

HO GENT

H2 Controlestructuren - Oefeningen

Table of Contents

1. Doelstellingen per oefening	1
2. Oefeningen	1
2.1. Stel voor met (geneste) if/else	1
2.2. Stel voor met een conditionele operator	1
2.3. Zet de volgende geneste if-structuren om in een switch-case - structuur	2
2.4. Verbeter volgend stukje code	2
2.5. Geef de inhoud van a, b en c na elk statement	2
2.6. Welke uitvoer geven volgende statements?	3
2.7. Welke output geven volgende applicaties?	3
2.8. Maak een for-herhalingsstructuur	4
2.9. Schrijf een applicatie gebruikmakend van een for-lus	4
2.10. Schrijf een applicatie gebruikmakend van een while-lus met schildwacht	6
2.11. Maak een do-while-herhalingsstructuur	7
2.12. Schrijf een applicatie gebruikmakend van een do-while - lus	7
2.13. Schrijf een applicatie gebruikmakend van een while-structuur	8
2.14. Welke uitvoer geven volgende statements?	8
2.15. Schrijf een applicatie die volgende patronen onder elkaar op het scherm zet, enkel gebruikmakend van	8
2.16. Herneem oefening 15 uit hoofdstuk 1 en breid uit met:	9
2.17. Schrijf een applicatie die het volgende doet:	9
2.18. Lees een postnummer in en druk de naam van de overeenkomstige stad af, aan de hand van volgende tabel:	10
2.19. Lees een afstand in als een geheel aantal kilometer. Moet strikt positief zijn! Controleer! ..	11
2.20. Werk een applicatie met een main-methode uit die juist 10 gehele getallen opvraagt aan de gebruiker en het op één na grootste hiervan bepaalt.	11
2.21. Werk een applicatie met een main-methode uit die aan de gebruiker een strikt positief geheel getal vraagt en controleert. Toon op het scherm alle delers van dit getal.	12
2.22. Lees een rij positieve gehele getallen in (controle op invoer!), afgesloten door het getal 0. Bepaal hoeveel van deze getallen door 2 deelbaar zijn, hoeveel door 3 en hoeveel door 6. Let op meervoud/enkelvoud bij de zinsbouw.	12
2.23. Werk een applicatie met een main-methode uit die een deler vraagt (strikt positief - controleer!) en een aantal positief gehele getallen. Stop deze invoer met -1.	13
2.24. Werk een applicatie met een main-methode uit die aan de gebruiker een strikt positief geheel getal vraagt en controleert.	14
2.25. Quotient en rest	15
2.26. Even of oneven?	15
2.27. Meerkeuze	16
2.28. Meerkeuze 2	16

2.29. Groter en dan kleiner	17
2.30. Priemgetallen	18

1. Doelstellingen per oefening

- Oefening 1: gebruikmaken van geneste selecties
- Oefening 2: gebruikmaken van de conditionele operator
- Oefening 3: gebruikmaken van een switch-case - structuur
- Oefening 4: code verbeteren
- Oefening 5: werken met verkorte operatoren
- Oefening 6: de uitvoer van statements met logische operatoren bepalen
- Oefening 7: de output van een applicatie met een lus bepalen
- Oefening 8: een for-herhalingsstructuur maken
- Oefening 9: een applicatie gebruikmakend van een for-lus schrijven
- Oefening 10: een applicatie gebruikmakend van een while-lus met schildwacht schrijven
- Oefening 11: een do-while - herhalingsstructuur maken
- Oefening 12: een applicatie gebruikmakend van een do-while - lus schrijven
- Oefening 13: een applicatie gebruikmakend van een herhalingsstructuur én een break-statement schrijven
- Oefening 14: de uitvoer van geneste for-lussen bepalen
- Oefening 15: een applicatie gebruikmakend van geneste for-lussen schrijven
- Oefening 16 - 18: de juiste selectiestructuur leren gebruiken
- Oefening 19 - 24: de juiste herhalingsstructuren leren gebruiken
- Oefening 25 - 26: de juiste selectiestructuur leren gebruiken
- Oefening 27 - 30: de juiste controlestructuren leren gebruiken

2. Oefeningen

2.1. Stel voor met (geneste) if/else

- als $x \geq 0$ dan druk "positief" anders druk "negatief"
- als $temp \leq 10$ dan druk "koud" als temp tussen 10 en 20 ligt dan druk "goed" als $temp \geq 20$ dan druk "warm"
- als $x \geq 0$ en even dan druk "positief en even" als $x \geq 0$ en oneven dan druk "positief en oneven" als $x < 0$ en even dan druk "negatief en even" als $x < 0$ en oneven dan druk "negatief en oneven"

2.2. Stel voor met een conditionele operator

```
if (aantal == 1)
```

```
        System.out.println("Student");
    else
        System.out.println("Studenten");
}
// is equivalent met
// ?
```

2.3. Zet de volgende geneste if-structuren om in een switch-case - structuur

- Oefening1

```
int i, k;
// invoer van een waarde voor i
if (i == 1)
    k = 3;
else if (i == 2) k = 6;
    else if ( i == 3 || i == 4) k = 10;
        else k = 20;
```

- Oefening2

```
int x, y = 0;
// invoer van een waarde voor x
if (x == 100 || x == 150 || x == 170 || x == 199) y = y + 1;
```

2.4. Verbeter volgend stukje code

```
if ( age >= 65 );
{ System.out.println ("Age greater than or equal to 65" );}
else
System.out.println ("Age is less than 65 ");
```

2.5. Geef de inhoud van a, b en c na elk statement

```
package werkcollege2;

public class Werkcollege_12
{
    public static void main(String[] args)
    {
        new Werkcollege_12().berekenIncrementEnDecrement();
    }
}
```

```

private void berekenIncrementEnDecrement()
{
    int a = 0, b = 0, c = 0;

    a = ++b + ++c;    // a =      b =      c =

    a = b++ + c++;    // a =      b =      c =

    a = ++b + c++;    // a =      b =      c =

    a = b-- + c--;    // a =      b =      c =
}
}

```

2.6. Welke uitvoer geven volgende statements?

Stel: i = 1, j = 2, k = 3 en m = 2

- `System.out.println(i == 1);`
- `System.out.println(j == 3);`
- `System.out.println(i >= 1 && j < 4);`
- `System.out.println(m <= 99 && k < m);`
- `System.out.println(j >= i || k == m);`
- `System.out.println(k + m < j || 3 - j >= k);`
- `System.out.println(!(k > m));`

2.7. Welke output geven volgende applicaties?

- Oefening1

```

package werkcollege2;

public class Mystery1
{
    public static void main(String[] args)
    {
        new Mystery1().callMysteryMethod();
    }

    private void callMysteryMethod()
    {
        int y, x = 1, total = 0;

        for( ;x <= 10; x = x + 1)
        {
            y = x * x;

```

```

        System.out.printf("%d%n", y);
        total = total + y;
    }

    System.out.printf("Total is %d%n", total);
}
}

```

- Oefening2

```

package werkcollege2;

public class Mystery2
{
    public static void main (String[] args)
    {
        new Mystery2().callMysteryMethod();
    }

    private void callMysteryMethod()
    {
        for (int count = 1; count <= 10 ; count = count +1)
        {
            System.out.println(
                count % 2 == 1 ? "****" : "++++++" );
        }
    }
}

```

2.8. Maak een for-herhalingsstructuur

- Oefening1: Alle gehele getallen tussen 1 en 500, die deelbaar zijn door 12, worden opgeteld.
- Oefening2: Alle positieve oneven getallen, beginnend bij 51 worden in dalende volgorde weergegeven.
- Oefening3: Het alfabet wordt weergegeven.

2.9. Schrijf een applicatie gebruikmakend van een for-lus

- Oefening1: Schrijf een applicatie die de som maakt van 5 in te lezen gehele getallen.

```
run:
Geef getal 1: 3
Geef getal 2: 6
Geef getal 3: -9
Geef getal 4: 2
Geef getal 5: 0
De som van de getallen = 2
BUILD SUCCESSFUL (total time: 13 seconds)
```

- Oefening2: Vraag aan de gebruiker 5 getallen en schrijf uit hoeveel getallen ervan even zijn en hoeveel getallen deelbaar zijn door 3.

```
run:
Geef getal 1 in: 900
Geef getal 2 in: 61
Geef getal 3 in: 20
Geef getal 4 in: 78
Geef getal 5 in: 12
Je gaf 4 even getallen en 3 getallen deelbaar door 3 in.
BUILD SUCCESSFUL (total time: 27 seconds)
```

```
run:
Geef getal 1 in: 12
Geef getal 2 in: 802
Geef getal 3 in: 610
Geef getal 4 in: 32
Geef getal 5 in: 14
Je gaf 5 even getallen en 1 getal deelbaar door 3 in.
BUILD SUCCESSFUL (total time: 22 seconds)
```


- Oefening3: Schrijf een applicatie die via een lus de volgende tabel met waarden afdruckt:

```
run:
N      10*N      100*N      1000*N
1       10       100       1000
2       20       200       2000
3       30       300       3000
4       40       400       4000
5       50       500       5000
BUILD SUCCESSFUL (total time: 0 seconds)
```

2.10. Schrijf een applicatie gebruikmakend van een while-lus met schildwacht

- Oefening1: Vraag aan de gebruiker een aantal gehele getallen, afgesloten door 0 (wordt niet meer verwerkt). Na de invoer komt het gemiddelde van de negatieve getallen op het scherm (1 cijfer na de komma). Werden er geen negatieve getallen ingegeven, dan verschijnt er een passende boodschap.

```
Geef een getal (0 om te stoppen): 0
Er werden geen negatieve getallen ingevoerd!
```

```
Geef een getal (0 om te stoppen): 2
Geef een getal (0 om te stoppen): 3
Geef een getal (0 om te stoppen): 6
Geef een getal (0 om te stoppen): 5
Geef een getal (0 om te stoppen): 0
Er werden geen negatieve getallen ingevoerd!
```

```
Geef een getal (0 om te stoppen): 2
Geef een getal (0 om te stoppen): -3
Geef een getal (0 om te stoppen): -6
Geef een getal (0 om te stoppen): 5
Geef een getal (0 om te stoppen): -1
Geef een getal (0 om te stoppen): 0
Het gemiddelde van alle negatieve getallen is -3,3
```

- Oefening2: Lees een aantal gehele getallen in, sluit af met 0 (wordt niet meer verwerkt). Vervolgens wordt het kleinste en het grootste getal weergegeven. Indien de gebruiker onmiddellijk nul ingeeft, moet er een foutboodschap verschijnen.

```
run:
Geef een getal (0 om te stoppen): 0
Er werden geen geldige getallen ingevoerd!
BUILD SUCCESSFUL (total time: 2 seconds)
```

```
run:
Geef een getal (0 om te stoppen): 3
Geef een getal (0 om te stoppen): 6
Geef een getal (0 om te stoppen): 5
Geef een getal (0 om te stoppen): 2
Geef een getal (0 om te stoppen): 9
Geef een getal (0 om te stoppen): 4
Geef een getal (0 om te stoppen): 12
Geef een getal (0 om te stoppen): 5
Geef een getal (0 om te stoppen): 0
Het grootste van alle ingevoerde getallen is 12.
Het kleinste is 2.
BUILD SUCCESSFUL (total time: 21 seconds)
```

2.11. Maak een do-while-herhalingsstructuur

- Oefening1: Er wordt gevraagd om een strikt negatief oneven getal, in te voeren; de invoer wordt vervolgens gecontroleerd; indien deze niet voldoet, wordt een nieuwe waarde aan de gebruiker gevraagd.
- Oefening2: Er wordt gevraagd om twee getallen in te geven. Het eerste getal moet verschillend van 1000 en mag niet deelbaar zijn door 12. Het tweede getal moet groter zijn dan het eerste getal. Indien de invoer niet voldoet, wordt een nieuwe waarde aan de gebruiker gevraagd.

2.12. Schrijf een applicatie gebruikmakend van een do-while - lus

In een afgesloten natuurpark zijn er 50 leeuwen aanwezig. Het aantal groeit jaarlijks met 15% aan.

Hoelang duurt het tot er meer dan 1000 leeuwen zijn?

2.13. Schrijf een applicatie gebruikmakend van een while-structuur

Lees een aantal gehele getallen in. De invoer stopt als de gebruiker -1 ingeeft (-1 wordt niet meer verwerkt) of als de gebruiker reeds 10 getallen heeft ingegeven.

Het gemiddelde van de ingegeven getallen verschijnt op het scherm of een melding als er geen getallen werden ingegeven.

2.14. Welke uitvoer geven volgende statements?

- Oefening1

```
for (int i = 1; i <= 10; i++)
{
    for (int j = 1; j <= 5; j++)
        System.out.print('@');

    System.out.println();
}
```

- Oefening2

```
for (int i = 1; i <= 5; i++)
{
    for (int j = 1; j <= 3; j++)
    {
        for (int k = 1; k <= 4; k++)
        {
            System.out.print('*');
        }
        System.out.println();
    }
    System.out.println();
}
```

2.15. Schrijf een applicatie die volgende patronen onder elkaar op het scherm zet, enkel gebruikmakend van

- `System.out.print("*")`
- `System.out.println()`
- Alleen bij (B) en (D): `System.out.print(" ")`

(A)

(B)

(C)

(D)

★	★★★★★★★★	★★★★★★★★★	★
★★	★★★★★★★★	★★★★★★★★	★★
★★★	★★★★★★★★	★★★★★★★★	★★★
★★★★	★★★★★★★★	★★★★★★	★★★★
★★★★★	★★★★★★	★★★★★	★★★★★
★★★★★★	★★★★★	★★★★	★★★★★★
★★★★★★★	★★★★	★★★★	★★★★★★★
★★★★★★★★	★★★	★★★	★★★★★★★★
★★★★★★★★★	★★	★★	★★★★★★★★★
★★★★★★★★★★	★	★	★★★★★★★★★

2.16. Herneem oefening 15 uit hoofdstuk 1 en breid uit met:

Schrijf een applicatie die aan de gebruiker drie gehele getallen vraagt en daarvan OOK het grootste getal weergeeft. Bepaal het grootste getal zo performant mogelijk!

Geef eerste getal in: 6

Geef tweede getal in: 3

Geef derde getal in: 12

Van de ingevoerde getallen 6, 3 en 12

is de som 21

het gemiddelde 7

de rest 0

en het grootste getal 12

2.17. Schrijf een applicatie die het volgende doet:

- Vraag een geheel getal aan de gebruiker.
- Als dit getal strikt positief is, trek er dan 10 van af. Als dit getal negatief is, vermeerder het dan met 10. Als dit getal gelijk aan nul is, vermeerder het met 1.
- Schrijf in een volzin uit wat er gebeurd is. De uitvoer kan dus zijn: Het ingegeven getal was strikt positief en werd verminderd met 10. Het heeft nu de waarde

Geef een getal in: 6

Het ingegeven getal was strikt positief en werd verminderd met 10.
Het heeft nu de waarde -4.

Of

Het ingegeven getal was negatief en werd vermeerderd met 10. Het heeft nu de waarde

Geef een getal in: -23

Het ingegeven getal was negatief en werd vermeerderd met 10.
Het heeft nu de waarde -13.

Of

Het ingegeven getal was nul en werd vermeerderd met 1.

Geef een getal in: 0

Het ingegeven getal was nul en werd vermeerderd met 1.

2.18. Lees een postnummer in en druk de naam van de overeenkomstige stad af, aan de hand van volgende tabel:

Postnummer	Naam stad
9300	Aalst
2000	Antwerpen
1000	Brussel
9200	Dendermonde
9000	Gent
8500	Kortrijk
9700	Oudenaarde
2300	Turnhout

Geef een postcode (4 cijfers): 12345

Geef een postcode (4 cijfers): 8500

Postnummer 8500 komt overeen met de stad Kortrijk

Het programma geeft een foutmelding bij verkeerde of niet in de tabel voorkomende postnummers:

Geef een postcode (4 cijfers): 1234

Postnummer 1234 bestaat niet of komt overeen met een stad die niet in de tabel is opgenomen

2.19. Lees een afstand in als een geheel aantal kilometer. Moet strikt positief zijn! Controleer!

Druk een tabel af, bestaande uit drie kolommen. Kolom 1 bevat de afstand, kolom 2 bevat de snelheid, variërend van 40 km/u, 50 km/u, 60 km/u tot en met 140 km/u. In de derde kolom staat de tijd (in uren en minuten) die nodig is om die afstand af te leggen aan die bepaalde snelheid. Zorg voor een identiek geformatteerde uitvoer!

```
Geef een afstand in kilometer (strikt positief geheel getal): 0
Geef een afstand in kilometer (strikt positief geheel getal): -6
Geef een afstand in kilometer (strikt positief geheel getal): 35
| 35 km      40 km/u      0 u 52 min
  35 km      50 km/u      0 u 42 min
  35 km      60 km/u      0 u 35 min
  35 km      70 km/u      0 u 30 min
  35 km      80 km/u      0 u 26 min
  35 km      90 km/u      0 u 23 min
  35 km     100 km/u      0 u 21 min
  35 km     110 km/u      0 u 19 min
  35 km     120 km/u      0 u 17 min
  35 km     130 km/u      0 u 16 min
  35 km     140 km/u      0 u 15 min
```

2.20. Werk een applicatie met een main-methode uit die juist 10 gehele getallen opvraagt aan de gebruiker en het op één na grootste hiervan bepaalt.

```
Geef getal 1 in: 1
Geef getal 2 in: 10
Geef getal 3 in: 2
Geef getal 4 in: 20
Geef getal 5 in: 13
Geef getal 6 in: 5
Geef getal 7 in: -6
Geef getal 8 in: 0
Geef getal 9 in: 9
Geef getal 10 in: -2
|
Het op één na grootste getal is 13
```

2.21. Werk een applicatie met een main-methode uit die aan de gebruiker een strikt positief geheel getal vraagt en controleert. Toon op het scherm alle delers van dit getal.

Hint: de delers zijn nooit groter dan de helft van het getal (behalve het getal zelf)

```
Geef een strikt positief geheel getal in: 28
De delers zijn: 1    2    4    7   14 en 28
```

2.22. Lees een rij positieve gehele getallen in (controle op invoer!), afgesloten door het getal 0. Bepaal hoeveel van deze getallen door 2 deelbaar zijn, hoeveel door 3 en hoeveel door 6. Let op meervoud/enkelvoud bij de zinsbouw.

Merk op: een getal dat deelbaar is door 6 is OOK deelbaar door 2 én door 3.

```
Geef een positief geheel getal (stoppen met 0): 6
Geef een positief geheel getal (stoppen met 0): 9
Geef een positief geheel getal (stoppen met 0): 12
Geef een positief geheel getal (stoppen met 0): 3
Geef een positief geheel getal (stoppen met 0): 28
Geef een positief geheel getal (stoppen met 0): 14
Geef een positief geheel getal (stoppen met 0): 80
Geef een positief geheel getal (stoppen met 0): 0
Er zijn 5 getallen deelbaar door 2
Er zijn 4 getallen deelbaar door 3
Er zijn 2 getallen deelbaar door 6
```

```
Geef een positief geheel getal (stoppen met 0): 6
Geef een positief geheel getal (stoppen met 0): 17
Geef een positief geheel getal (stoppen met 0): 11
Geef een positief geheel getal (stoppen met 0): 5
Geef een positief geheel getal (stoppen met 0): 0
Er is 1 getal deelbaar door 2
Er is 1 getal deelbaar door 3
Er is 1 getal deelbaar door 6
```

2.23. Werk een applicatie met een main-methode uit die een deler vraagt (strikt positief - controleer!) en een aantal positief gehele getallen. Stop deze invoer met -1.

Bepaal hoeveel getallen deelbaar zijn door dé deler en druk dit af. Denk hierbij aan enkelvoud/meervoud.

```
Geef een strikt positieve deler in: -6
Geef een strikt positieve deler in: 0
Geef een strikt positieve deler in: 3
Geef positief getal 1 in (of stop met -1): -1
|
Er zijn 0 getallen deelbaar door 3
```

```
Geef een strikt positieve deler in: 3
Geef positief getal 1 in (of stop met -1): 22
Geef positief getal 2 in (of stop met -1): 8
Geef positief getal 3 in (of stop met -1): 5
Geef positief getal 4 in (of stop met -1): 10
Geef positief getal 5 in (of stop met -1): 9
Geef positief getal 6 in (of stop met -1): -1
|
Er is 1 getal deelbaar door 3
```



```
Geef een strikt positieve deler in: 3
Geef positief getal 1 in (of stop met -1): 12
Geef positief getal 2 in (of stop met -1): 28
Geef positief getal 3 in (of stop met -1): 10
Geef positief getal 4 in (of stop met -1): 21
Geef positief getal 5 in (of stop met -1): 9
Geef positief getal 6 in (of stop met -1): -1
```

Er zijn 3 getallen deelbaar door 3

2.24. Werk een applicatie met een main-methode uit die aan de gebruiker een strikt positief geheel getal vraagt en controleert.

Als het even is, deel het getal door 2.

Als het oneven is, vermenigvuldig het getal met 3 en tel er 1 bij op. Zo bekom je een nieuw getal.

Herhaal deze bewerkingen op het nieuwe getal tot je 1 bekomt.

Toon op het scherm hoelang het duurt, m.a.w hoeveel keer het getal wijzigt tot het gelijk is aan 1.

```
Geef een strikt positief geheel getal in: 2
Het getal wijzigt 1 keer
```

```
Geef een strikt positief geheel getal in: 1
Het getal wijzigt 0 keren
```

```
Geef een strikt positief geheel getal in: 9
Het getal wijzigt 19 keren
```

2.25. Quotient en rest

Schrijf een applicatie die aan de gebruiker twee gehele getallen (teller en noemer) vraagt en daarvan het resultaat van de gehele deling, de rest en als de rest nul is de vereenvoudigde breuk weergeeft.

```
Geef de teller in van de breuk: -12
```

```
Geef de noemer in van de breuk: 3
```

```
-12/3 = -4
```

```
rest = 0
```

```
vereenvoudigde breuk = -4 / 1
```

```
Geef de teller in van de breuk: 12
```

```
Geef de noemer in van de breuk: 5
```

```
12/5 = 2
```

```
rest = 2
```

2.26. Even of oneven?

Schrijf een applicatie die aan de gebruiker een geheel getal vraagt en controleert of het getal even of oneven is. Geef een duidelijke boodschap weer.

```
Geef een geheel getal in: 22
```

```
Het ingevoerde getal 22 is een even getal
```

```
Geef een geheel getal in: 13
```

```
Het ingevoerde getal 13 is een oneven getal
```

2.27. Meerkeuze

Schrijf een applicatie die van 15 meerkeuzevragen de antwoorden inleest. De mogelijke antwoorden zijn 1, 2, 3 of 4. Geef duidelijk aan in de vraagstelling van welke vraag je het antwoord invoert. Een foutieve invoer resulteert in een foutboodschap. Schrijf op het einde uit hoeveel keer de respectievelijke antwoorden aan bod kwamen.

```
Geef antwoord 1 in (1, 2, 3 of 4): 1
Geef antwoord 2 in (1, 2, 3 of 4): 2
Geef antwoord 3 in (1, 2, 3 of 4): 3
Geef antwoord 4 in (1, 2, 3 of 4): 4
Geef antwoord 5 in (1, 2, 3 of 4): 5
Foutieve waarde! Probeer opnieuw!
Geef antwoord 5 in (1, 2, 3 of 4): 4
Geef antwoord 6 in (1, 2, 3 of 4): 3
Geef antwoord 7 in (1, 2, 3 of 4): 2
Geef antwoord 8 in (1, 2, 3 of 4): 1
Geef antwoord 9 in (1, 2, 3 of 4): 0
Foutieve waarde! Probeer opnieuw!
Geef antwoord 9 in (1, 2, 3 of 4): -5
Foutieve waarde! Probeer opnieuw!
Geef antwoord 9 in (1, 2, 3 of 4): 3
Geef antwoord 10 in (1, 2, 3 of 4): 3
Geef antwoord 11 in (1, 2, 3 of 4): 2
Geef antwoord 12 in (1, 2, 3 of 4): 1
Geef antwoord 13 in (1, 2, 3 of 4): 5
Foutieve waarde! Probeer opnieuw!
Geef antwoord 13 in (1, 2, 3 of 4): 4
Geef antwoord 14 in (1, 2, 3 of 4): 4
Geef antwoord 15 in (1, 2, 3 of 4): 1
Aantal 1 = 4
Aantal 2 = 3
Aantal 3 = 4
Aantal 4 = 4
```

2.28. Meerkeuze 2

Herneem oefening 28 maar het aantal vragen is niet vooraf gekend. Stop de invoer met 0. Geef duidelijk aan in de vraagstelling van welke vraag je het antwoord invoert. Een foutieve invoer resulteert in een foutboodschap.

```
Geef antwoord 1 in (1, 2, 3 of 4), STOP met 0: 1
Geef antwoord 2 in (1, 2, 3 of 4), STOP met 0: 2
Geef antwoord 3 in (1, 2, 3 of 4), STOP met 0: 3
Geef antwoord 4 in (1, 2, 3 of 4), STOP met 0: 4
Geef antwoord 5 in (1, 2, 3 of 4), STOP met 0: 5
Foutieve waarde! Probeer opnieuw!
Geef antwoord 5 in (1, 2, 3 of 4), STOP met 0: 2
Geef antwoord 6 in (1, 2, 3 of 4), STOP met 0: 1
Geef antwoord 7 in (1, 2, 3 of 4), STOP met 0: 1
Geef antwoord 8 in (1, 2, 3 of 4), STOP met 0: 3
Geef antwoord 9 in (1, 2, 3 of 4), STOP met 0: -5
Foutieve waarde! Probeer opnieuw!
Geef antwoord 9 in (1, 2, 3 of 4), STOP met 0: 2
Geef antwoord 10 in (1, 2, 3 of 4), STOP met 0: 0
Aantal 1 = 3
Aantal 2 = 3
Aantal 3 = 2
Aantal 4 = 1
```

2.29. Groter en dan kleiner

Schrijf een applicatie die een aantal gehele getallen inleest; sluit af met -2000 (wordt NIET meer verwerkt) en bepaal hoe vaak een groter getal DIRECT gevolgd wordt door een kleiner getal.

```
Geef een geheel getal (-2000 om te stoppen): 5
Geef een geheel getal (-2000 om te stoppen): -6
Geef een geheel getal (-2000 om te stoppen): 2
Geef een geheel getal (-2000 om te stoppen): 45
Geef een geheel getal (-2000 om te stoppen): 23
Geef een geheel getal (-2000 om te stoppen): 12
Geef een geheel getal (-2000 om te stoppen): 47
Geef een geheel getal (-2000 om te stoppen): 47
Geef een geheel getal (-2000 om te stoppen): -2000
Het aantal getallen dat direct gevolgd wordt door een kleiner getal is 3
```

```
Geef een geheel getal (-2000 om te stoppen): -2000
Het aantal getallen dat direct gevolgd wordt door een kleiner getal is 0
```

2.30. Priemgetallen

Schrijf een applicatie die een aantal gehele getallen inleest; sluit af met 0 (wordt NIET meer verwerkt) en bepaal hoe vaak een priemgetal wordt ingelezen.

```
Geef een geheel getal (0 om te stoppen): 1
1 is GEEN priemgetal!
Geef een geheel getal (0 om te stoppen): 7
7 is EEN priemgetal!
Geef een geheel getal (0 om te stoppen): 20
20 is GEEN priemgetal!
Geef een geheel getal (0 om te stoppen): -13
-13 is GEEN priemgetal!
Geef een geheel getal (0 om te stoppen): 0
Het aantal priemgetallen is 1
```

```
Geef een geheel getal (0 om te stoppen): 4
4 is GEEN priemgetal!
Geef een geheel getal (0 om te stoppen): 20
20 is GEEN priemgetal!
Geef een geheel getal (0 om te stoppen): -6
-6 is GEEN priemgetal!
Geef een geheel getal (0 om te stoppen): 0
Het aantal priemgetallen is 0
```

```
Geef een geheel getal (0 om te stoppen): 0
Het aantal priemgetallen is 0
```