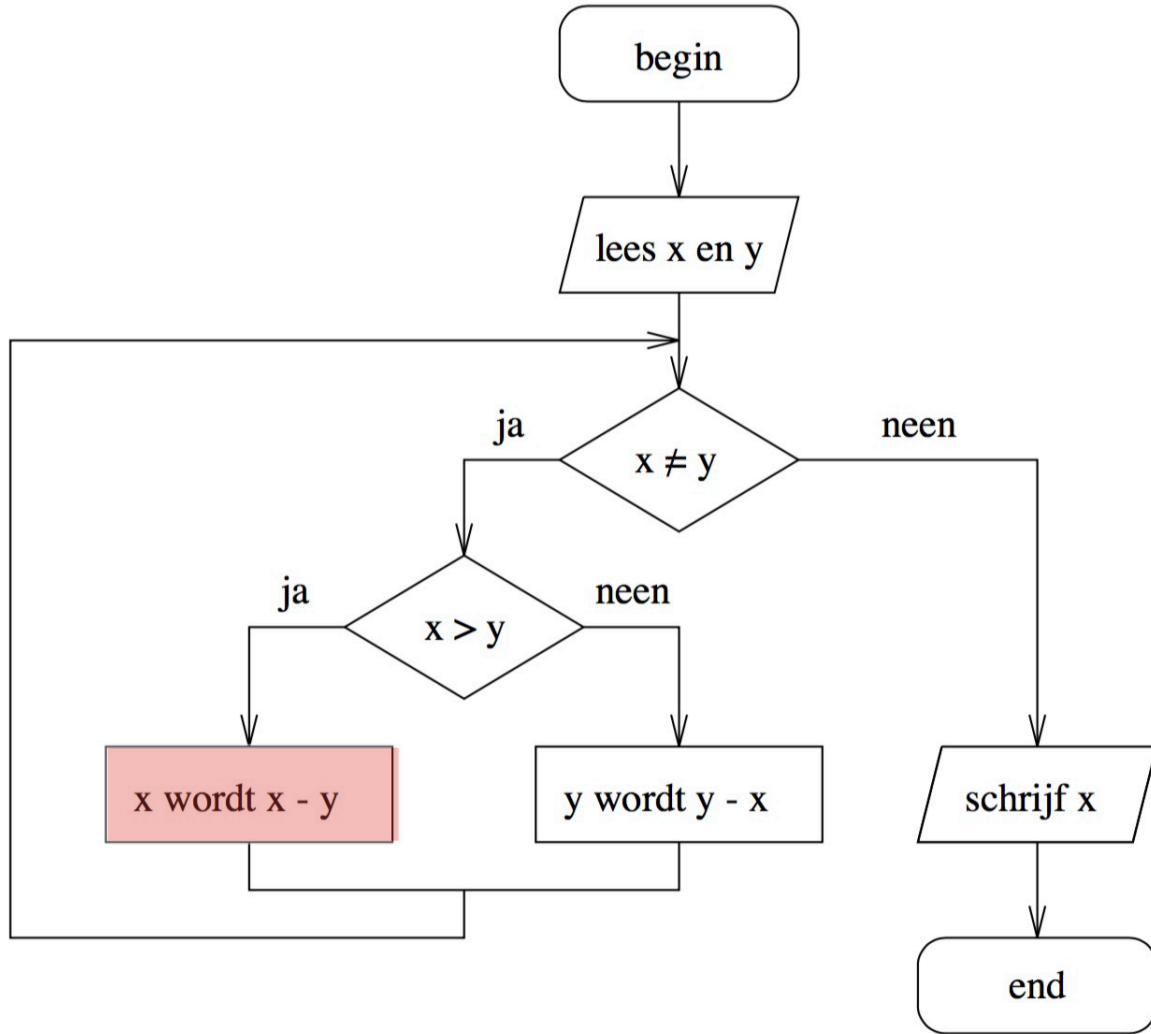
x	y
12	8

# Berekenen van ggd: Algoritme van Euclides



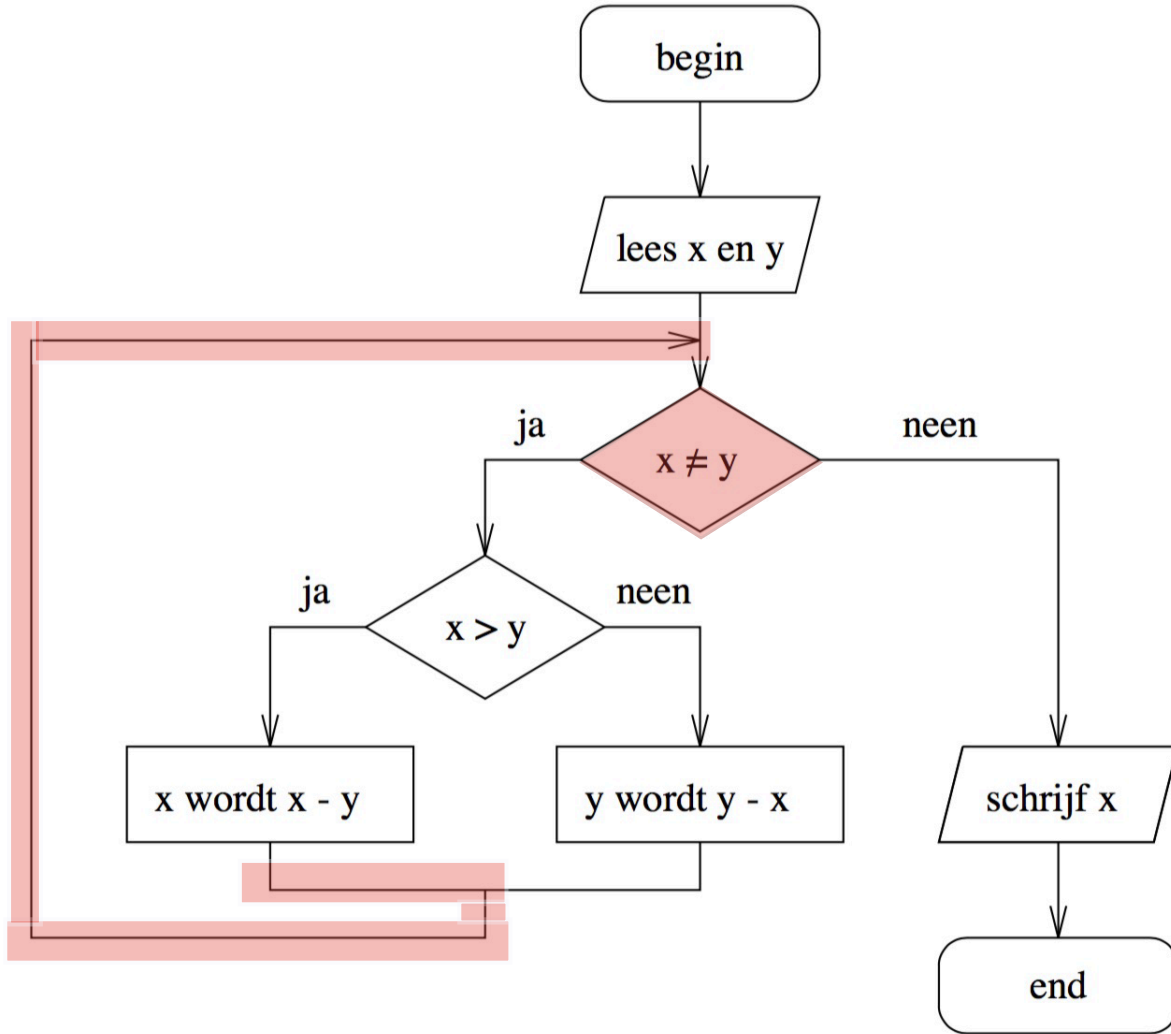
x	y
12	8

# Berekenen van ggd: Algoritme van Euclides



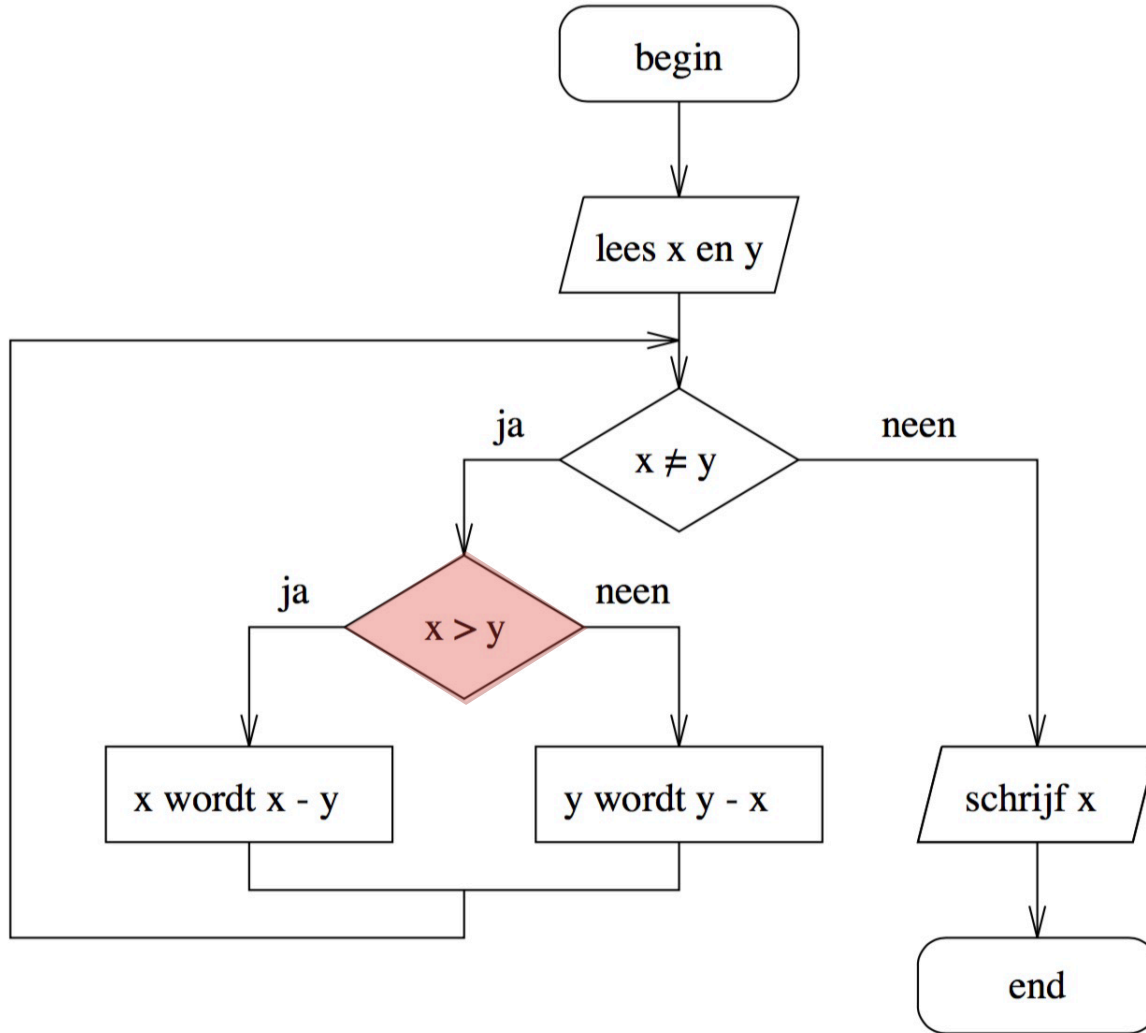
x	y
12	8
4	8

# Berekenen van ggd: Algoritme van Euclides



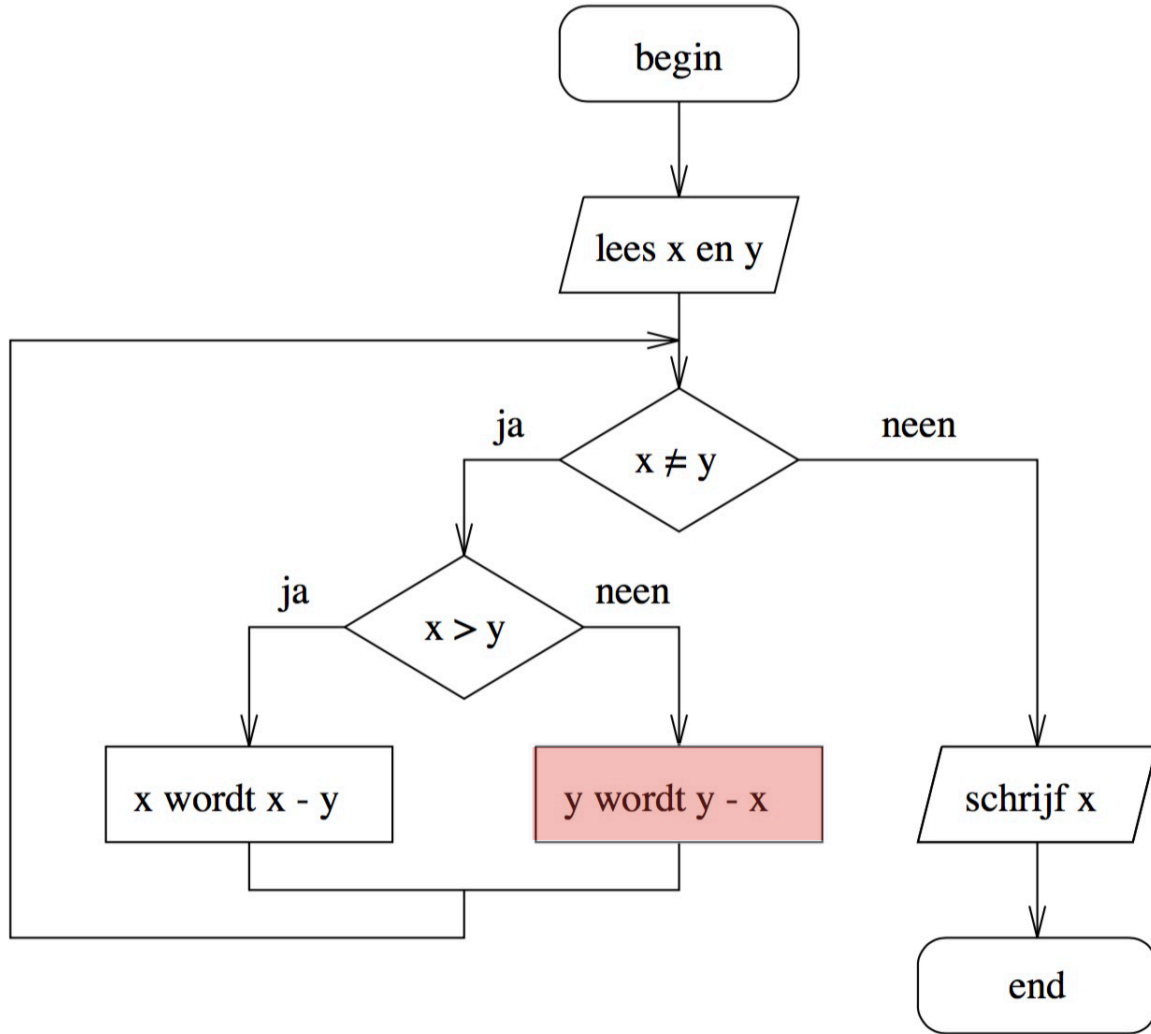
x	y
12	8
4	8

# Berekenen van ggd: Algoritme van Euclides



x	y
12	8
4	8

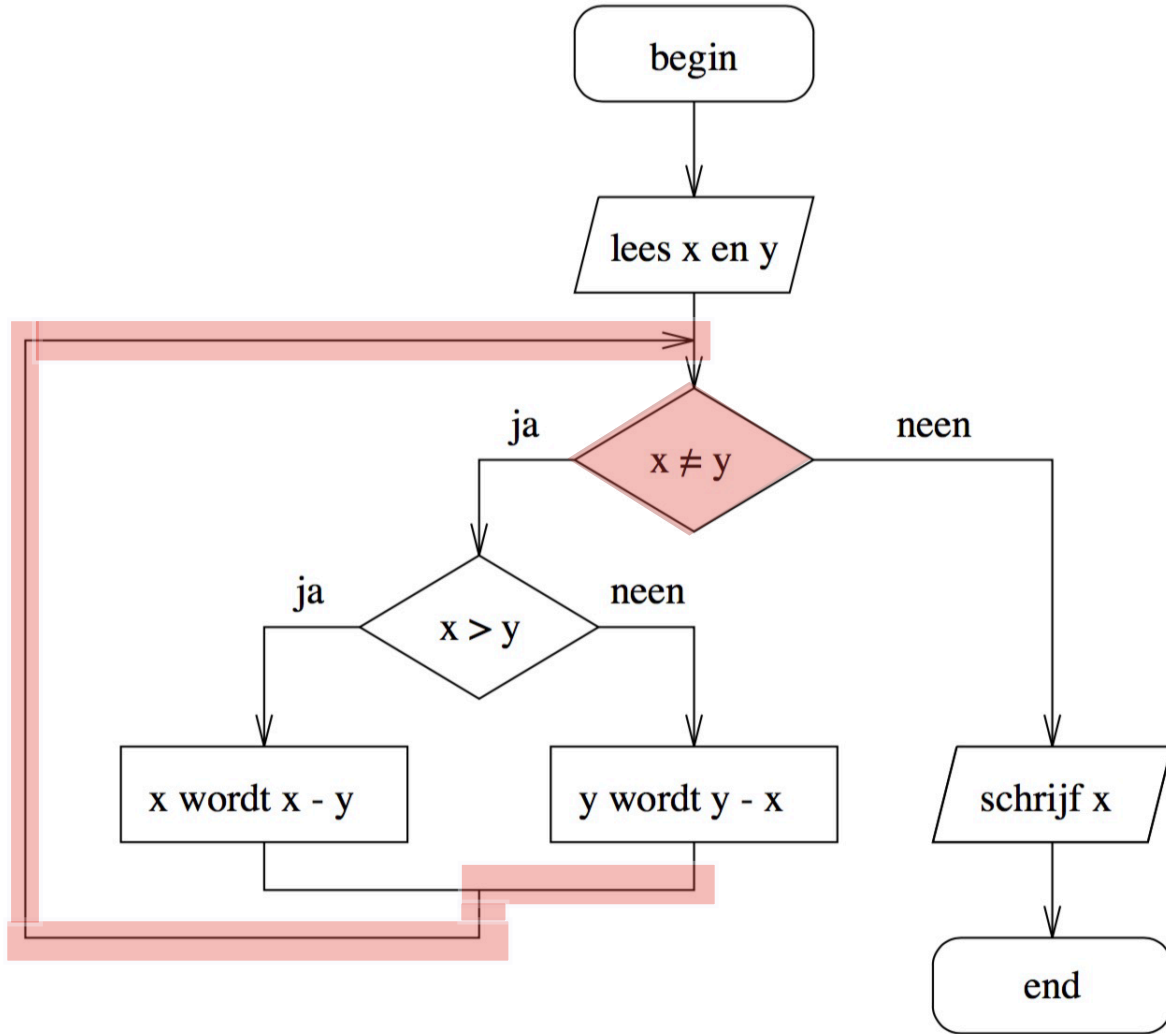
# Berekenen van ggd: Algoritme van Euclides



x	y
12	8
4	8
4	4

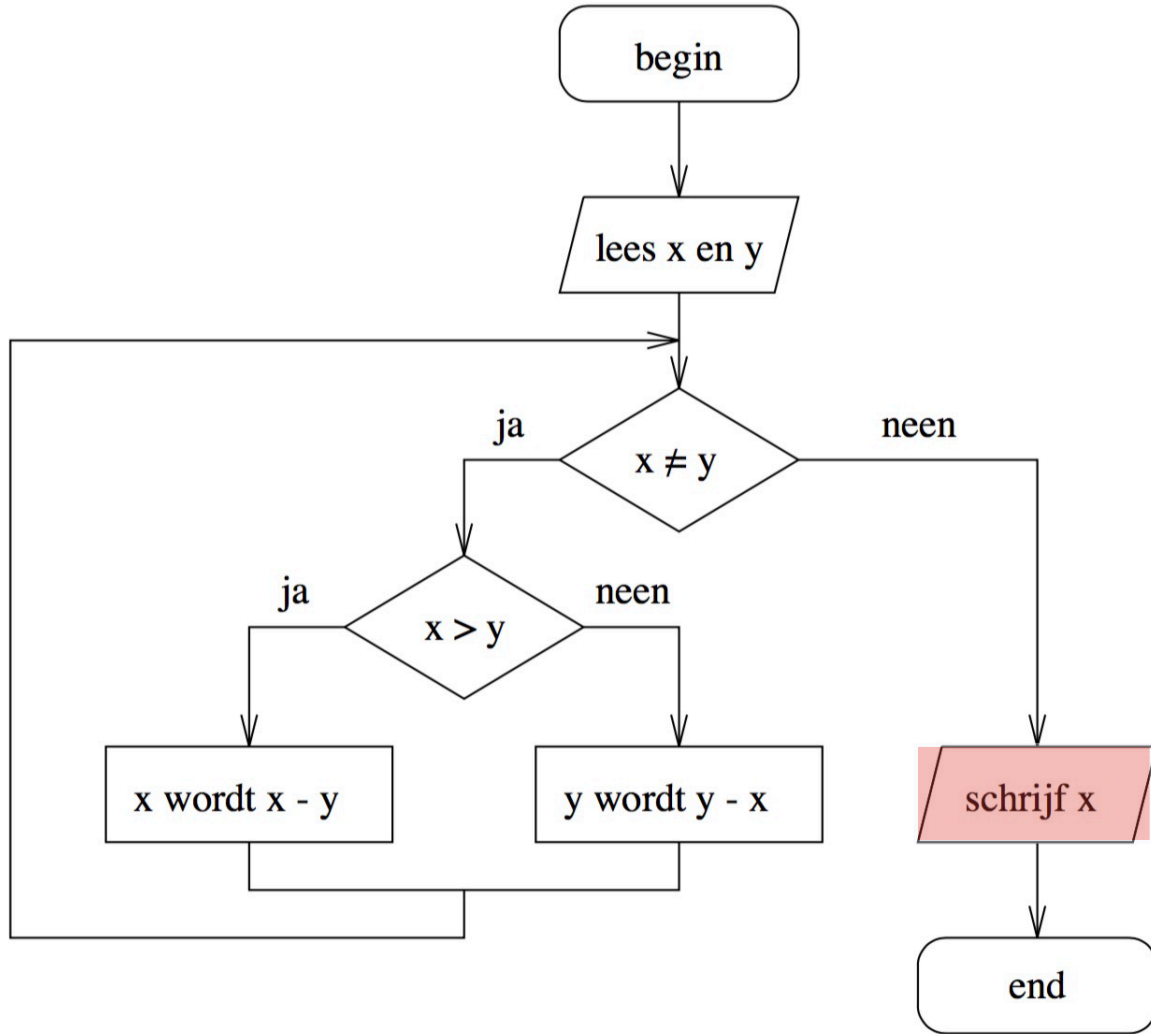


# Berekenen van ggd: Algoritme van Euclides



x	y
12	8
4	8
4	4

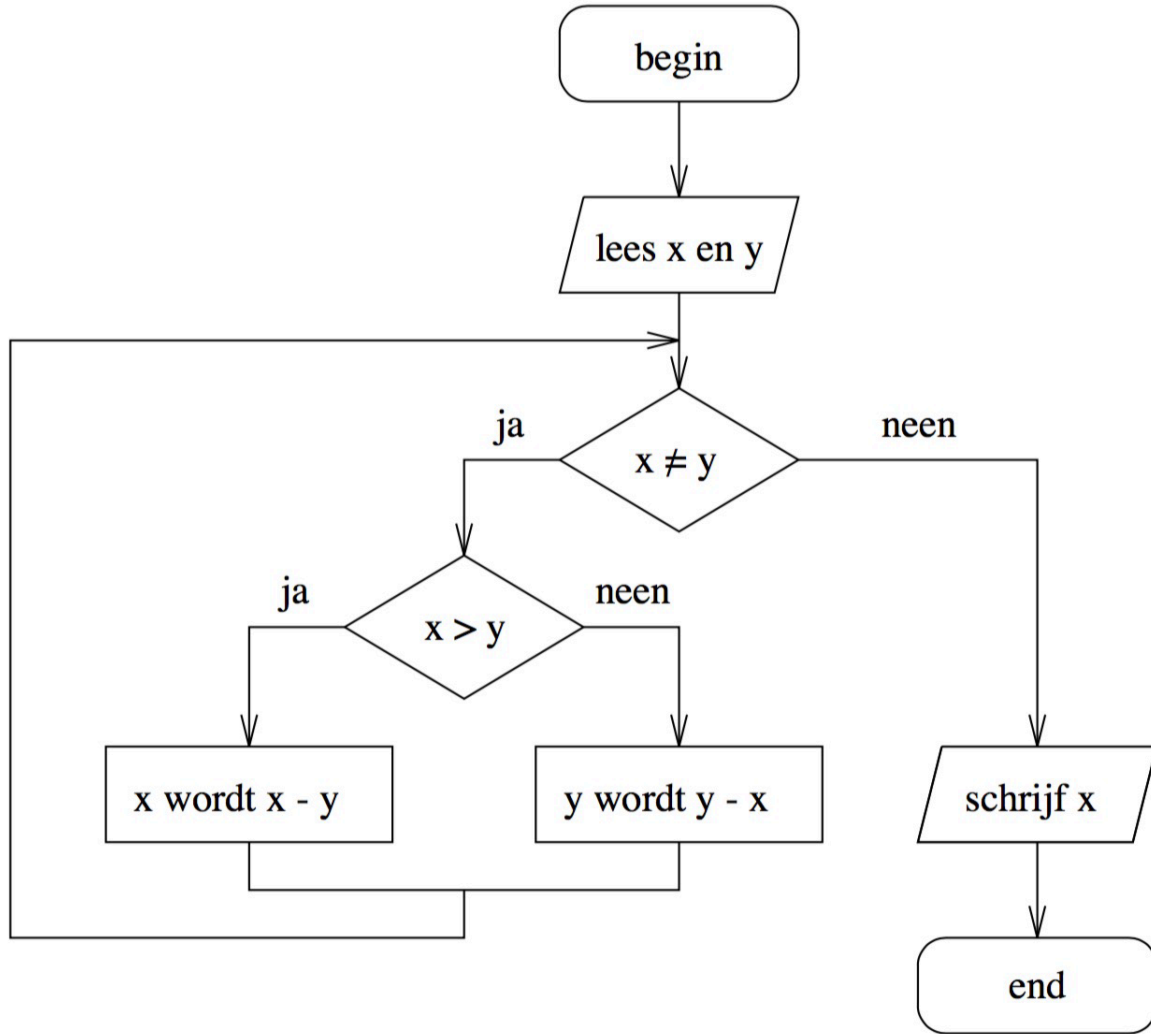
# Berekenen van ggd: Algoritme van Euclides



x	y
12	8
4	8
4	4

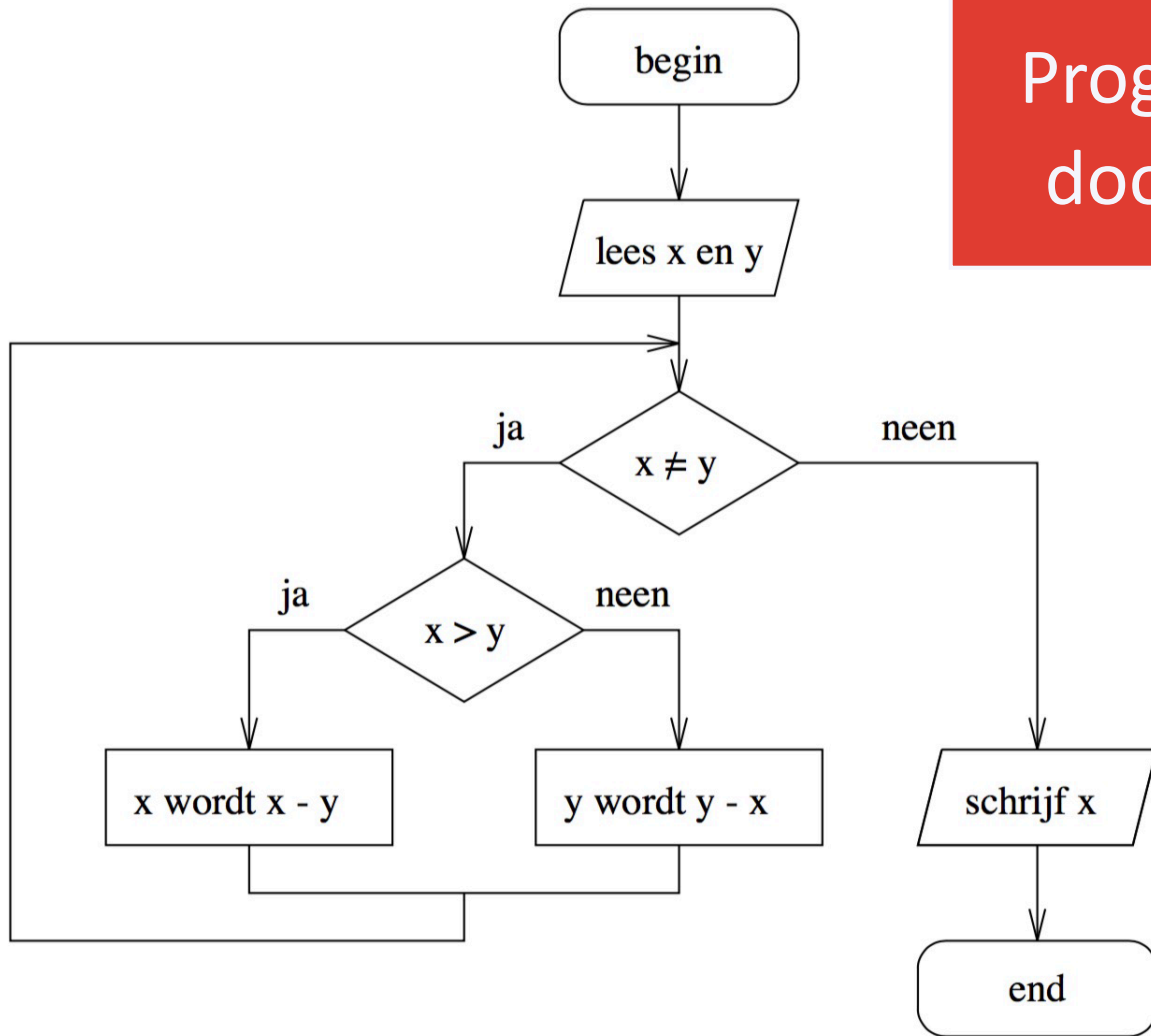
Antwoord is 4

# Berekenen van ggd: Algoritme van Euclides



- Stapsgewijs
- Eenvoudig om uit te voeren
- Correctheid?

Programma = algoritme dat rechtstreeks door computer kan uitgevoerd worden

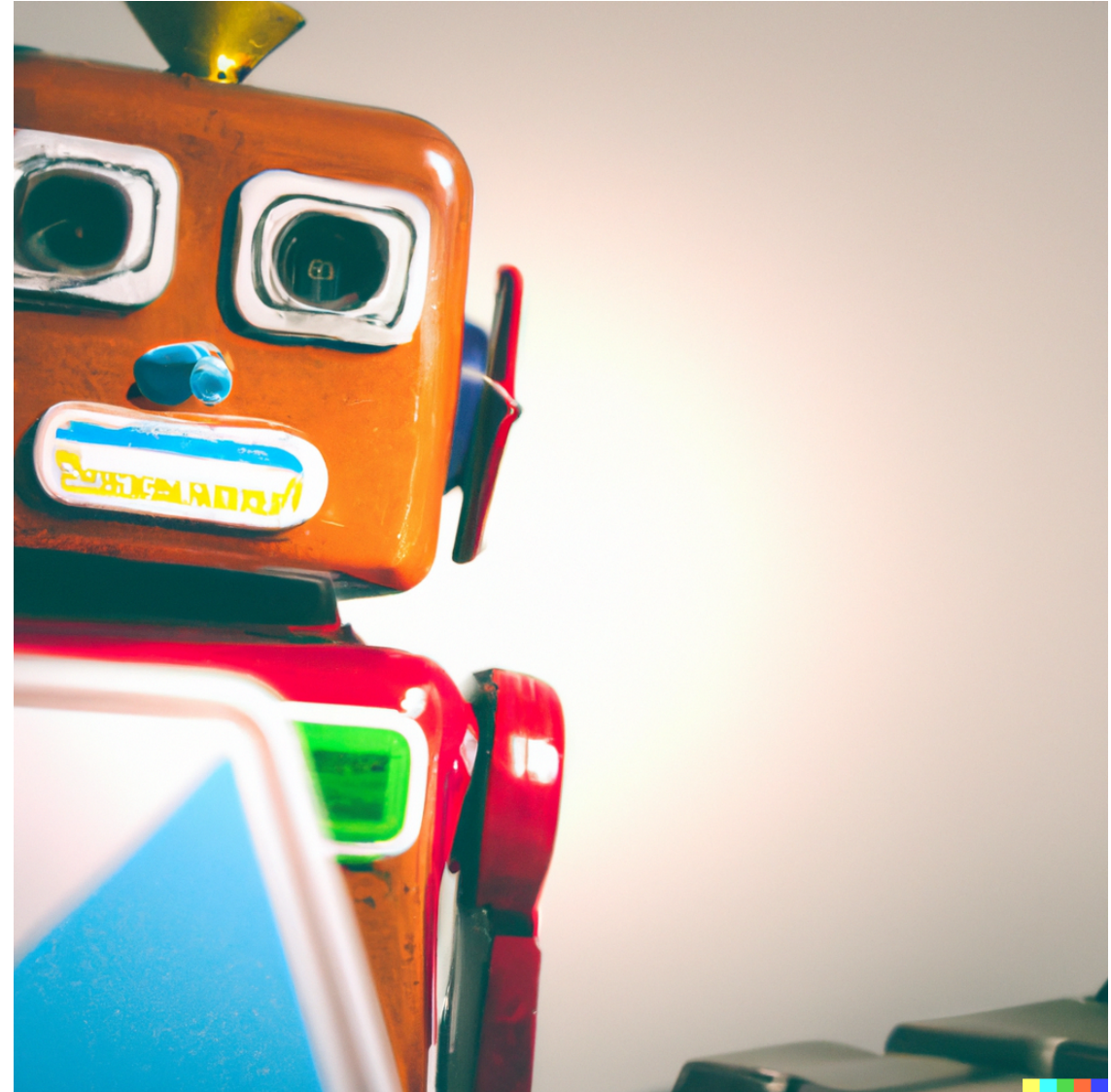


```
x = int(input("Geef een waarde voor x "))
y = int(input("Geef een waarde voor y "))
while x != y:
    if x > y:
        x = x - y
    else:
        y = y - x
print("De ggd is ", x)
```



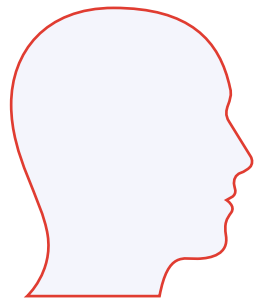
# Programmeren

- Bedenken van een algoritme
  - Creatief proces
  - Kennis
    - Bouwstenen van algoritmen
    - Standaardalgoritmen
    - Datastructuren
- Implementeren
  - Iteratief proces (testen, debuggen)
  - Kennis programmeertaal

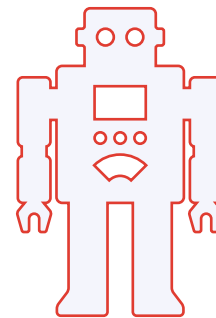


# Bedenken van algoritmen

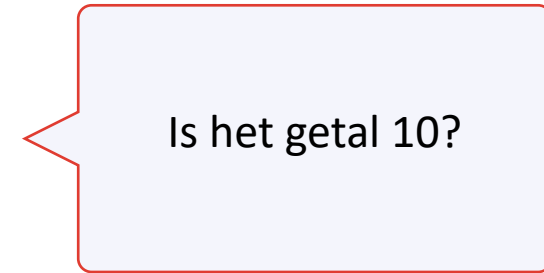
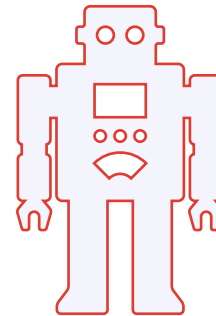
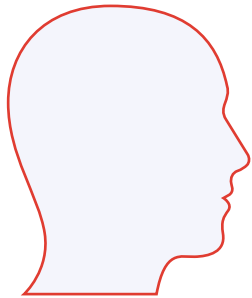
# Raden van een geheim getal



Ik heb een geheim  
getal gekozen tussen 1  
en 100.

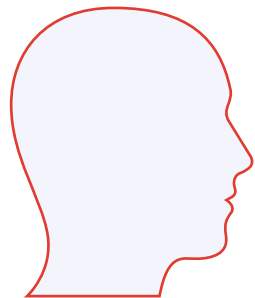


# Raden van een geheim getal

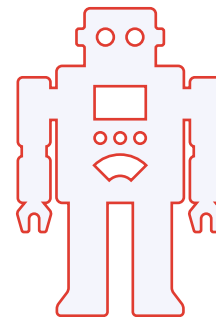




# Raden van een geheim getal

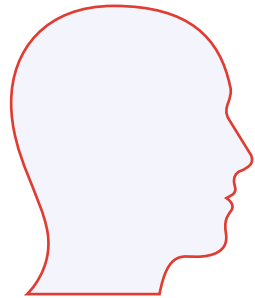


Nee, het geheime  
getal is groter?

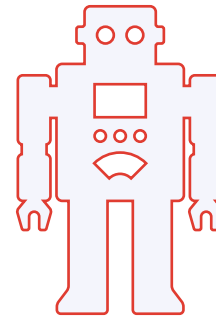


Is het getal 10?

# Raden van een geheim getal

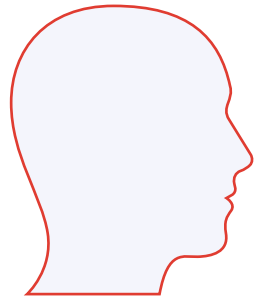


Nee, het geheime getal is kleiner?

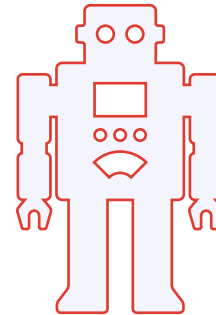


Is het getal 90?

# Raden van een geheim getal



Nee, het geheime getal is kleiner?



Is het getal 90?

**Wat is de beste strategie voor de robot?**

# Algoritme: binair zoeken

- Basisidee: gok telkens het **middelste getal** om de zoekruimte zoveel mogelijk te verkleinen

# Algoritme: binair zoeken

- Basisidee: gok telkens het **middelste getal** om de zoekruimte zoveel mogelijk te verkleinen

1

100



# Algoritme: binair zoeken

- Basisidee: gok telkens het **middelste getal** om de zoekruimte zoveel mogelijk te verkleinen

1



100



# Algoritme: binair zoeken

- Basisidee: gok telkens het **middelste getal** om de zoekruimte zoveel mogelijk te verkleinen

1



100



# Algoritme: binair zoeken

- Basisidee: gok telkens het **middelste getal** om de zoekruimte zoveel mogelijk te verkleinen

1



100



1

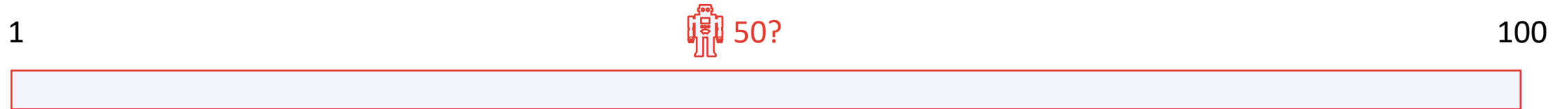
49





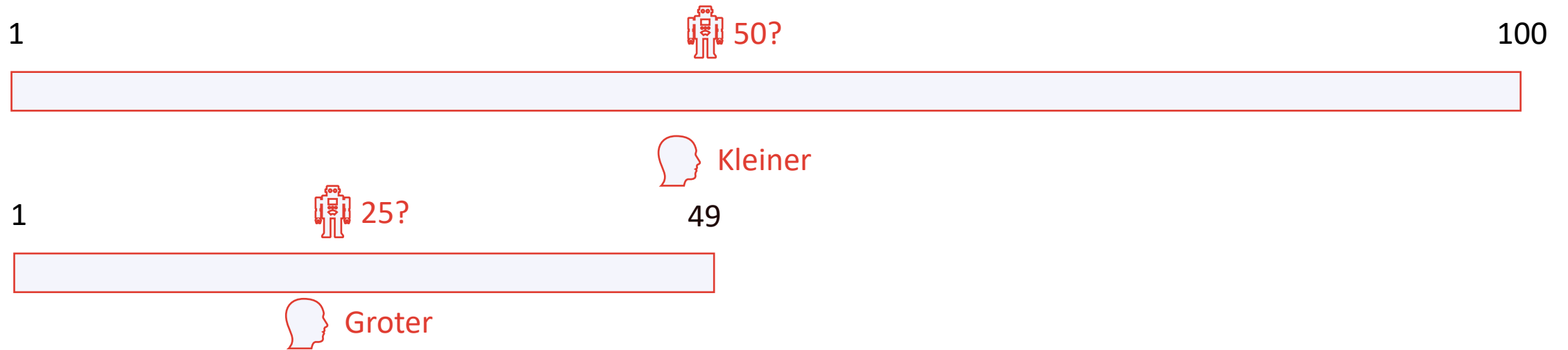
# Algoritme: binair zoeken

- Basisidee: gok telkens het **middelste getal** om de zoekruimte zoveel mogelijk te verkleinen



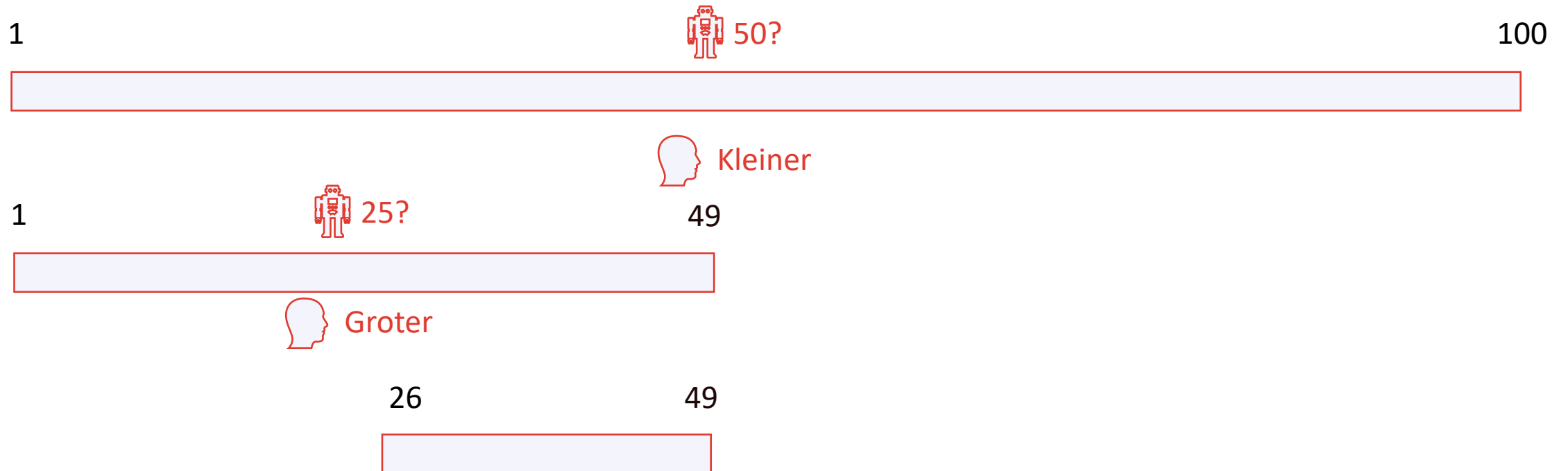
# Algoritme: binair zoeken

- Basisidee: gok telkens het **middelste getal** om de zoekruimte zoveel mogelijk te verkleinen



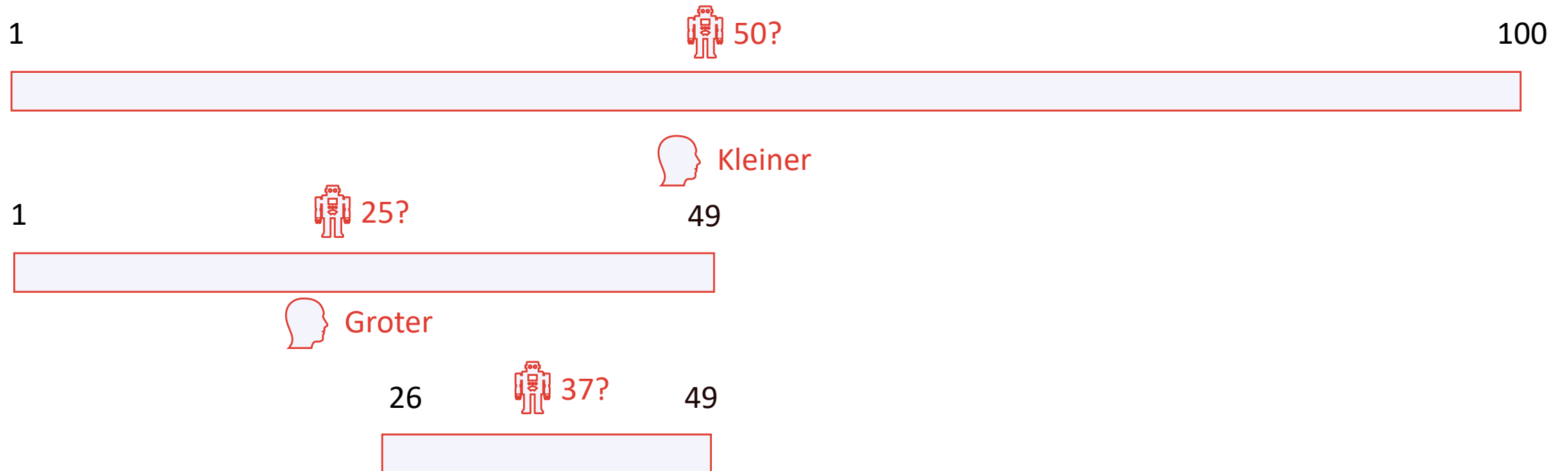
# Algoritme: binair zoeken

- Basisidee: gok telkens het **middelste getal** om de zoekruimte zoveel mogelijk te verkleinen



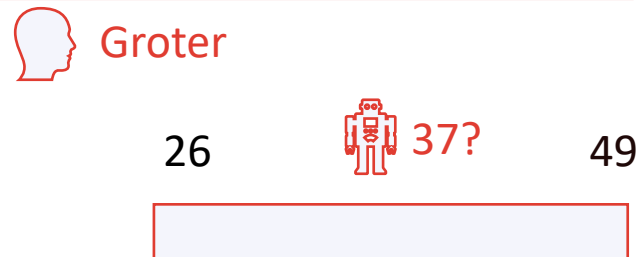
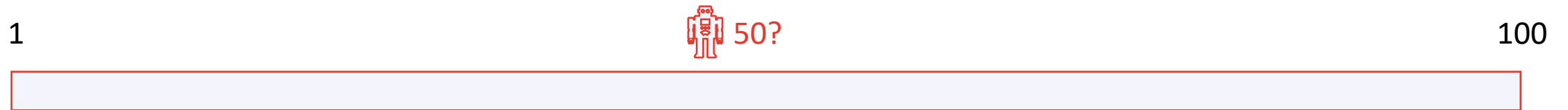
# Algoritme: binair zoeken

- Basisidee: gok telkens het **middelste getal** om de zoekruimte zoveel mogelijk te verkleinen

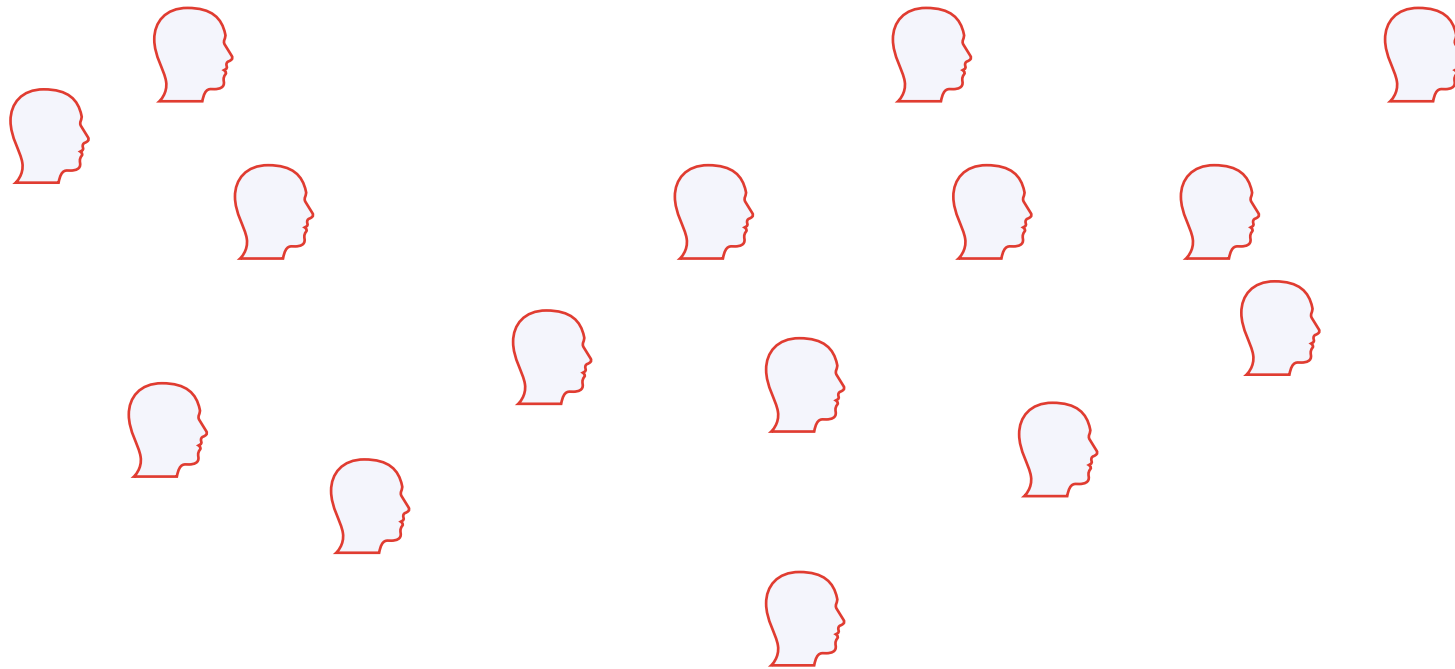


# Algoritme: binair zoeken

- Basisidee: gok telkens het **middelste getal** om de zoekruimte zoveel mogelijk te verkleinen



# Aantal personen tellen in een groep

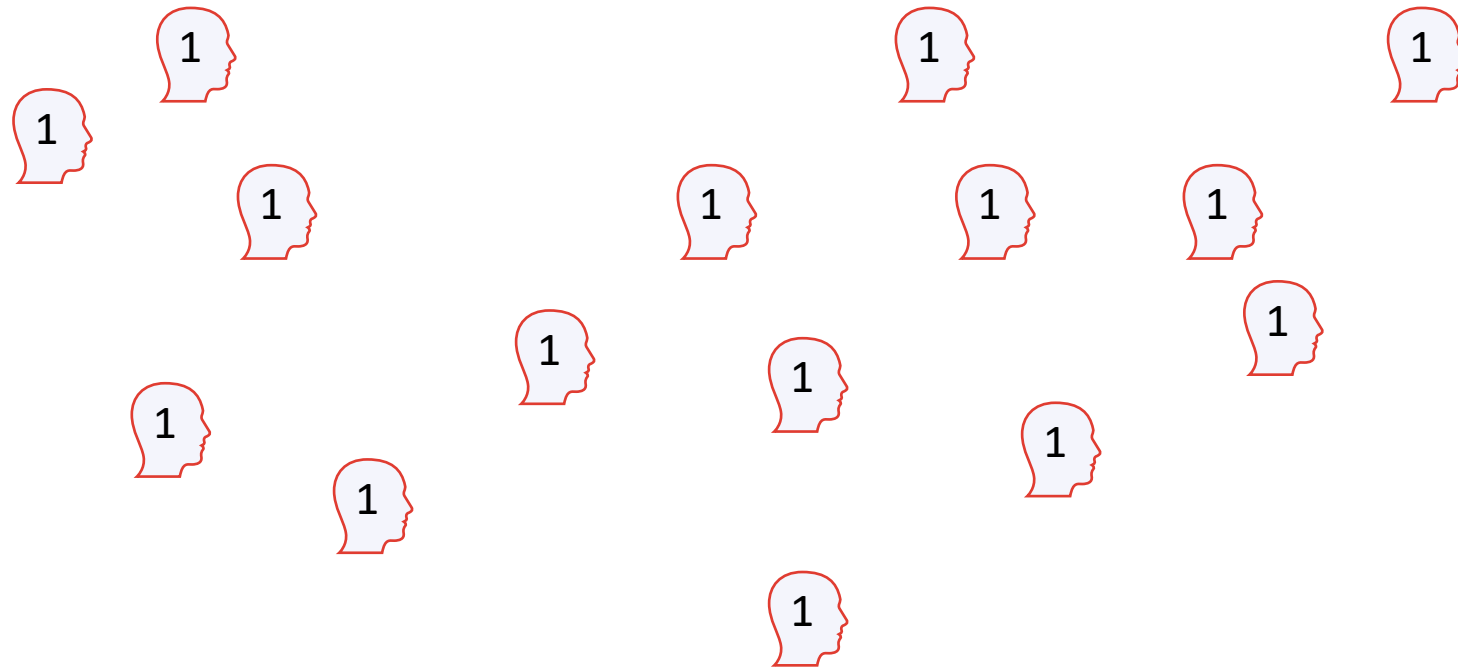


# Aantal personen tellen in een groep

Basisidee:

- Werken in rondes
- In elke ronde wordt het aantal nog te tellen personen gehalveerd

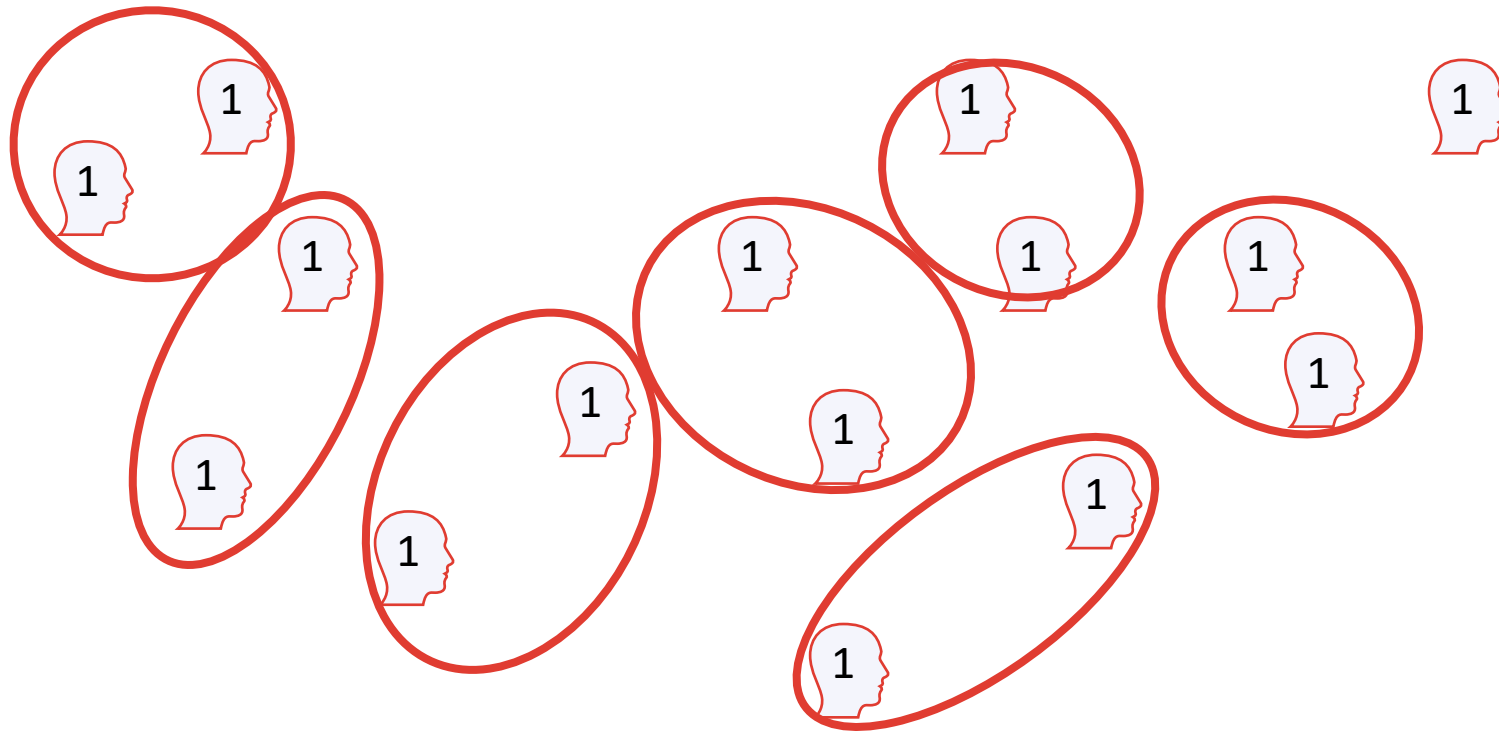
# Aantal personen tellen in een groep





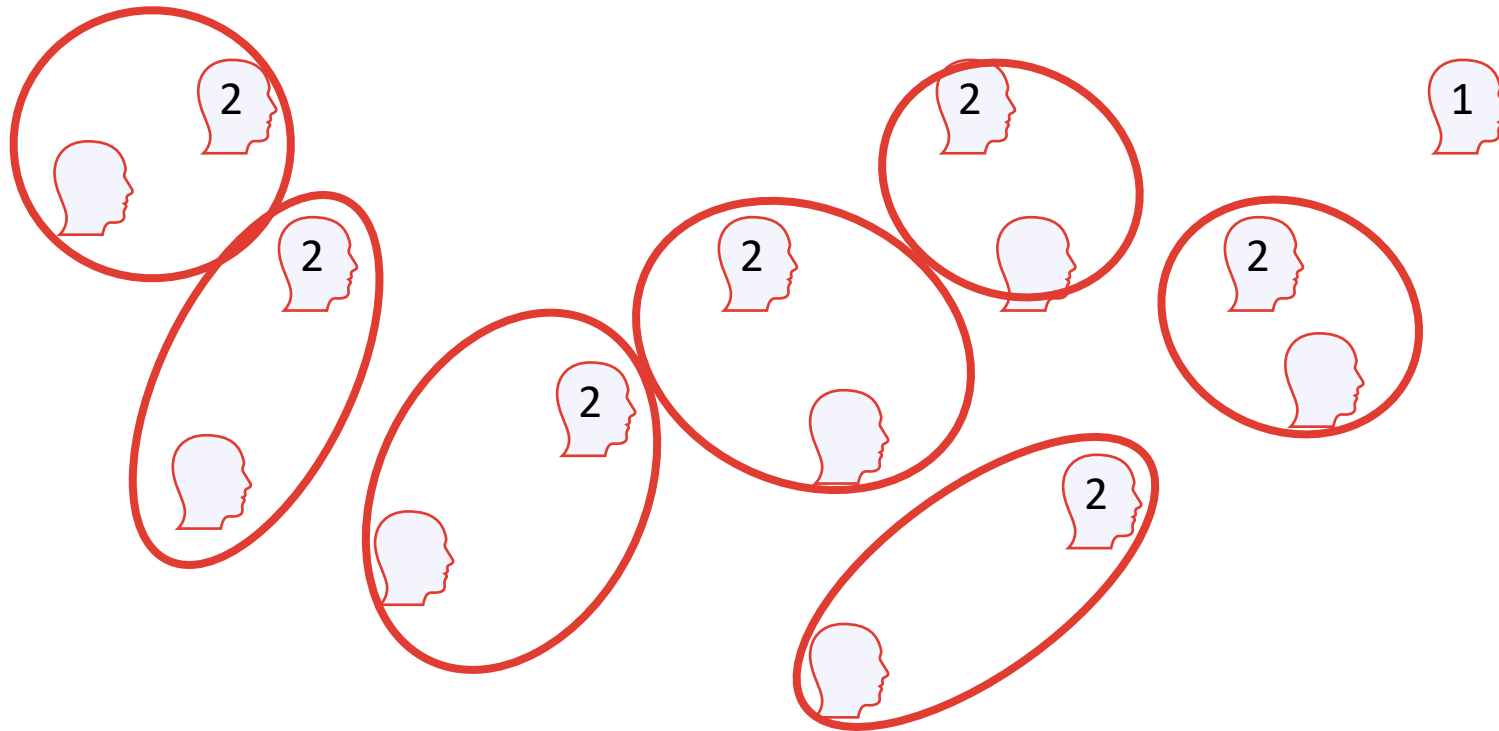
# Aantal personen tellen in een groep

1ste telling



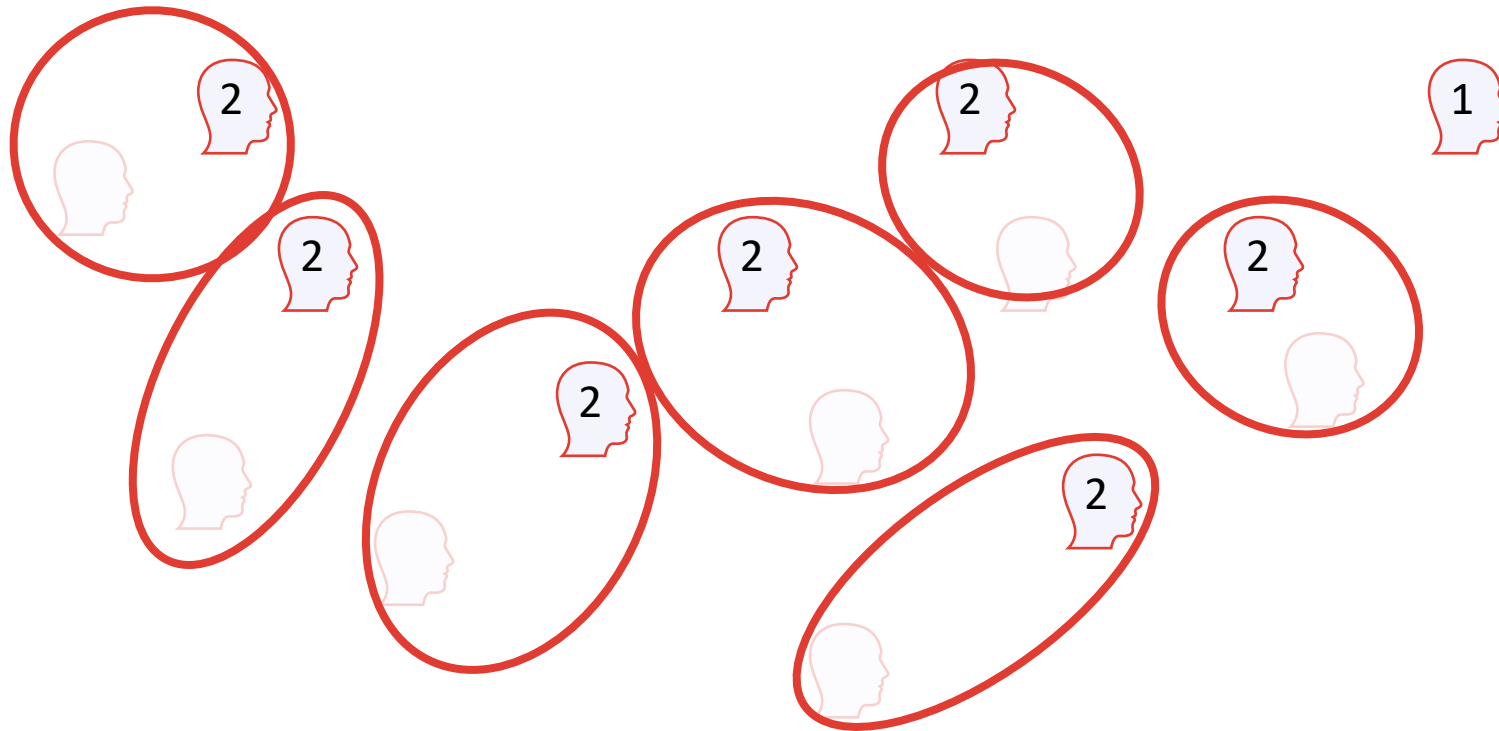
# Aantal personen tellen in een groep

1ste telling



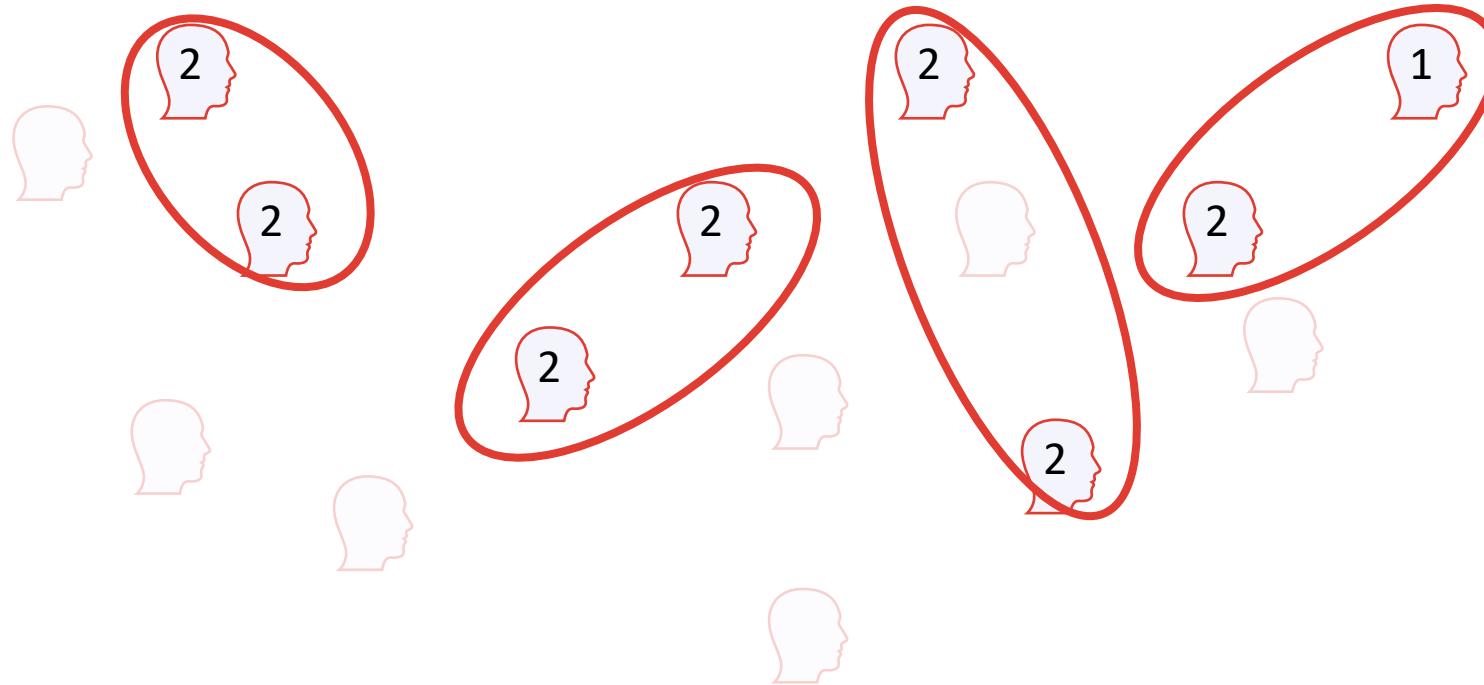
# Aantal personen tellen in een groep

1ste telling



# Aantal personen tellen in een groep

2de telling



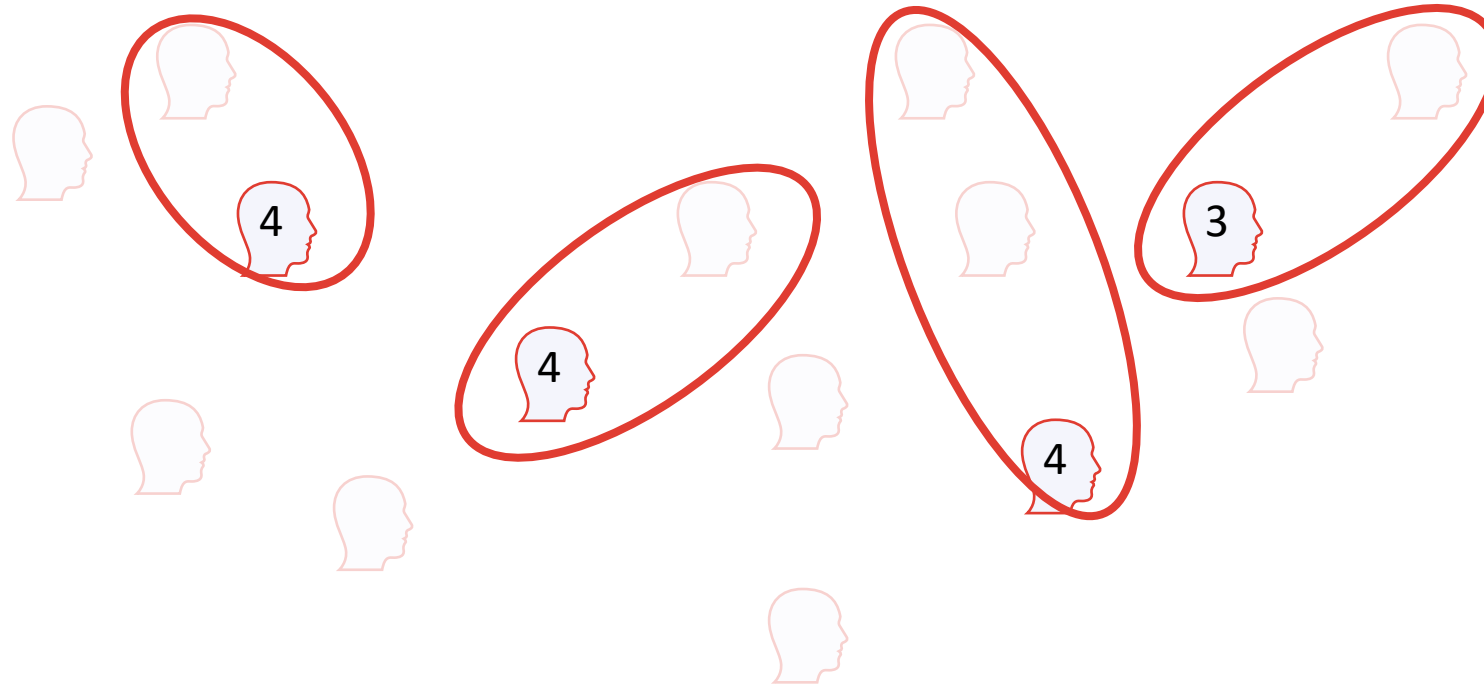
# Aantal personen tellen in een groep

2de telling



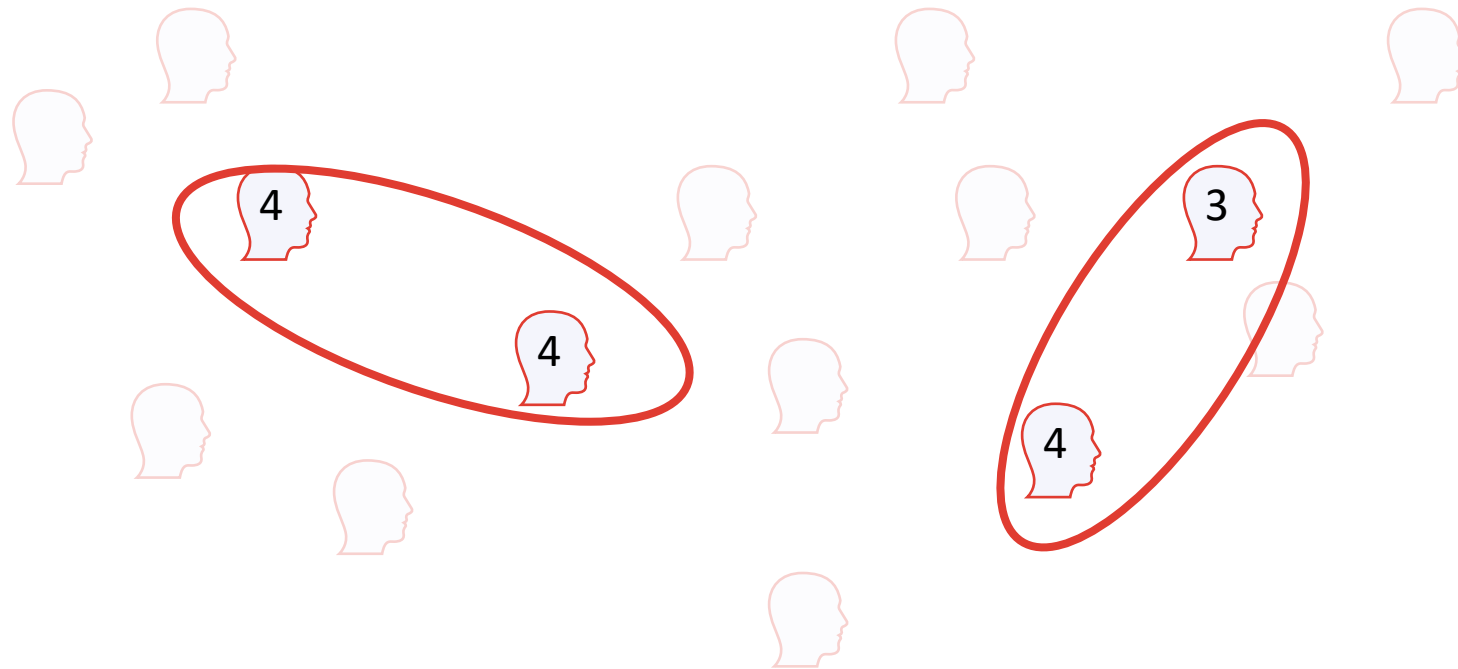
# Aantal personen tellen in een groep

2de telling



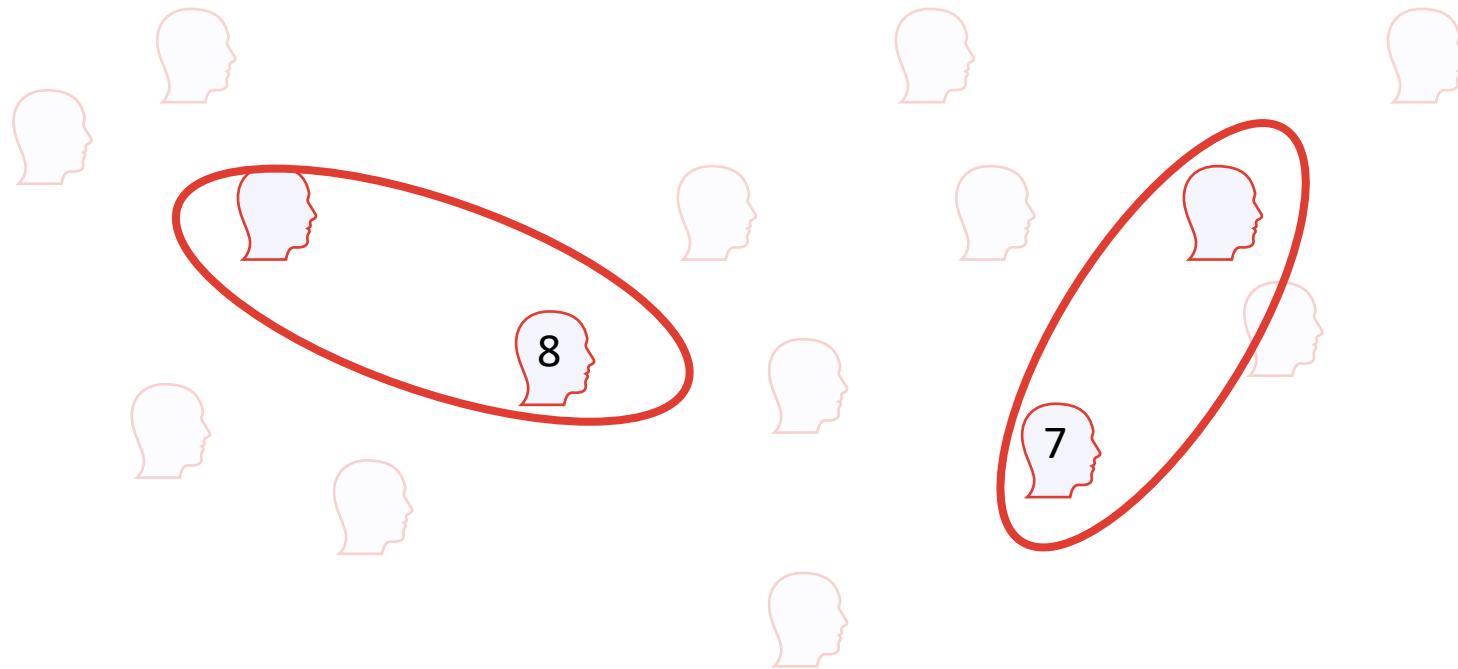
# Aantal personen tellen in een groep

3de telling



# Aantal personen tellen in een groep

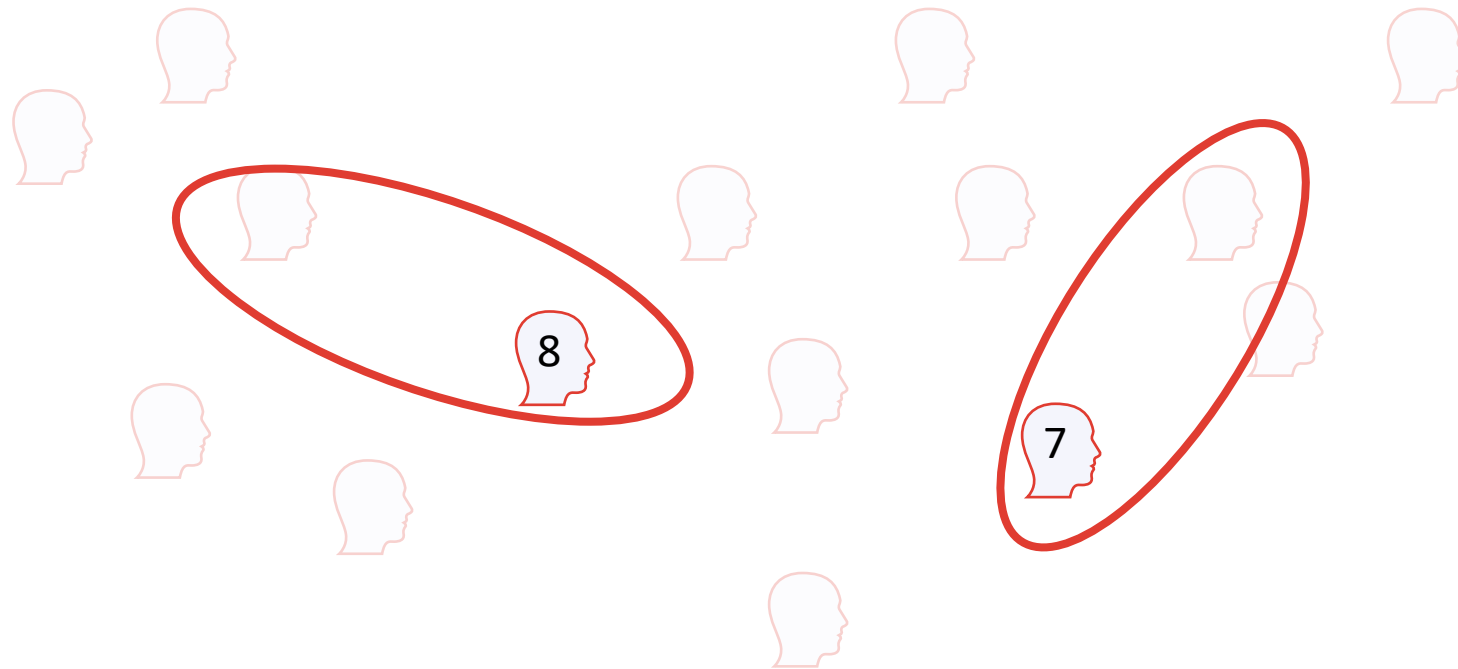
3de telling





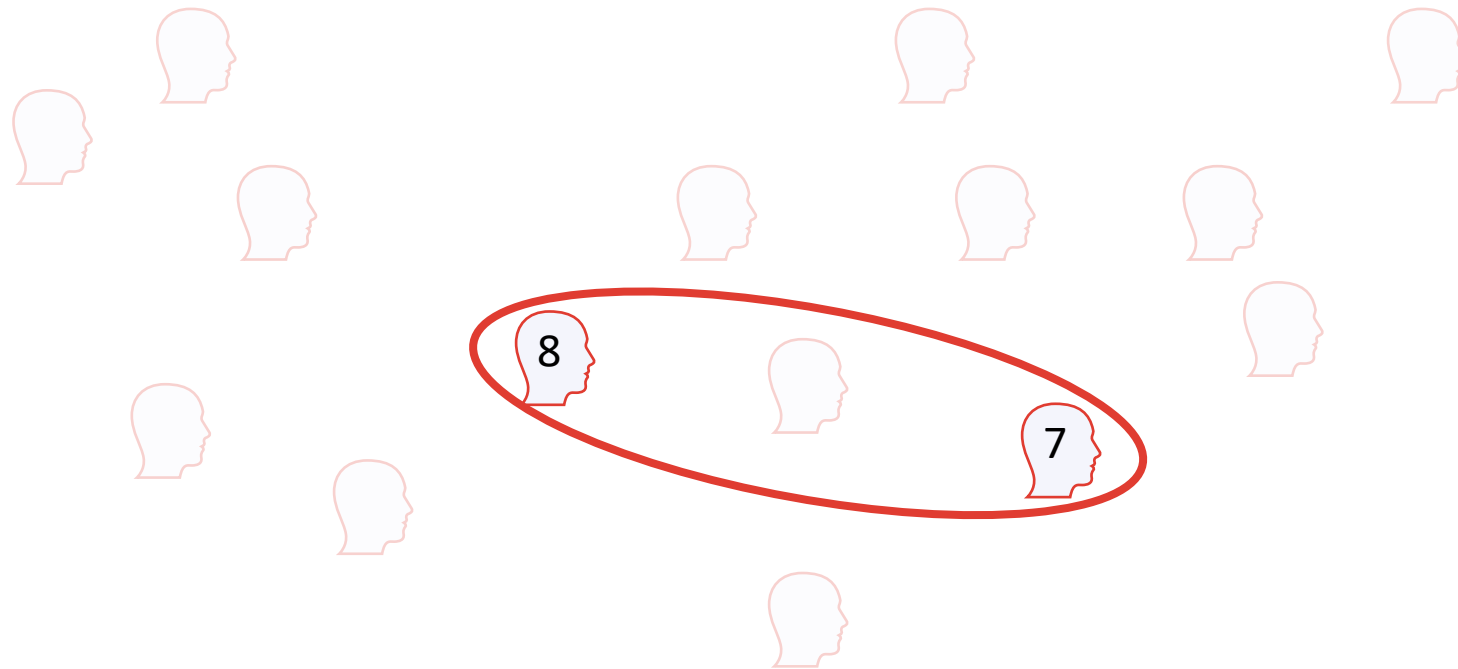
# Aantal personen tellen in een groep

3de telling



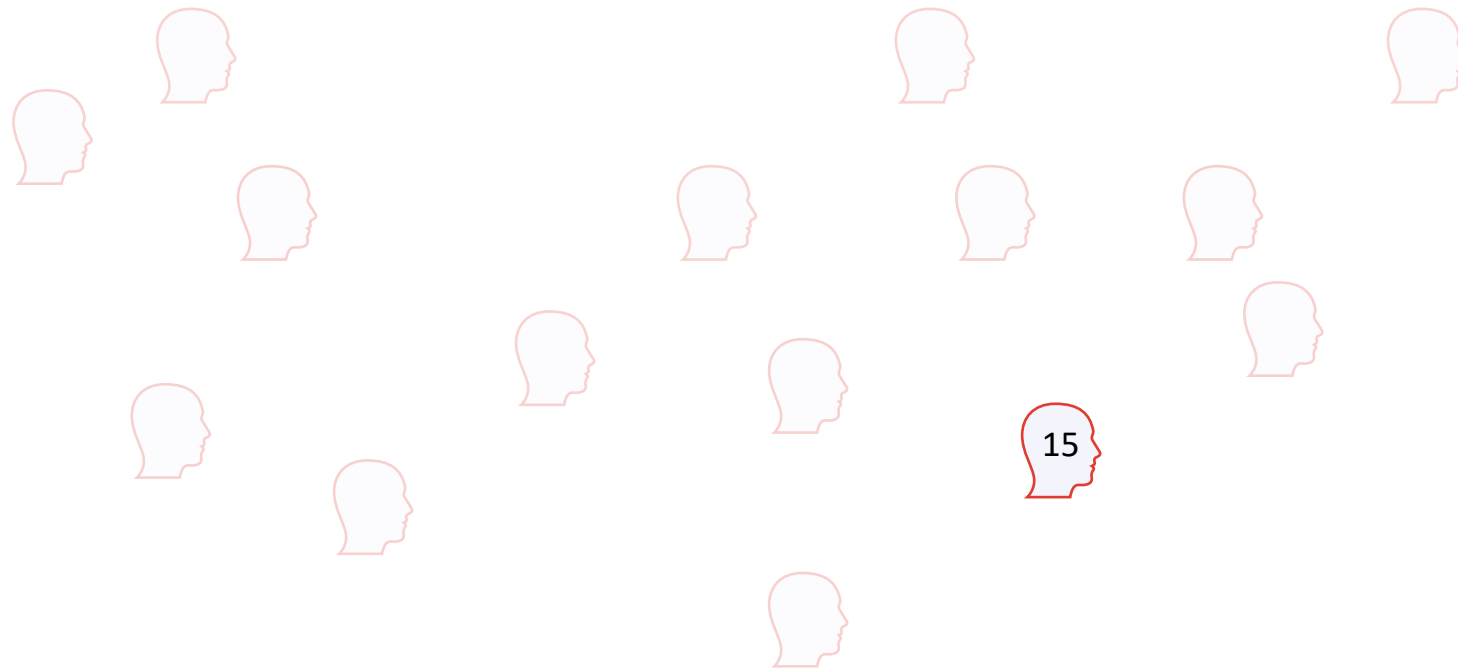
# Aantal personen tellen in een groep

4de telling



# Aantal personen tellen in een groep

4de telling



# Voorbeeld van datastructuur: lijst

A ● ■  
B ■ ● ● ●  
C ■ ● ■ ●  
D ■ ● ●  
E ●  
F ● ● ■ ●  
G ■ ■ ●  
H ● ● ● ●  
I ● ●  
J ● ■ ■ ■  
K ■ ● ■  
L ● ■ ● ●  
M ■ ■  
N ■ ●  
O ■ ■ ■  
P ● ■ ■ ●  
Q ■ ■ ● ■  
R ● ■ ●  
S ● ● ●  
T ■

U ● ● ■  
V ● ● ● ■  
W ● ■ ■  
X ■ ● ● ■  
Y ■ ● ■ ■  
Z ■ ■ ● ●

1 ● ■ ■ ■ ■  
2 ● ● ■ ■ ■  
3 ● ● ● ■ ■  
4 ● ● ● ● ■  
5 ● ● ● ● ●  
6 ■ ● ● ● ●  
7 ■ ■ ● ● ●  
8 ■ ■ ■ ● ●  
9 ■ ■ ■ ■ ●  
0 ■ ■ ■ ■ ■

# Voorbeeld van datastructuur: lijst

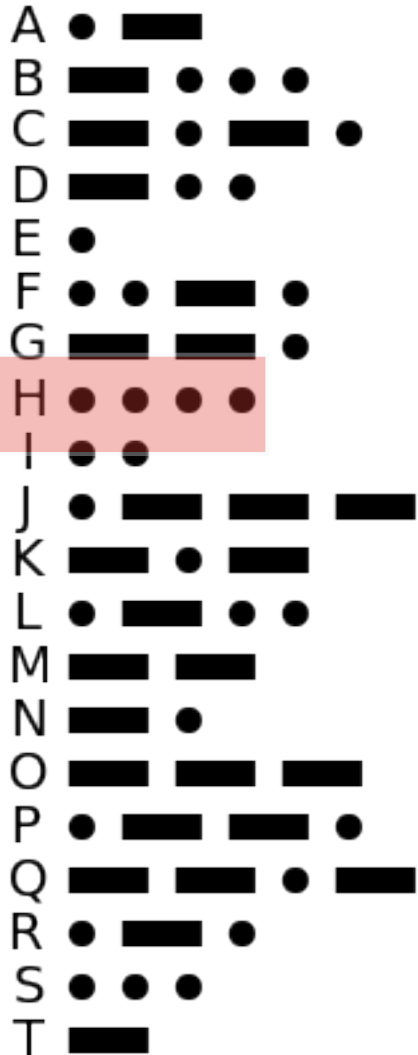
A ● ■  
B ■ ● ● ●  
C ■ ● ■ ●  
D ■ ● ●  
E ●  
F ● ● ■ ●  
G ■ ■ ●  
H ● ● ● ●  
I ● ●  
J ● ■ ■ ■  
K ■ ● ■  
L ● ■ ● ●  
M ■ ■  
N ■ ●  
O ■ ■ ■  
P ● ■ ■ ●  
Q ■ ■ ● ■  
R ● ■ ●  
S ● ● ●  
T ■

U ● ● ■  
V ● ● ● ■  
W ● ■ ■  
X ■ ● ● ■  
Y ■ ● ■ ■  
Z ■ ■ ● ●

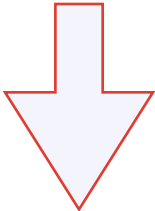
1 ● ■ ■ ■ ■  
2 ● ● ■ ■ ■  
3 ● ● ● ■ ■  
4 ● ● ● ● ■  
5 ● ● ● ● ●  
6 ■ ● ● ● ●  
7 ■ ■ ● ● ●  
8 ■ ■ ■ ● ●  
9 ■ ■ ■ ■ ●  
0 ■ ■ ■ ■ ■

Hallo

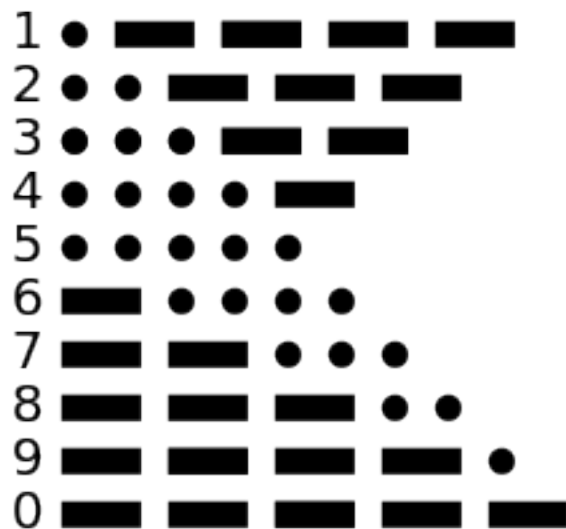
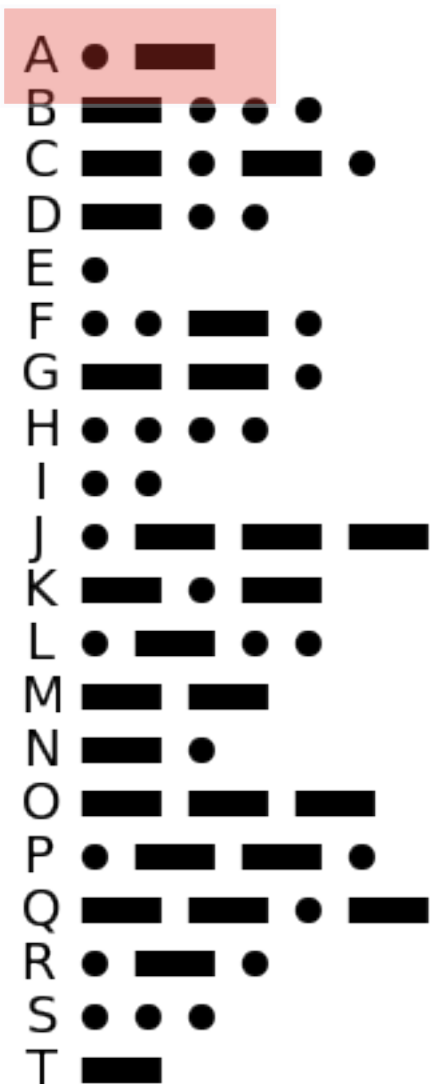
# Voorbeeld van datastructuur: lijst



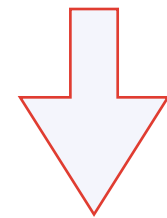
Hallo



# Voorbeeld van datastructuur: lijst



Hallo



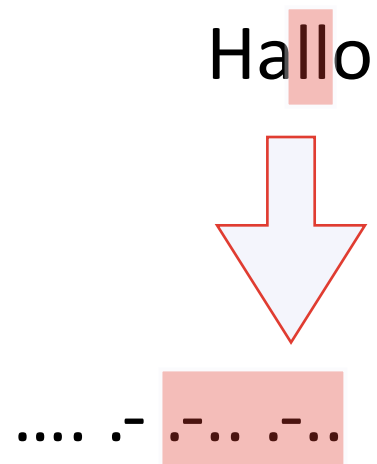
.... -

# Voorbeeld van datastructuur: lijst

A ● ■■■  
B ■■■ ● ● ●  
C ■■■ ● ■■■ ●  
D ■■■ ● ●  
E ●  
F ● ● ■■■ ●  
G ■■■ ■■■ ●  
H ● ● ● ●  
I ● ●  
J ● ■■■ ■■■ ■■■  
K ■■■ ● ■■■  
L ● ■■■ ● ●  
M ■■■ ■■■  
N ■■■ ●  
O ■■■ ■■■ ■■■  
P ● ■■■ ■■■ ●  
Q ■■■ ■■■ ● ■■■  
R ● ■■■ ●  
S ● ● ●  
T ■■■

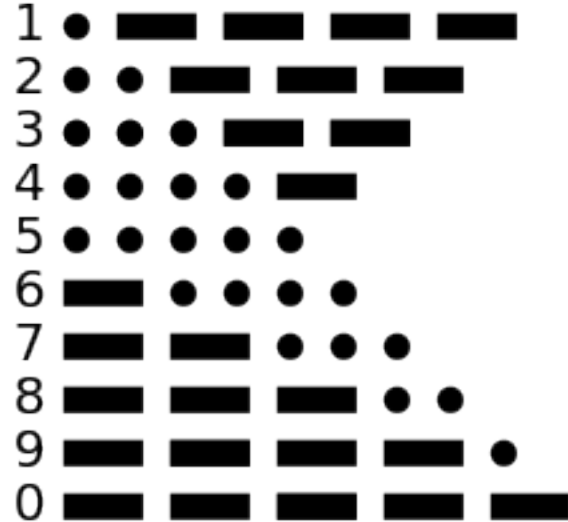
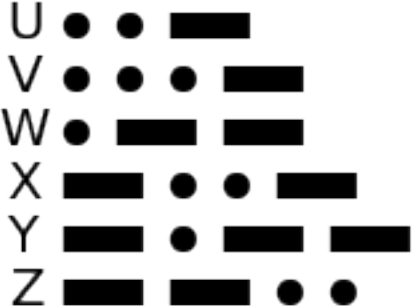
U ● ● ■■■  
V ● ● ● ■■■  
W ● ■■■ ■■■  
X ■■■ ● ● ■■■  
Y ■■■ ● ■■■ ■■■  
Z ■■■ ■■■ ● ●

1 ● ■■■ ■■■ ■■■ ■■■  
2 ● ● ■■■ ■■■ ■■■  
3 ● ● ● ■■■ ■■■  
4 ● ● ● ● ■■■  
5 ● ● ● ● ●  
6 ■■■ ● ● ● ●  
7 ■■■ ■■■ ● ● ●  
8 ■■■ ■■■ ■■■ ● ●  
9 ■■■ ■■■ ■■■ ■■■ ●  
0 ■■■ ■■■ ■■■ ■■■ ■■■

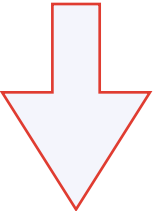




# Voorbeeld van datastructuur: lijst



Hallo



# Voorbeeld van datastructuur: lijst

A ● ■  
B ■ ● ● ●  
C ■ ● ■ ●  
D ■ ● ●  
E ●  
F ● ● ■ ●  
G ■ ■ ●  
H ● ● ● ●  
I ● ●  
J ● ■ ■ ■  
K ■ ● ■  
L ● ■ ● ●  
M ■ ■  
N ■ ●  
O ■ ■ ■  
P ● ■ ■ ●  
Q ■ ■ ● ■  
R ● ■ ●  
S ● ● ●  
T ■

U ● ● ■  
V ● ● ● ■  
W ● ■ ■  
X ■ ● ● ■  
Y ■ ● ■ ■  
Z ■ ■ ● ●

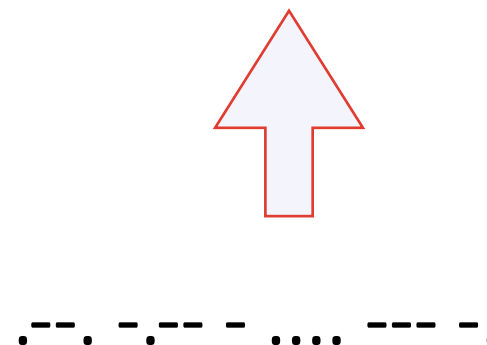
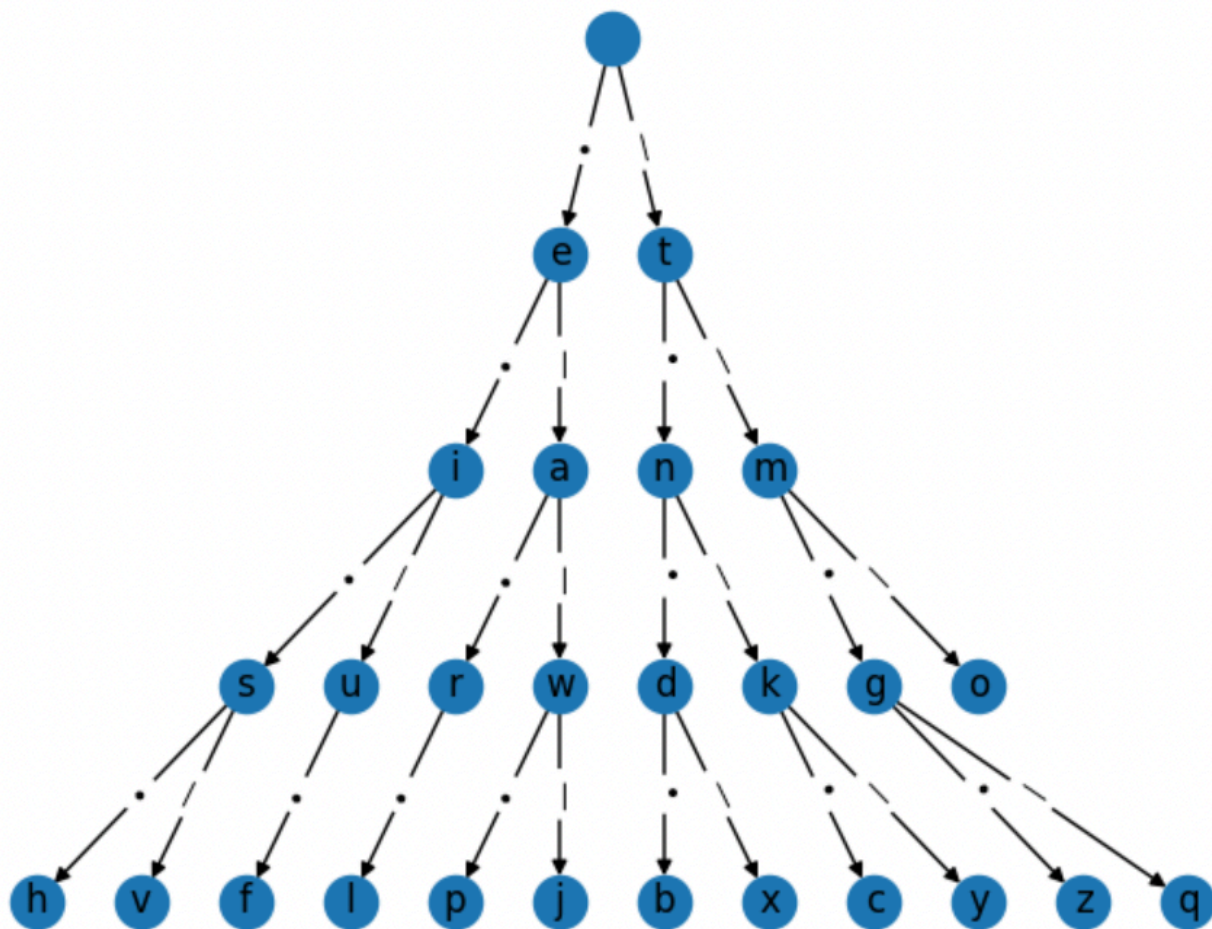
1 ● ■ ■ ■ ■  
2 ● ● ■ ■ ■  
3 ● ● ● ■ ■  
4 ● ● ● ● ■  
5 ● ● ● ● ●  
6 ■ ● ● ● ●  
7 ■ ■ ● ● ●  
8 ■ ■ ■ ● ●  
9 ■ ■ ■ ■ ●  
0 ■ ■ ■ ■ ■

????

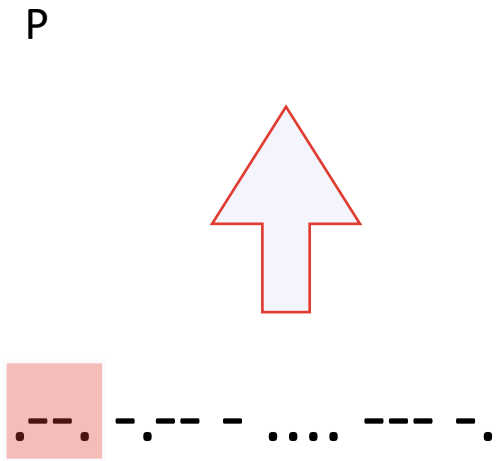
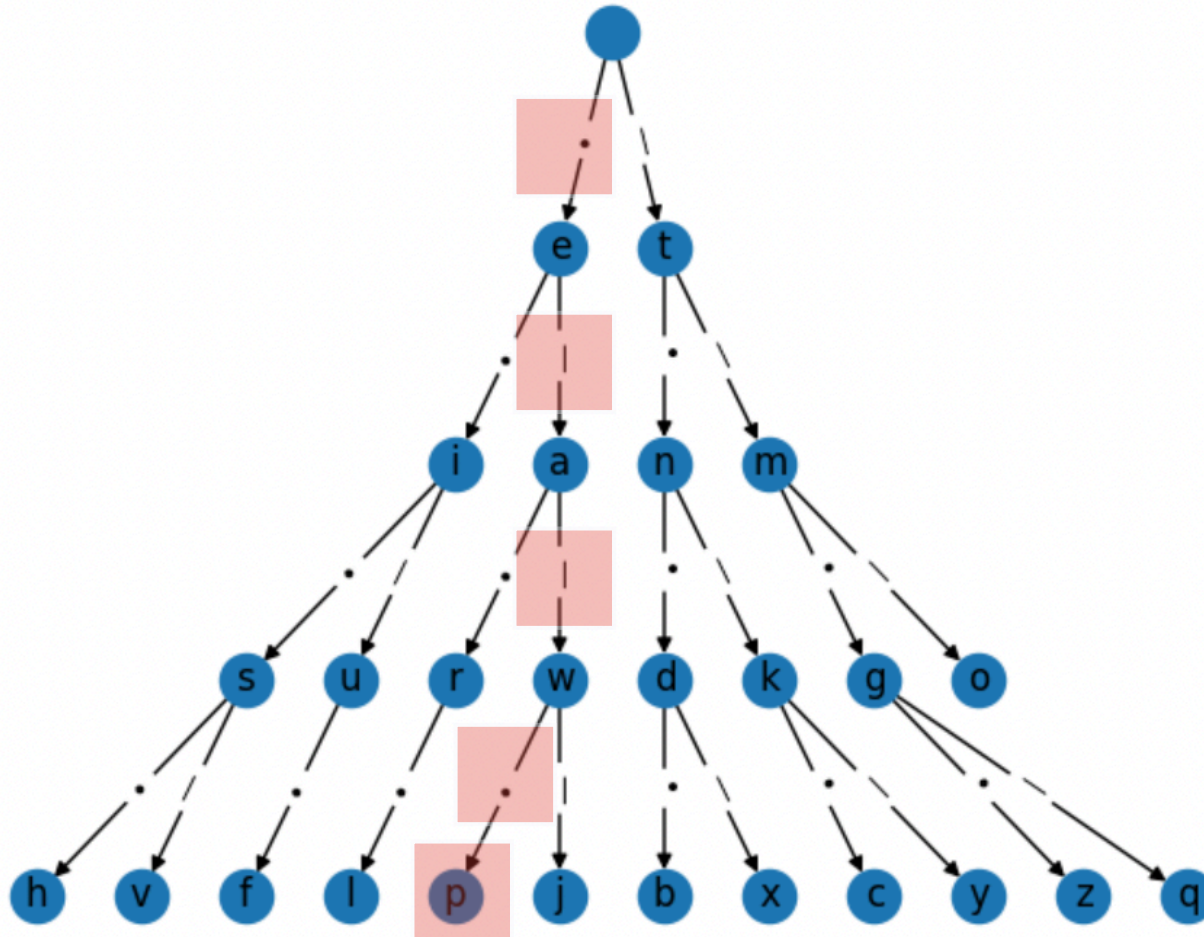


.....

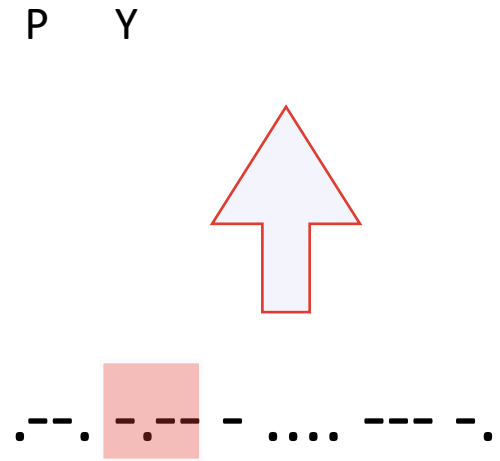
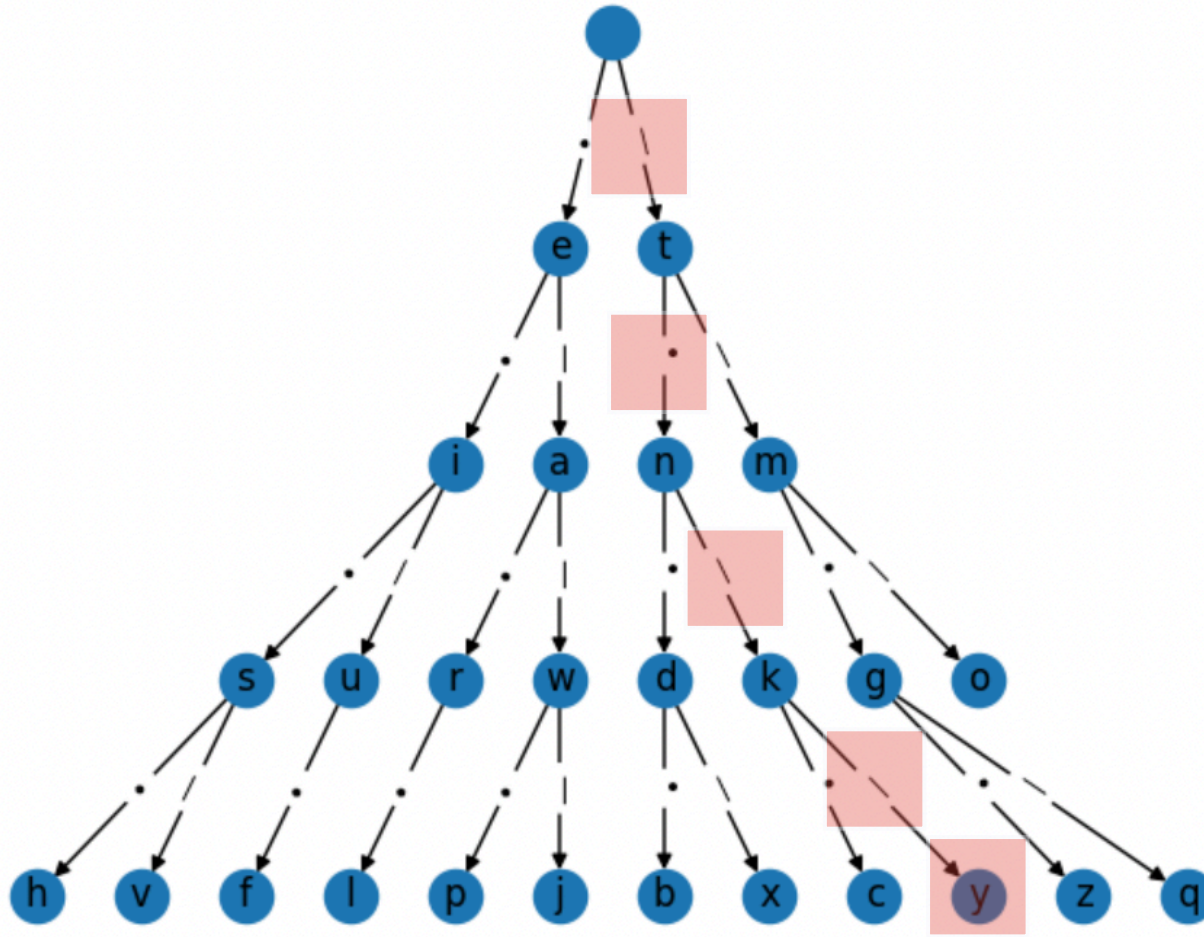
# Voorbeeld van datastructuur: zoekboom



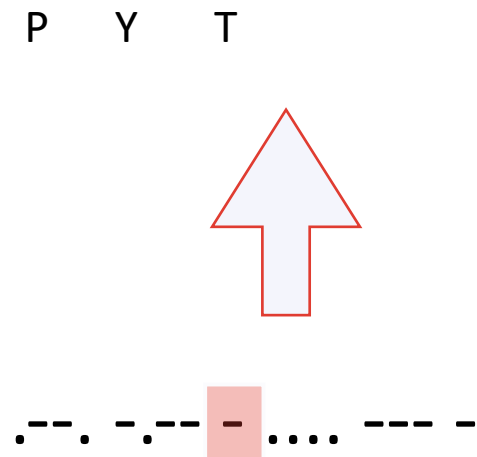
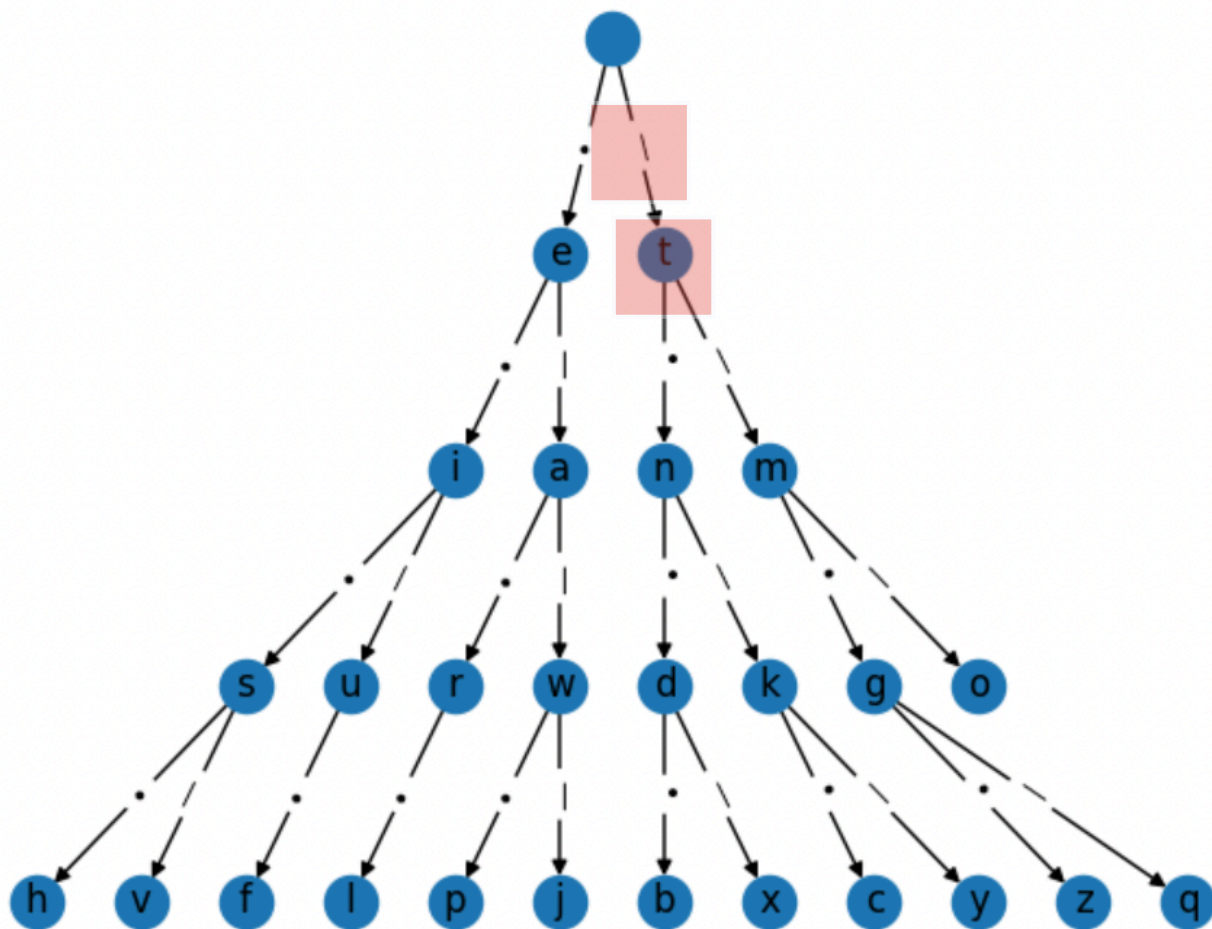
# Voorbeeld van datastructuur: zoekboom



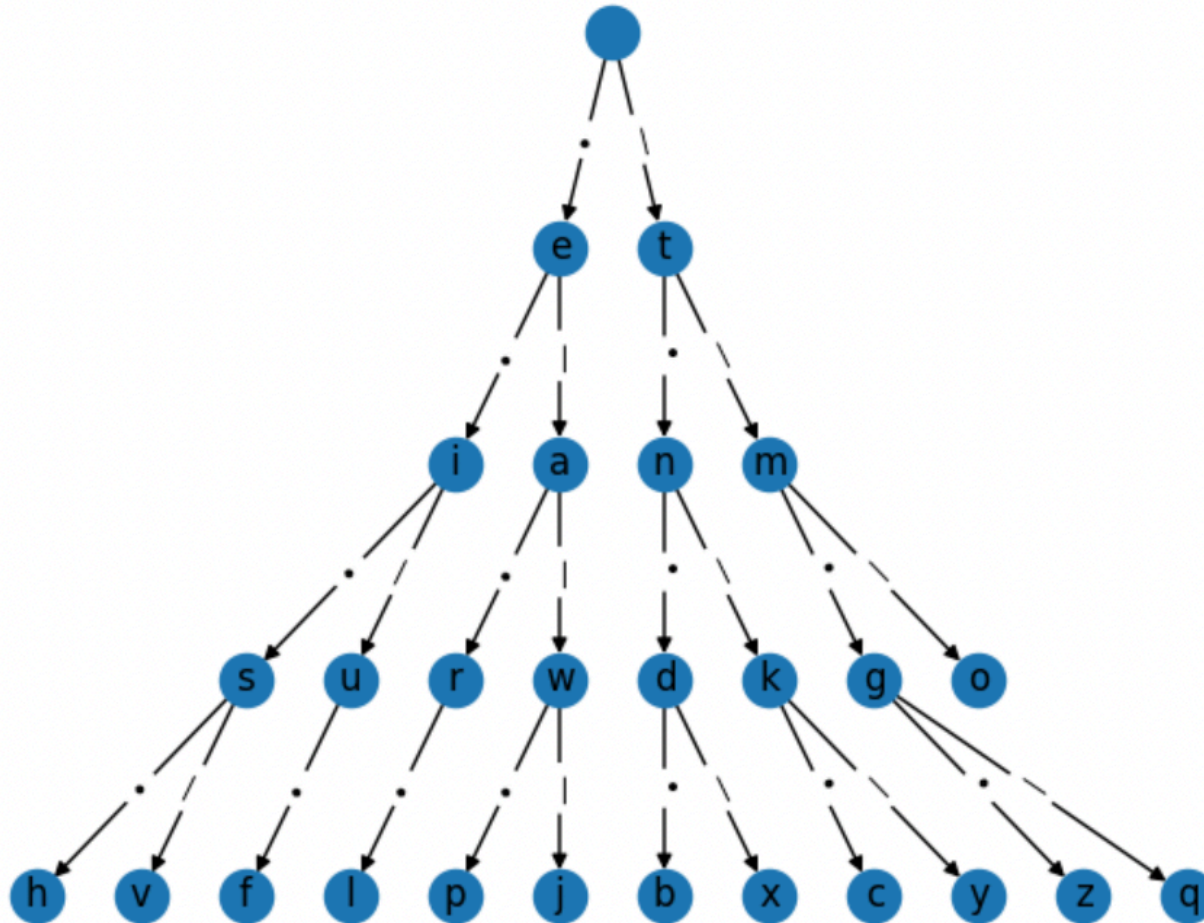
# Voorbeeld van datastructuur: zoekboom



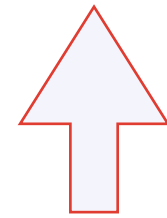
# Voorbeeld van datastructuur: zoekboom



# Voorbeeld van datastructuur: zoekboom



P Y T H O N



.....

# Programmeren

- Bedenken van een algoritme
  - Creatief proces
  - Kennis
    - Bouwstenen van algoritmen
    - Standaardalgoritmen
    - Datastructuren
- Implementeren
  - Iteratief proces (testen, debuggen)
  - Kennis programmeertaal



chatGPT ??





chatGPT

F

Schrijf een limerick over het plezier van programmeren



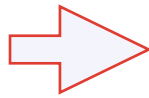
Een coder, heel blij met zijn scherm,  
Vond coderen vast, niet echt een term.  
Met een lach, heel sereen,  
Schreef hij code, heel clean,  
En de bugs overwon hij, standvastig en ferm.

# chatGPT in actie



chatGPT

# chatGPT in actie

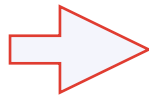


Schrijf een limerick over het plezier van programmeren



chatGPT

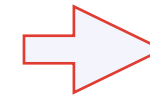
# chatGPT in actie



Schrijf een limerick over het plezier van programmeren

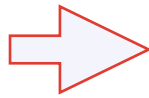


chatGPT

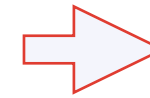


Een coder, heel blij met zijn scherm,  
Vond coderen vast, niet echt een term.  
Met een lach, heel sereen,  
Schreef hij code, heel clean,  
En de bugs overwon hij, standvastig  
en ferm.

# chatGPT in actie

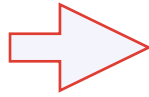


Schrijf een limerick over het plezier van programmeren

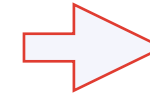


Een coder, heel blij met zijn scherm,  
Vond coderen vast, niet echt een term.  
Met een lach, heel sereen,  
Schreef hij code, heel clean,  
En de bugs overwon hij, standvastig  
en ferm.

# chatGPT in actie



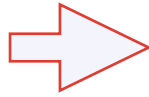
Schrijf een limerick over het plezier van programmeren



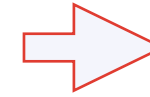
Een coder, heel blij met zijn scherm,  
Vond coderen vast, niet echt een term.  
Met een lach, heel sereen,  
Schreef hij code, heel clean,  
En de bugs overwon hij, standvastig  
en ferm.

Hoe tandwielen kiezen en plaatsen ?

# chatGPT in actie



Schrijf een limerick over het plezier van programmeren



Een coder, heel blij met zijn scherm,  
Vond coderen vast, niet echt een term.  
Met een lach, heel sereen,  
Schreef hij code, heel clean,  
En de bugs overwon hij, standvastig  
en ferm.

Hoe tandwielen kiezen en plaatsen ?

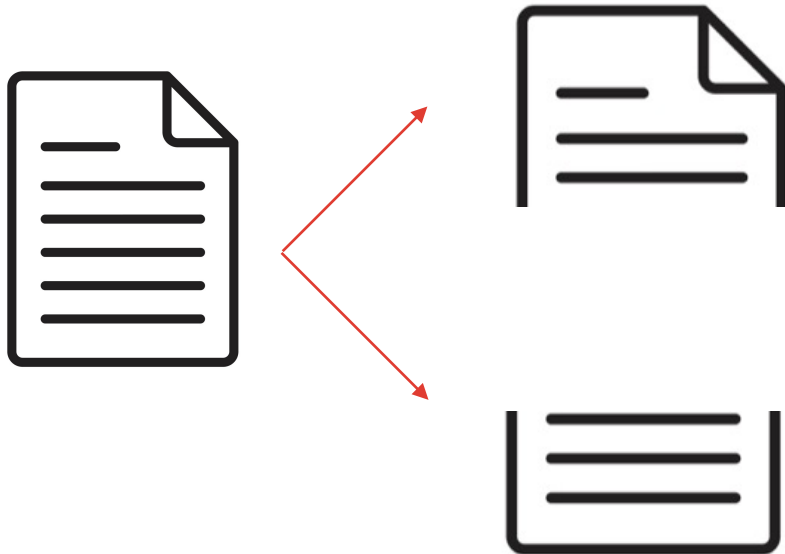
In een trainingsfase

# Trainingsfase

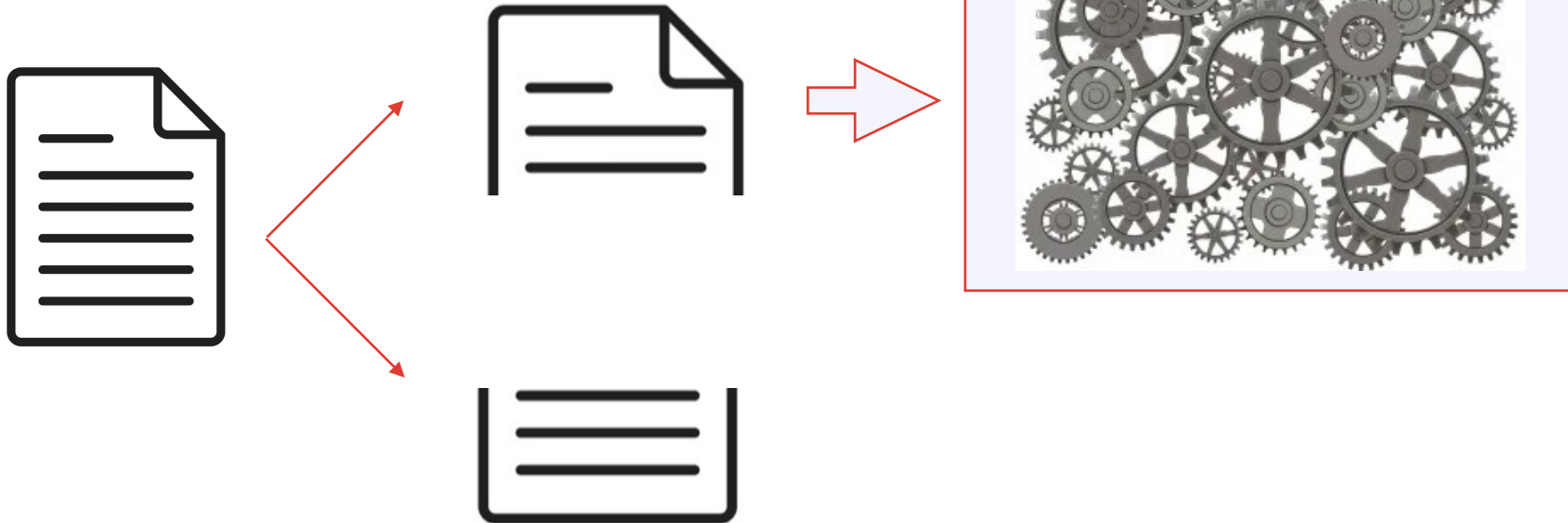




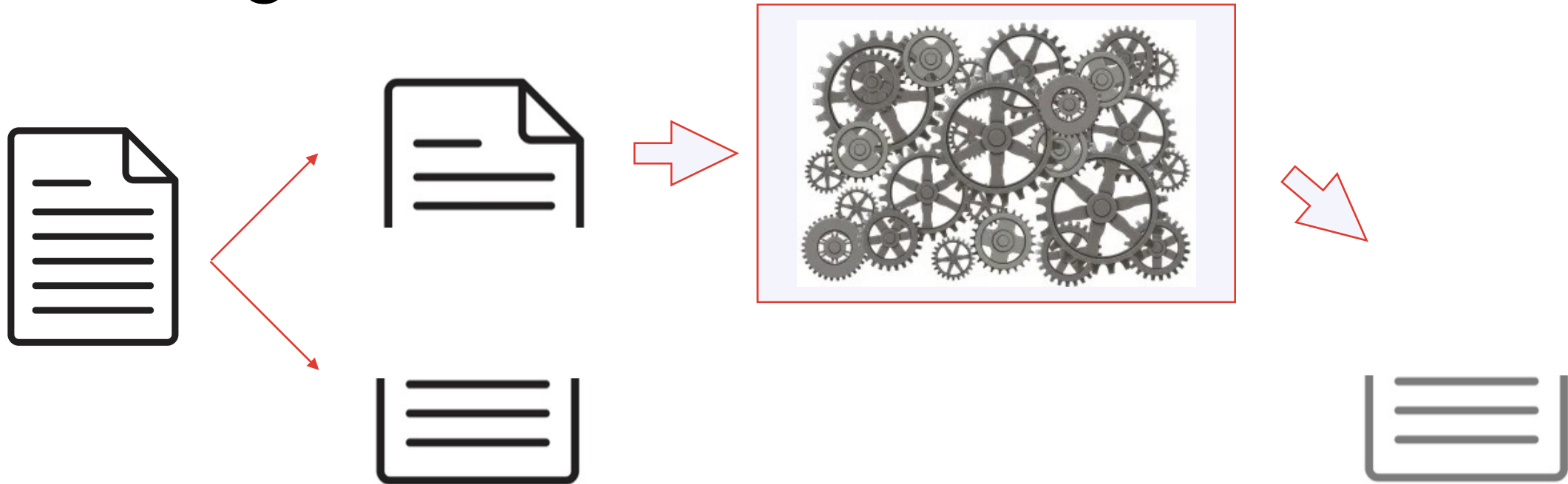
# Trainingsfase



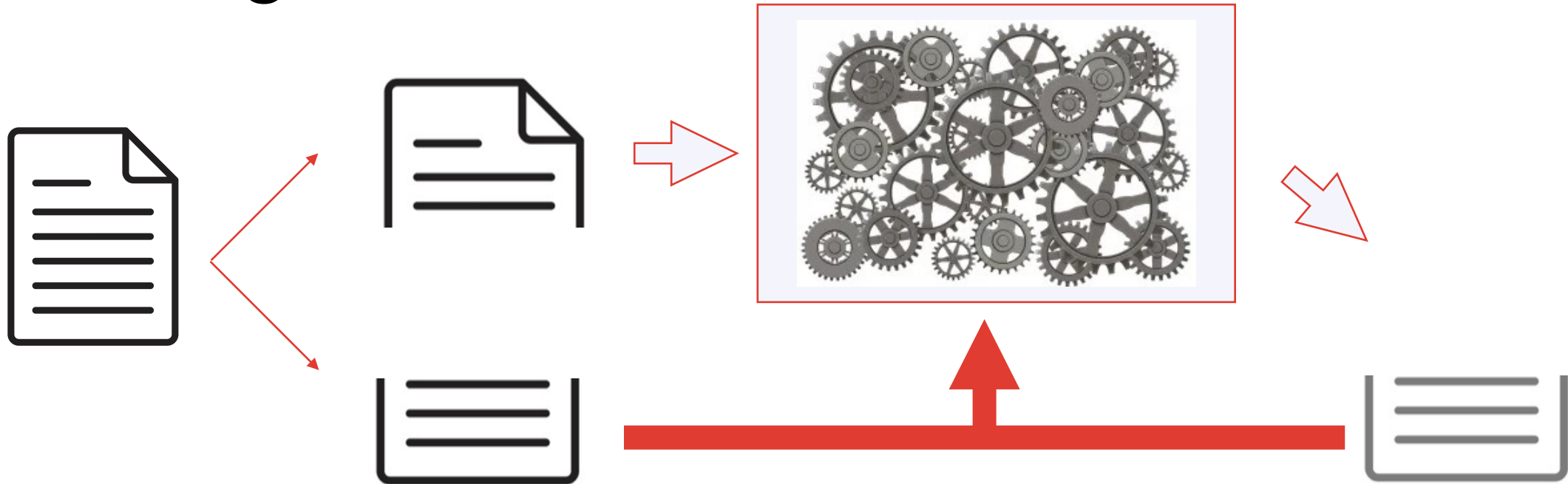
# Trainingsfase



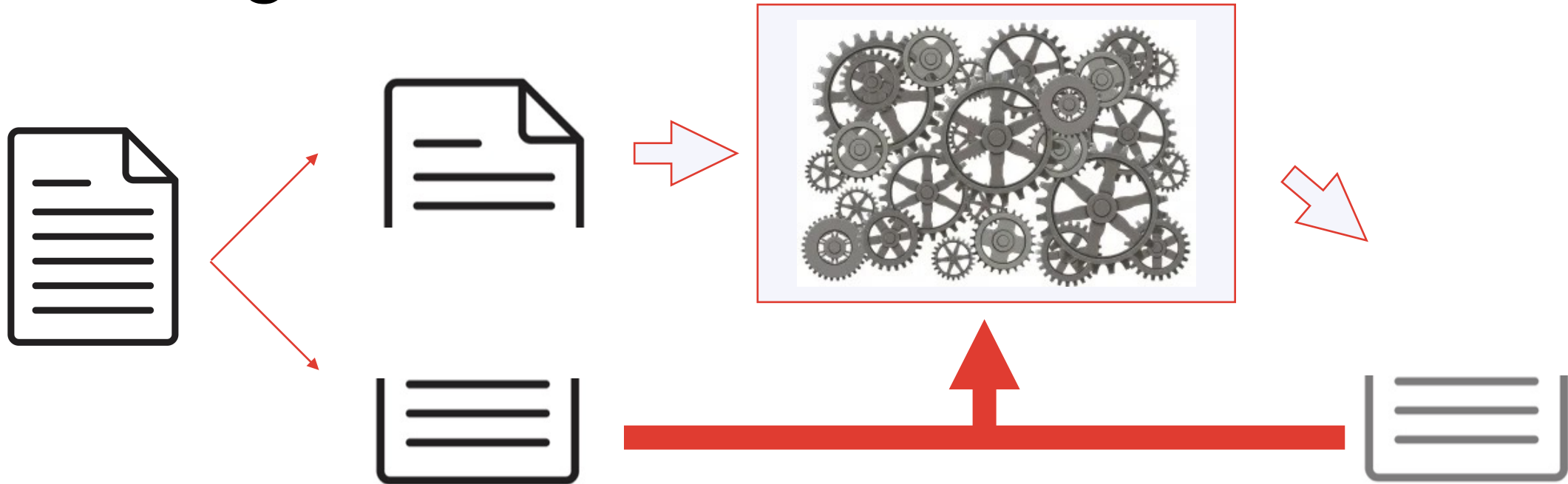
# Trainingsfase



# Trainingsfase

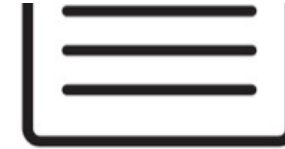
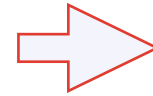
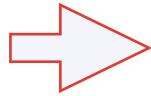


# Trainingsfase

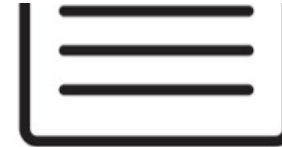
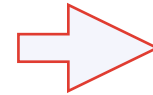
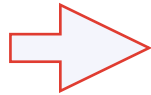


Wordt herhaald voor alle documenten op het Internet (Wikipedia, StackExchange, Github, ...)

# chatGPT vervolledigt documenten

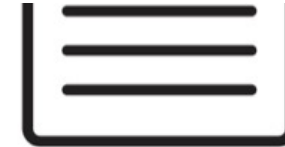
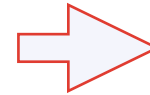
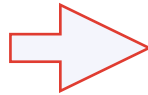


# chatGPT vervolledigt documenten



Wat is de hoofdstad van België?

# chatGPT vervolledigt documenten

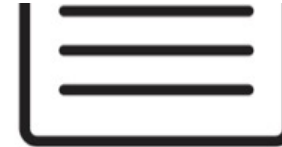
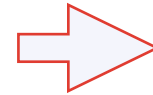
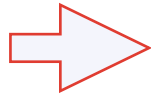


Wat is de hoofdstad van België?

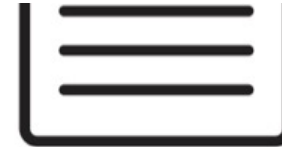
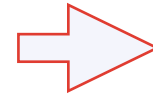
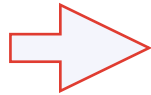
Wat is de hoofdstad van Nederland?  
Wat is de hoofdstad van Duitsland?  
Wat is de hoofdstad van Frankrijk?



# chatGPT vervolledigt documenten



# chatGPT vervolledigt documenten



Vraag: Wat is de hoofdstad van Nederland?

Antwoord: Amsterdam

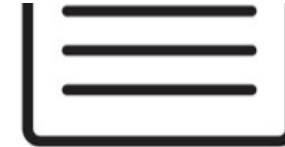
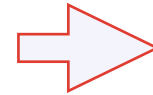
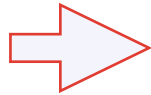
Vraag: Wat is de hoofdstad van Duitsland?

Antwoord: Berlijn

Vraag: Wat is de hoofdstad van België?

Antwoord:

# chatGPT vervolledigt documenten



Vraag: Wat is de hoofdstad van Nederland?

Antwoord: Amsterdam

Vraag: Wat is de hoofdstad van Duitsland?

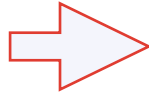
Antwoord: Berlijn

Vraag: Wat is de hoofdstad van België?

Antwoord:

Brussel

# chatGPT ombouwen naar een assistent in tweede fase van training



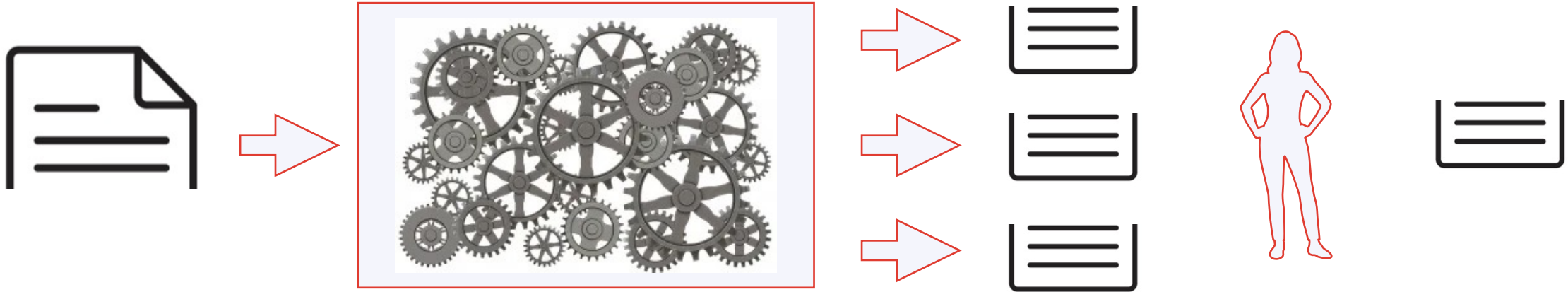
# chatGPT ombouwen naar een assistent in tweede fase van training



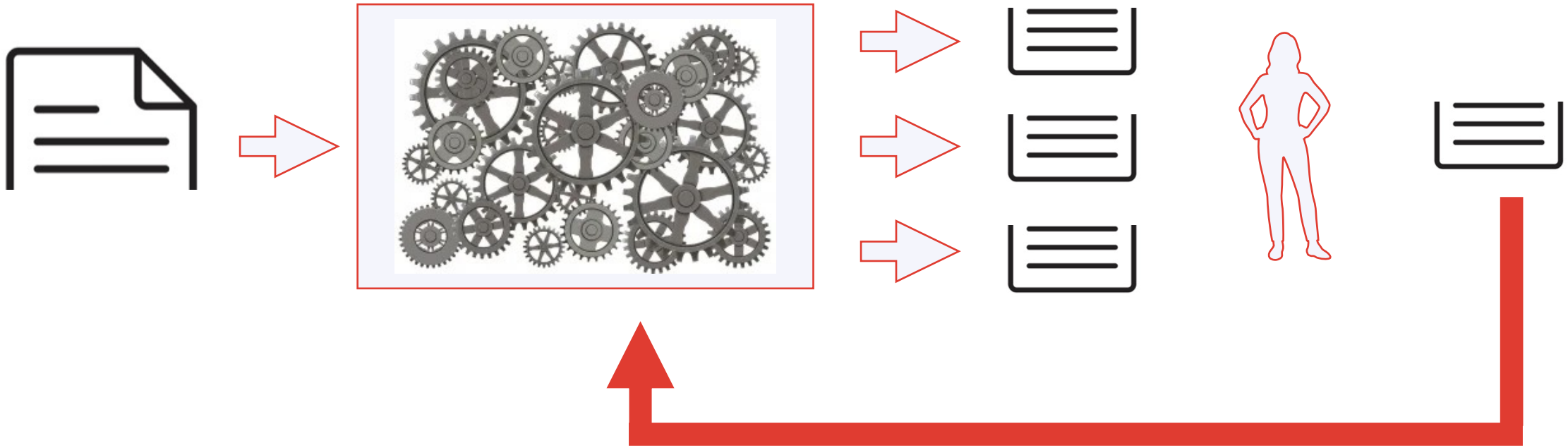
# chatGPT ombouwen naar een assistent in tweede fase van training



# chatGPT ombouwen naar een assistent in tweede fase van training



# chatGPT ombouwen naar een assistent in tweede fase van training





# Algoritme voor chatGPT

- Trainingsfase
  - Train chatGPT als een documentvervollediger op alle documenten van het Internet
  - Specialiseer chatGPT als assistent (om vragen te beantwoorden, opvolgen van instructies)
- Activeringsfase

# Waarom kan chatGPT deze instructie opvolgen?

F Schrijf een functie ggd die de grootste gemene deler van twee getallen berekent

```
def ggd(a, b):  
    while b != 0:  
        a, b = b, a % b  
    return a  
  
# Voorbeeldgebruik:  
x = 48  
y = 18  
print(f"De grootste gemene deler van {x} en {y} is {ggd(x, y)}")
```

# Waarom kan chatGPT deze instructie opvolgen?

F Schrijf een functie gcd die de grootste gemene deler van twee getallen berekent

```
def gcd(a, b):  
    while b != 0:  
        a, b = b, a % b  
    return a
```

# Voorbeeldgebruik:

x = 48

y = 18

```
print(f"De grootste gemene deler van {x} en {y} is {gcd(x, y)}")
```

Herhaald delen met rest kun je met Python uiteraard eenvoudig programmeren. Je begint met het tweetal  $(a, b)$  en als  $b \neq 0$ , dan pas je delen met rest toe:  $a = qb + r$ . Het tweetal wordt dan vervangen door  $(b, r)$ . Daar ga je mee door totdat het tweede getal van het tweetal 0 is. Het eerste is dan de grootste gemene deler. Dit wordt bereikt met de functie `gcd` in het bestand [arithmetics.py](#).

```
def gcd(a, b):  
    while b>0: a, b = b, a % b  
    return a
```

```
>>> gcd(1665,987)
```

```
3
```

```
>>> gcd(125788989999002,345367889990008)
```

```
2L
```

F

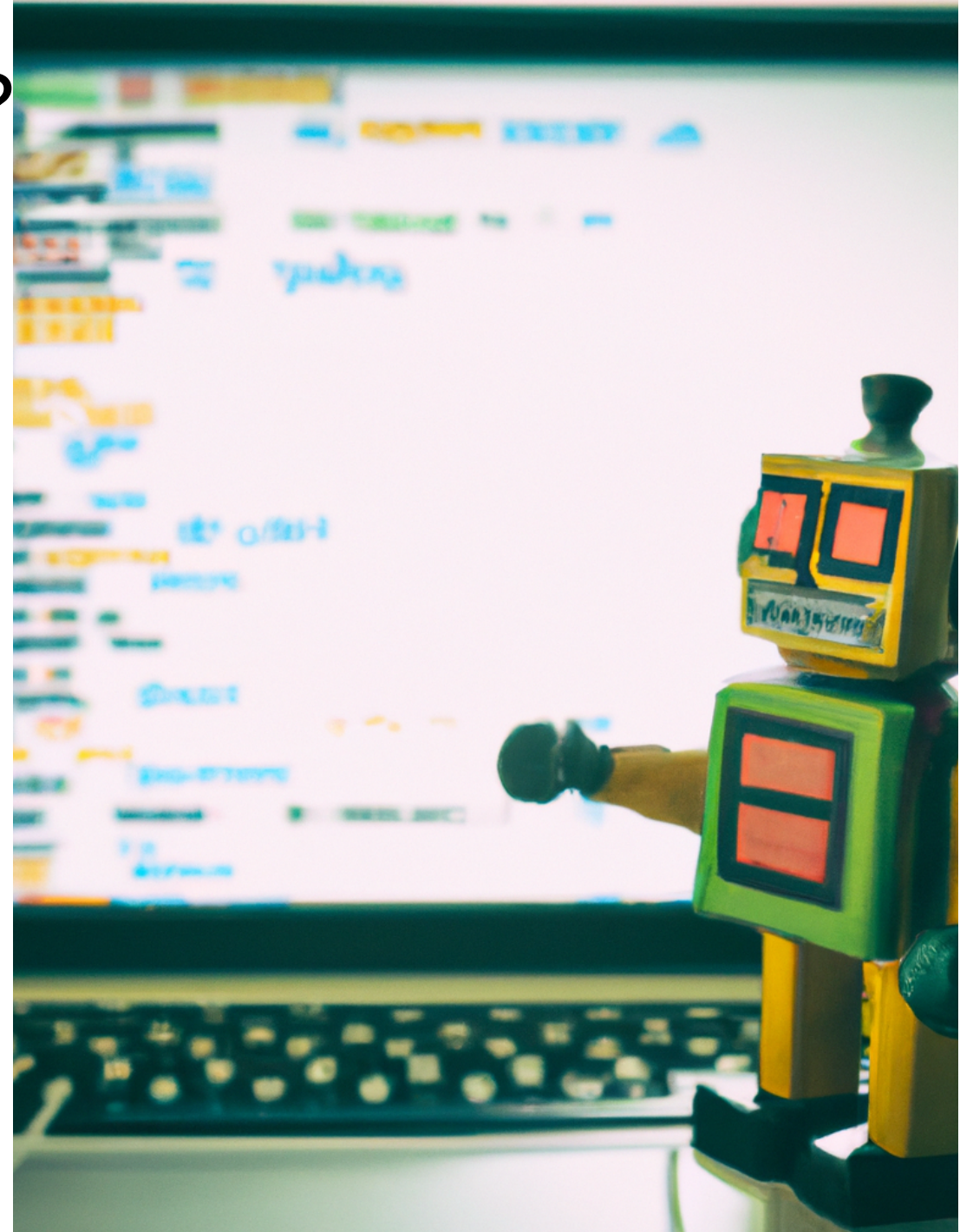
Genereer een scratch programma om de grootste gemene deler van twee getallen te berekenen.

Used **CreatiCode Scratch**

```
define find GCD of a b
  repeat until a = b
    if a > b then
      set a to a - b
    else
      set b to b - a
  return a
```

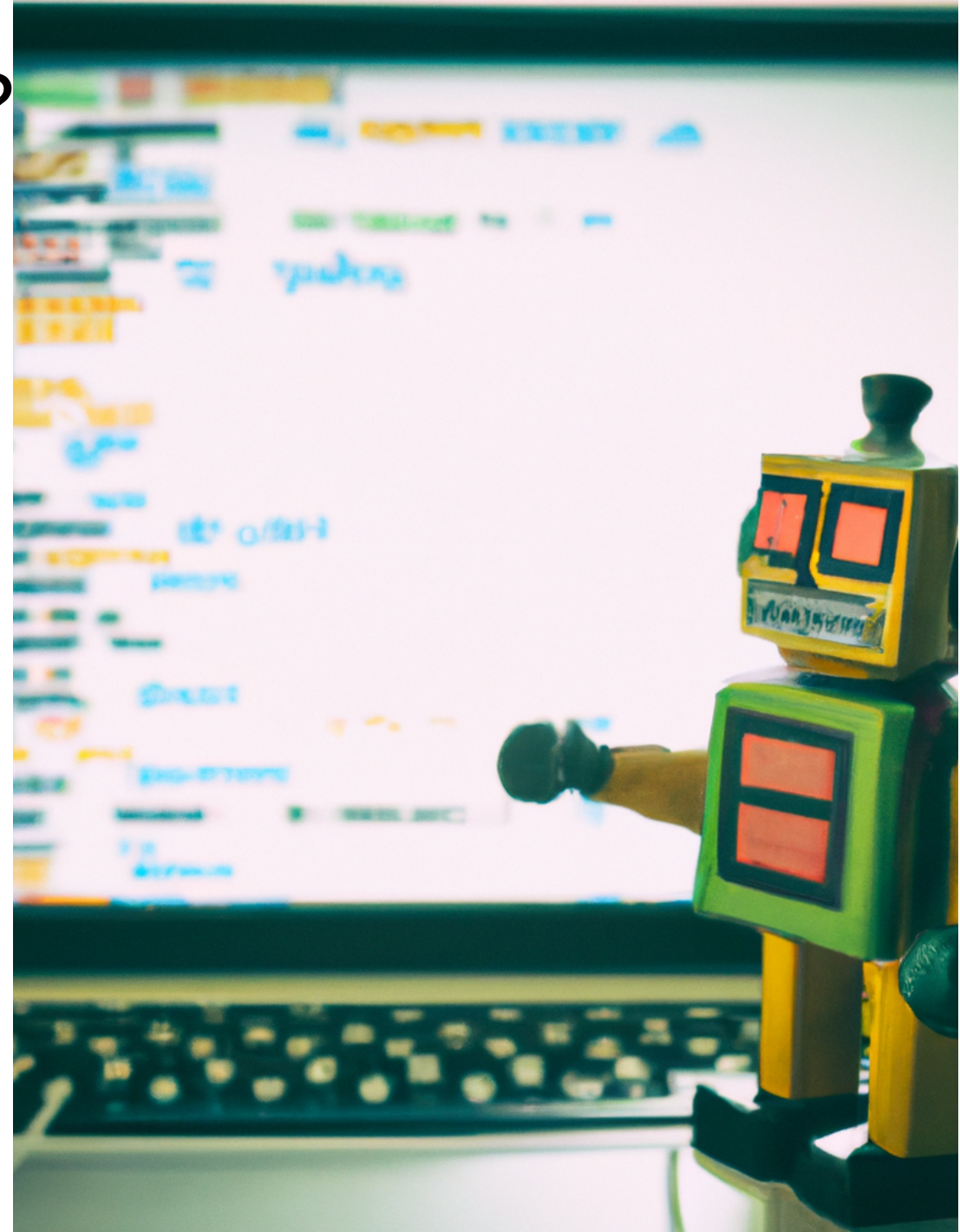
The image shows a Scratch script for finding the Greatest Common Divisor (GCD) of two numbers, a and b. The script is written in a Scratch-like block-based language. It starts with a 'define' block for a function named 'find GCD of' with two input parameters, 'a' and 'b'. The function body consists of a 'repeat until' loop with the condition 'a = b'. Inside the loop, there is an 'if' block that checks 'a > b'. If true, it sets 'a' to 'a - b'. If false, it sets 'b' to 'b - a'. After the loop, the function returns the value of 'a'.

# Kan chatGPT programmeren?



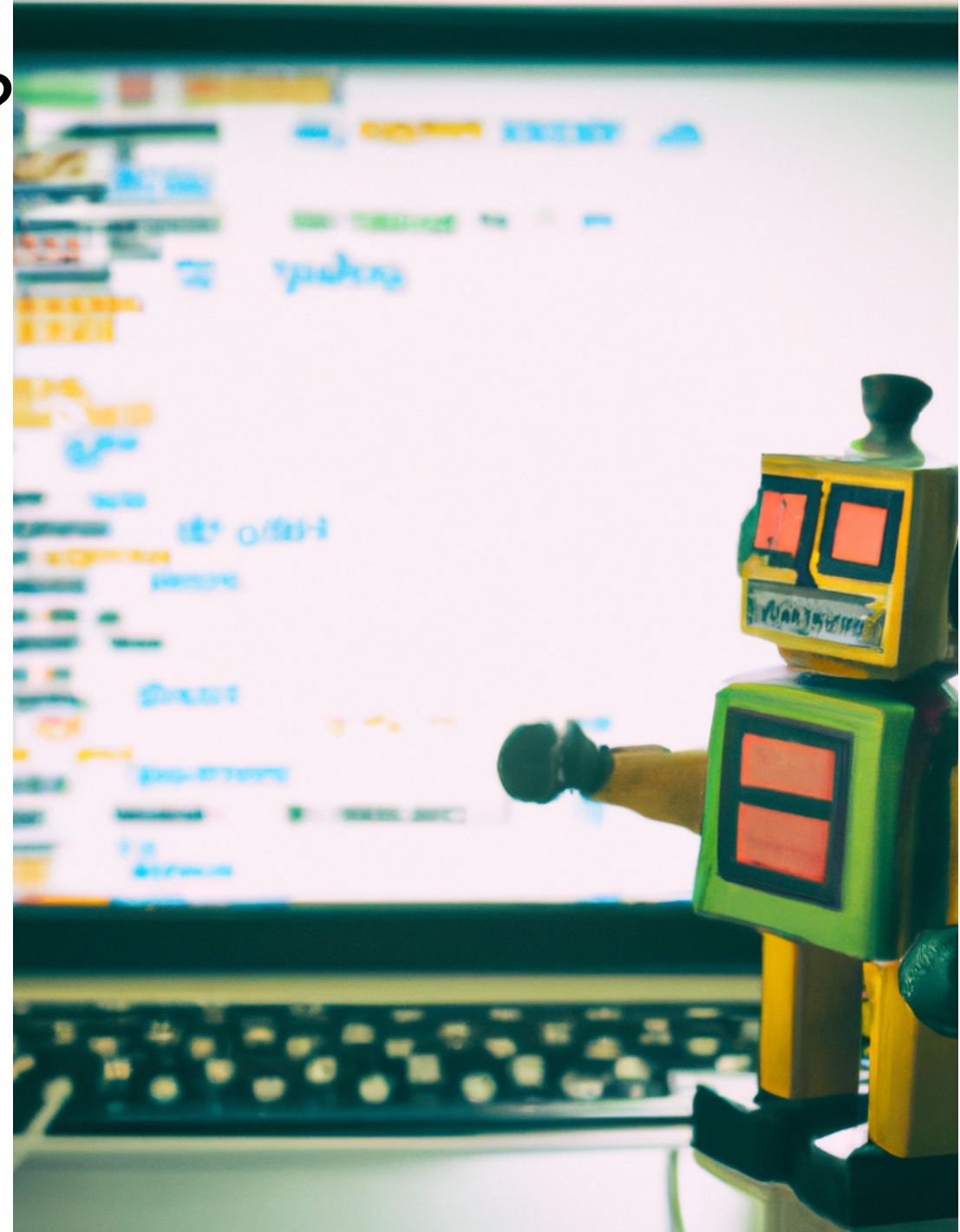
# Kan chatGPT programmeren?

- Zoals een mens dat kan? Nee.



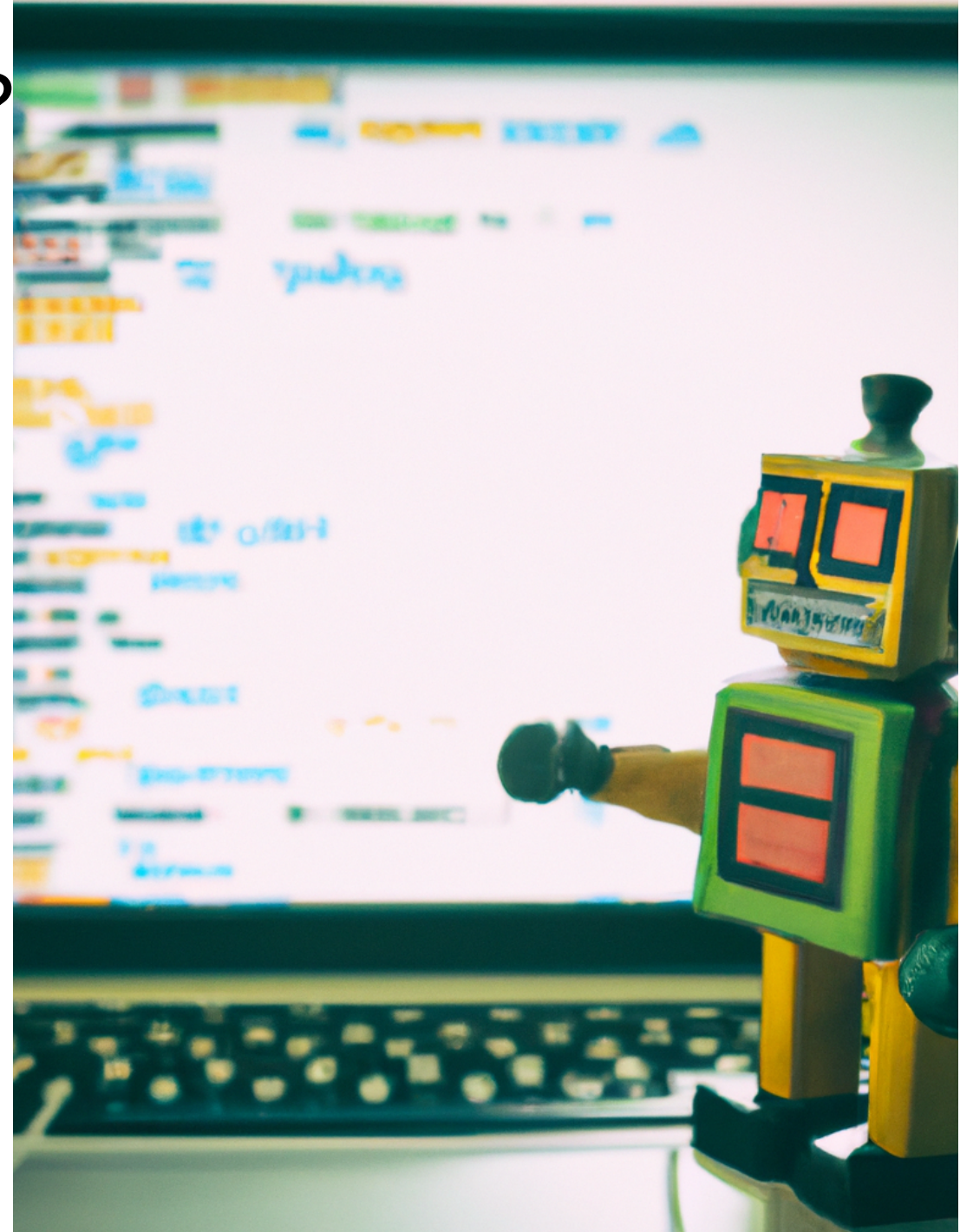
# Kan chatGPT programmeren?

- Zoals een mens dat kan? Nee.
- Een mens heeft begrip van programmeren, kan redeneren, maar steunt op (in vlg met chatGPT) beperkte ervaring.



# Kan chatGPT programmeren?

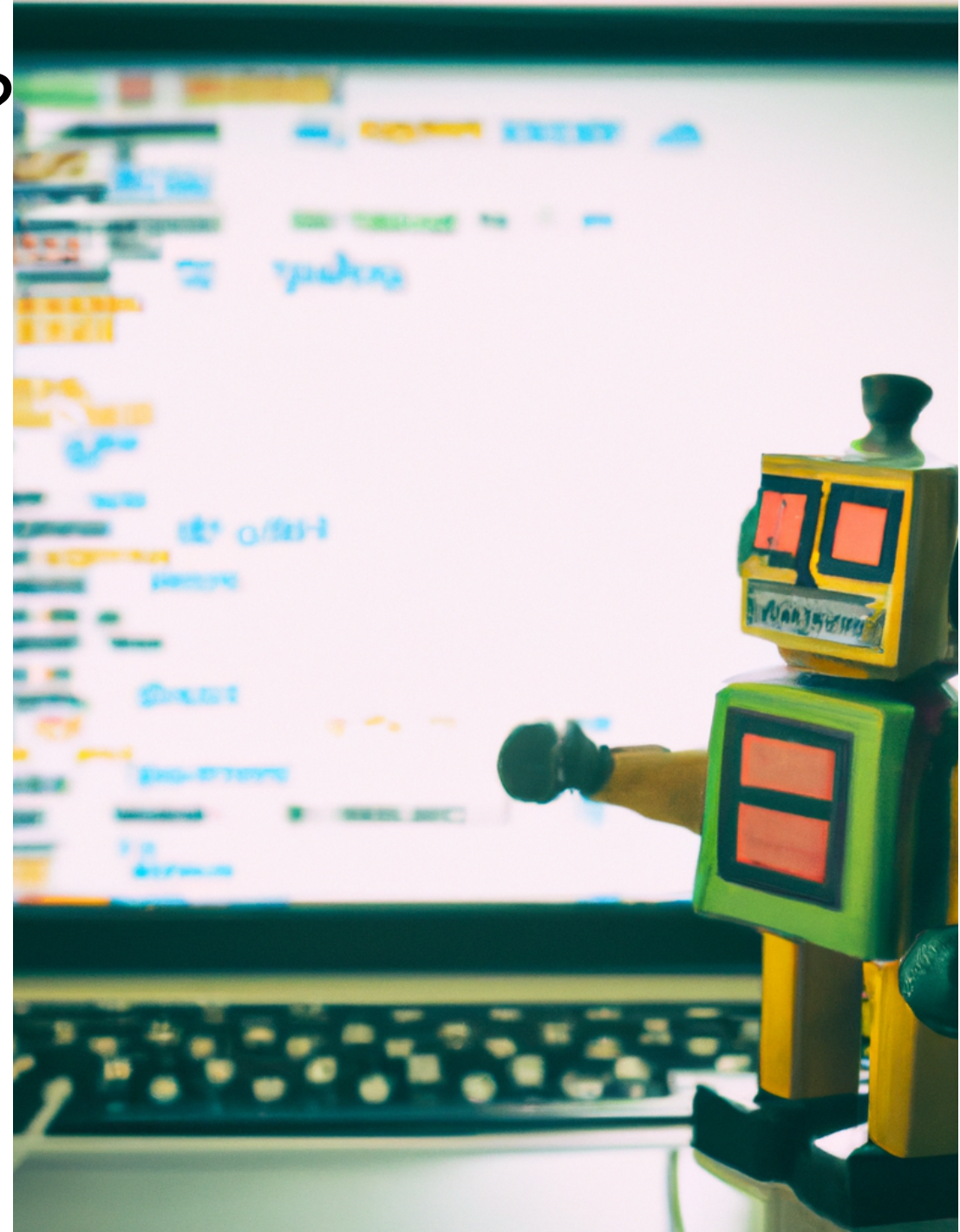
- Zoals een mens dat kan? Nee.
- Een mens heeft begrip van programmeren, kan redeneren, maar steunt op (in vlg met chatGPT) beperkte ervaring.
- chatGPT heeft enkel heel veel voorbeeldcode gezien en genereert op basis hiervan programma's (zonder begrip van die code)





# Kan chatGPT programmeren?

- Zoals een mens dat kan? Nee.
- Een mens heeft begrip van programmeren, kan redeneren, maar steunt op (in vlg met chatGPT) beperkte ervaring.
- chatGPT heeft enkel heel veel voorbeeldcode gezien en genereert op basis hiervan programma's (zonder begrip van die code)
- Gegenerateerde antwoorden kunnen fouten bevatten



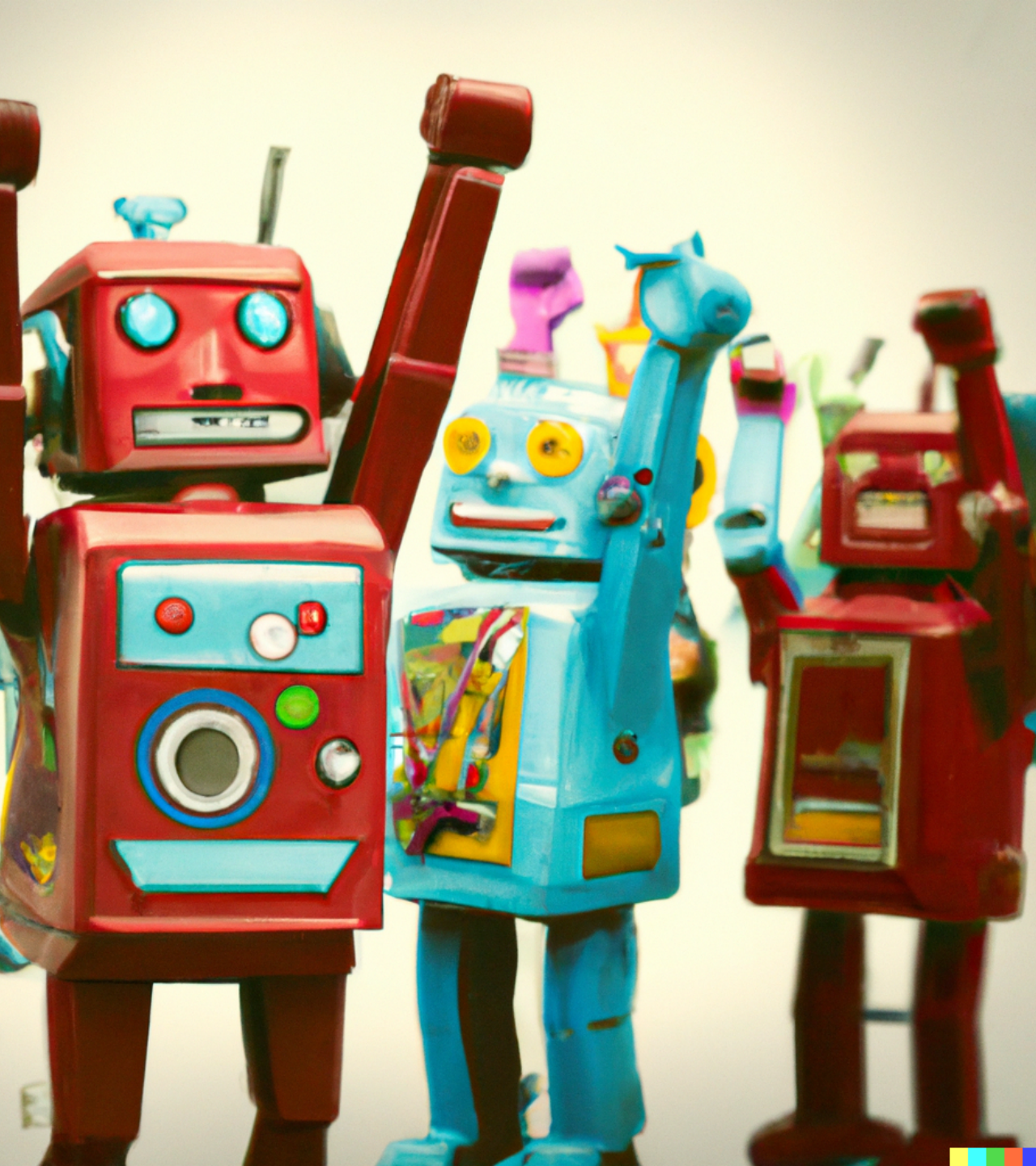
# Kan chatGPT programmeren?



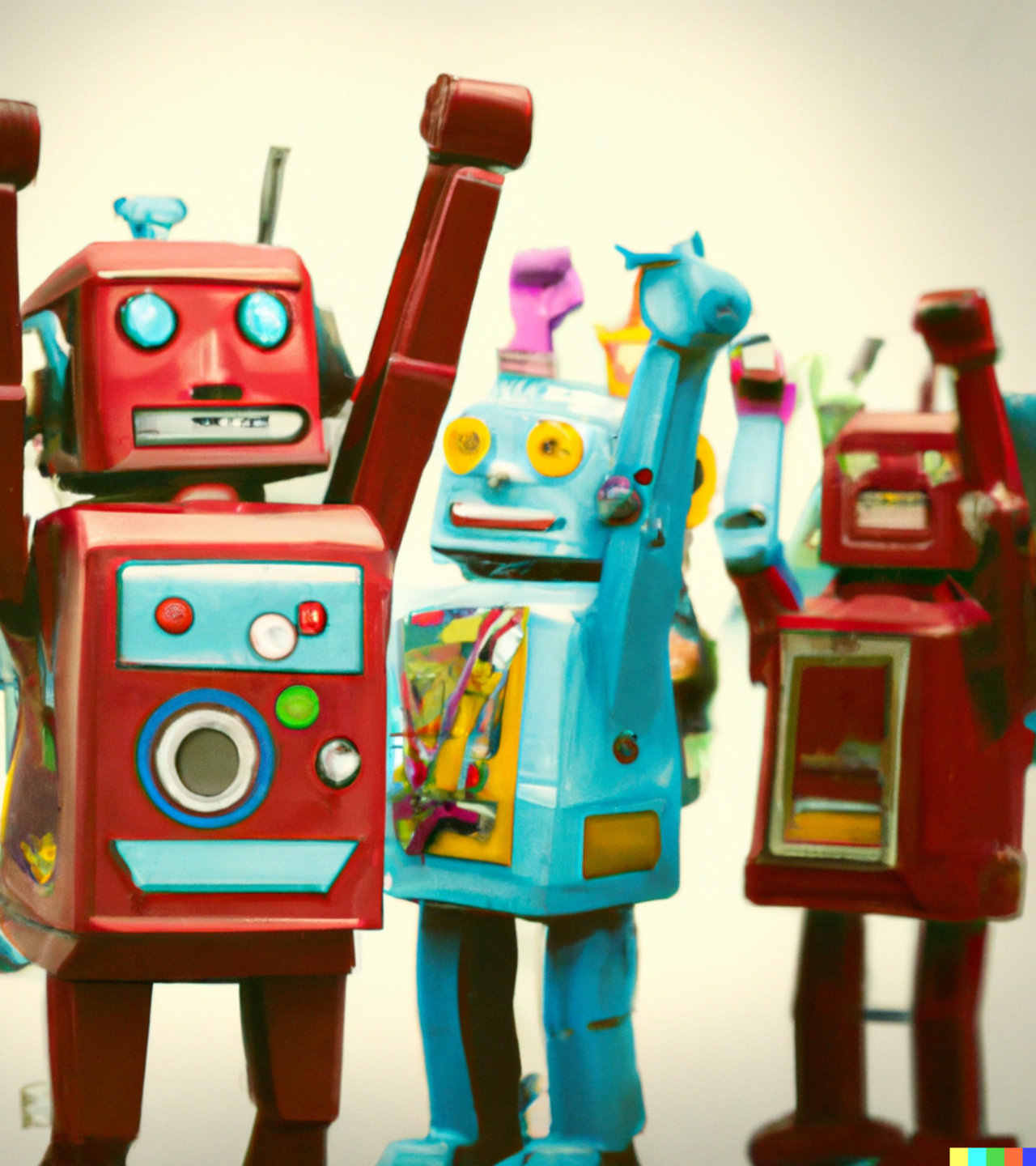
# Kan chatGPT programmeren?

- Maar chatGPT kan wel een goede **hulp** zijn voor de **kritische gebruiker**
  - Genereren van stappenplan
  - Code genereren voor deelproblemen
  - Schrijven van testen
  - Hulp om foutboodschappen te interpreteren
  - Inspiratie voor programmeerprojecten
  - ...





# Conclusie



# Conclusie

chatGPT maakt programmeren  
alleen maar relevanter

- programmeren wordt toegankelijker
- efficiëntie van programmeurs zal toenemen

# Kunnen computers programmeren?

Prof. dr. Frank Neven  
frank.neven@uhasselt.be



[WWW.UHASSELT.BE/DSI](http://WWW.UHASSELT.BE/DSI)

