

Naam: Klas: Studentnummer:

Instructie

Voor je ligt de eerste deeltoets van **CPL01**. Bij deze toets moeten enkele vragen worden uitgewerkt in **Flowgorithm** en **Code::Blocks**. De toets neemt **90 minuten** in beslag. Tijdens de toets mag je het boek “C How to Program” gebuiken. Internet, andere documenten en andere programma’s dan **Code::Blocks** en **Flowgorithm** zijn niet toegestaan en zullen worden opgemerkt door de **CPL01 logger**.

Als het klopt heb je Flowgorithm open met een bestand genaamd `jouwklas_jouwnaam_jouwstudnr.fprg` en heb je in Code::Blocks een project geopend. De source file in dit project heet `jouwklas_jouwnaam_jouwstudnr.c` en niet `main.c`.

Het `jouwklas_jouwnaam_jouwstudnr.c` bestand van dit project en de flowchart `jouwklas_jouwnaam_jouwstudnr.fprg` moeten worden ingeleverd in de inlevermap van CPL01: `..\inlever\ELEKTROTECHNIEK\2016-2017\CPL01\Deeltoets1`. **Let op: de bestanden in deze map slepen, en NIET direct vanuit een programma opslaan!**

Na het inleveren van je `.c` en `.fprg` bestand dien je dit opgaveblad, met je naam, klas en studentnummer ingevuld, terug te geven aan de docent. Zonder ingeleverd opgaveblad wordt je opdracht niet nagekeken!

Indien je vragen of problemen hebt die niet aan de programmeeropdracht zijn gerelateerd is de docent bereid te helpen. Succes met de toets!

Probleembeschrijving

We willen een programma maken dat een koffie/thee-automaat simuleert. De gebruiker moet een kopje koffie of kopje thee uit de automaat kunnen halen.

In de beginsituatie van de automaat zitten er drie kopjes koffie en drie kopjes thee in de machine.

- Het programma moet telkens afdrukken `Koffie:` met daarna voor elk nog aanwezig kopje koffie het teken `=`. Daarna moet worden afgedrukt `Thee:` met daarna voor elk nog aanwezig kopje thee het teken `=`.
- Vervolgens moet aan de gebruiker worden gevraagd: `Kies Koffie (1) of Thee (2):`.
- Daarna moet een getal worden ingelezen.
- Als de gebruiker het getal 1 heeft ingevoerd, dan moet het aantal aanwezige kopjes koffie met één worden verlaagd.

- Als de gebruiker het getal 2 heeft ingevoerd, dan moet het aantal aanwezige kopjes thee met één worden verlaagd.
- Als de gebruiker een ander getal heeft ingevoerd, dan moet dit genegeerd worden.
- In het geval dat er bijvoorbeeld geen kopjes koffie meer aanwezig zijn, maar nog wel kopjes thee, dan kun je uiteraard nog wel een kopje thee uit het apparaat halen. De keuze voor een kopje koffie wordt in dit geval genegeerd.
- Na het verwerken van de invoer begint het programma weer van voor af aan.
- Als er geen enkel kopje koffie en geen enkel kopje thee meer in het apparaat aanwezig is, dan moet het programma beëindigd worden.

Toetsvraag 1: Flowchart (LD1: 30 punten)

Maak van bovenstaande probleembeschrijving met behulp van **Flowgorithm** een flowchart. Hierbij gaat het er vooral om dat de structuur van het programma duidelijk wordt weergegeven. Het is niet nodig dat je de flowchart kunt 'uitvoeren' in Flowgorithm, maar dat mag natuurlijk wel. Als de flowchart 'uitgevoerd' wordt in Flowgorithm, dan zou dat eruit moeten zien zoals figuur 1.

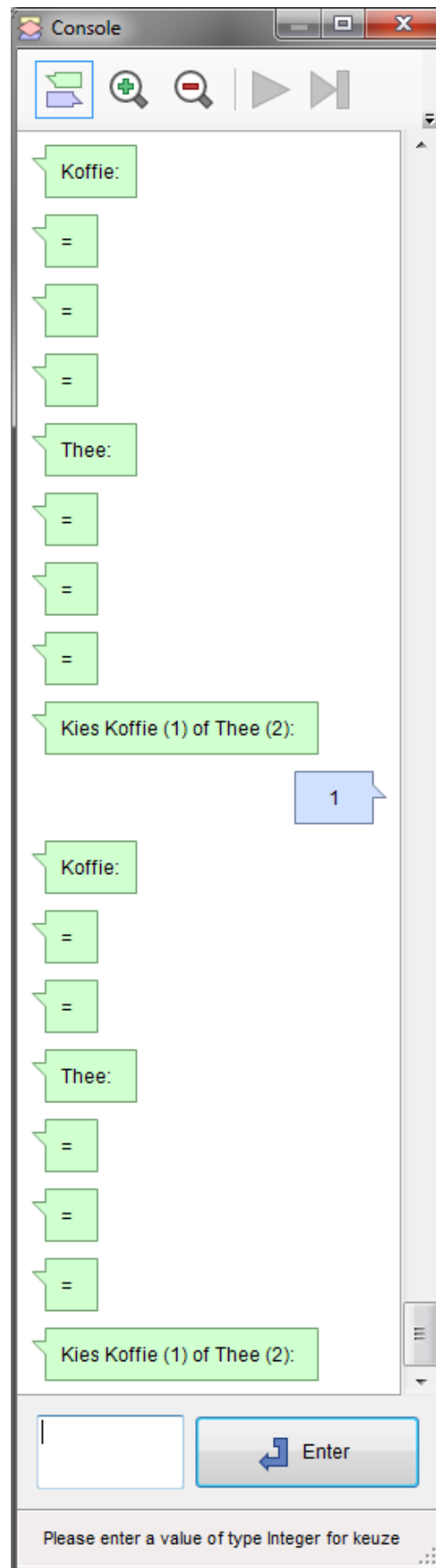
Toetsvraag 2: Programma (LD2,LD3,LD4: 50 punten)

Implementeer je oplossing in C met behulp van Code::Blocks. De opmaak van de uitvoer van het programma moet er exact zo uitzien als in figuur 2 gegeven is.

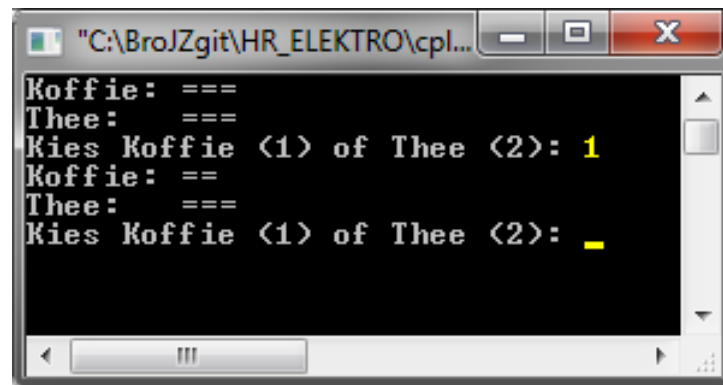
Toetsvraag 3: Uitbreiding (LD2,LD3,LD4: 20 punten)

Geef nu ook aan de gebruiker door wat het percentage kopjes koffie en het percentage kopjes thee is dat er nog in de machine zit. Er van uitgaande dat de machine in de beginsituatie voor 100% gevuld is. Geef het eindresultaat weer met twee cijfers achter de decimale punt.

De opmaak van de uitvoer van het programma moet er exact zo uitzien als in figuur 3 gegeven is.

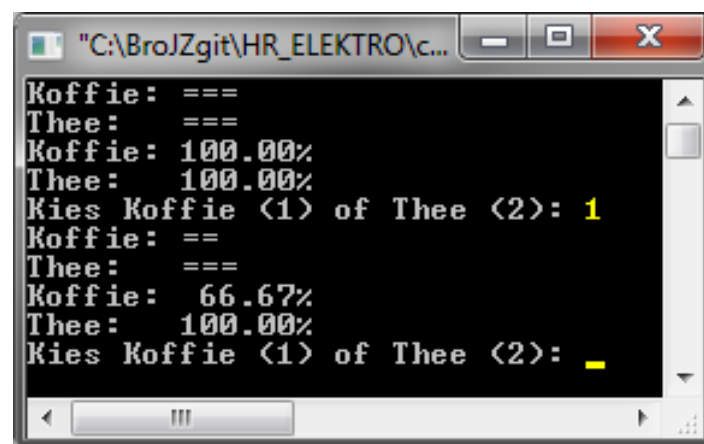


Figuur 1: Een mogelijk 'uitvoer' van Flowgorithm.



```
"C:\BroJZgit\HR_ELEKTRO\cpl..."  
Koffie: ===  
Thee: ===  
Kies Koffie <1> of Thee <2>: 1  
Koffie: ==  
Thee: ===  
Kies Koffie <1> of Thee <2>: _
```

Figuur 2: Een mogelijk run van het C-programma van toetsvraag 2.



```
"C:\BroJZgit\HR_ELEKTRO\c..."  
Koffie: ===  
Thee: ===  
Koffie: 100.00%  
Thee: 100.00%  
Kies Koffie <1> of Thee <2>: 1  
Koffie: ==  
Thee: ===  
Koffie: 66.67%  
Thee: 100.00%  
Kies Koffie <1> of Thee <2>: _
```

Figuur 3: Een mogelijk run van het C-programma van toetsvraag 3.

Naam: Klas: Studentnummer:

Instructie

Voor je ligt de eerste deeltoets van **CPL01**. Bij deze toets moeten enkele vragen worden uitgewerkt in **Flowgorithm** en **Code::Blocks**. De toets neemt **90 minuten** in beslag. Tijdens de toets mag je het boek “C How to Program” gebruiken. Internet, andere documenten en andere programma's dan **Code::Blocks** en **Flowgorithm** zijn niet toegestaan en zullen worden opgemerkt door de **CPL01 logger**.

Als het klopt heb je Flowgorithm open met een bestand genaamd `jouwklas_jouwnaam_jouwstudnr.fprg` en heb je in Code::Blocks een project geopend. De source file in dit project heet `jouwklas_jouwnaam_jouwstudnr.c` en niet `main.c`.

Het `jouwklas_jouwnaam_jouwstudnr.c` bestand van dit project en de flowchart `jouwklas_jouwnaam_jouwstudnr.fprg` moeten worden ingeleverd in de inlevermap van CPL01: `..\inlever\ELEKTROTECHNIEK\2016-2017\CPL01\Deeltoets1`. **Let op: de bestanden in deze map slepen, en NIET direct vanuit een programma opslaan!**

Na het inleveren van je `.c` en `.fprg` bestand dien je dit opgaveblad, met je naam, klas en studentnummer ingevuld, terug te geven aan de docent. Zonder ingeleverd opgaveblad wordt je opdracht niet nagekeken!

Indien je vragen of problemen hebt die niet aan de programmeeropdracht zijn gerelateerd is de docent bereid te helpen. Succes met de toets!

Probleembeschrijving

We willen een programma maken dat een snoepautomaat simuleert. De gebruiker moet een zakje drop of zakje zuurtjes uit de automaat kunnen halen.

In de beginsituatie van de automaat zitten er drie zakjes drop en drie zakjes zuurtjes in de machine.

- Het programma moet telkens afdrukken Drop: met daarna voor elk nog aanwezig zakje drop het teken +. Daarna moet worden afgedrukt Zuurtjes: met daarna voor elk nog aanwezig zakje zuurtjes het teken +.
- Vervolgens moet aan de gebruiker worden gevraagd: Kies Drop (1) of Zuurtjes (2):.
- Daarna moet een getal worden ingelezen.
- Als de gebruiker het getal 1 heeft ingevoerd, dan moet het aantal aanwezige zakjes drop met één worden verlaagd.

- Als de gebruiker het getal 2 heeft ingevoerd, dan moet het aantal aanwezige zakjes zuurtjes met één worden verlaagd.
- Als de gebruiker een ander getal heeft ingevoerd, dan moet dit genegeerd worden.
- In het geval dat er bijvoorbeeld geen zakjes drop meer aanwezig zijn, maar nog wel zakjes zuurtjes, dan kun je uiteraard nog wel een zakje zuurtjes uit het apparaat halen. De keuze voor een zakje drop wordt in dit geval genegeerd.
- Na het verwerken van de invoer begint het programma weer van voor af aan.
- Als er geen enkel zakje drop en geen enkel zakje zuurtjes meer in het apparaat aanwezig is, dan moet het programma beëindigd worden.

Toetsvraag 1: Flowchart (LD1: 30 punten)

Maak van bovenstaande probleembeschrijving met behulp van **Flowgorithm** een flowchart. Hierbij gaat het er vooral om dat de structuur van het programma duidelijk wordt weergegeven. Het is niet nodig dat je de flowchart kunt 'uitvoeren' in Flowgorithm, maar dat mag natuurlijk wel. Als de flowchart 'uitgevoerd' wordt in Flowgorithm, dan zou dat eruit moeten zien zoals figuur 1.

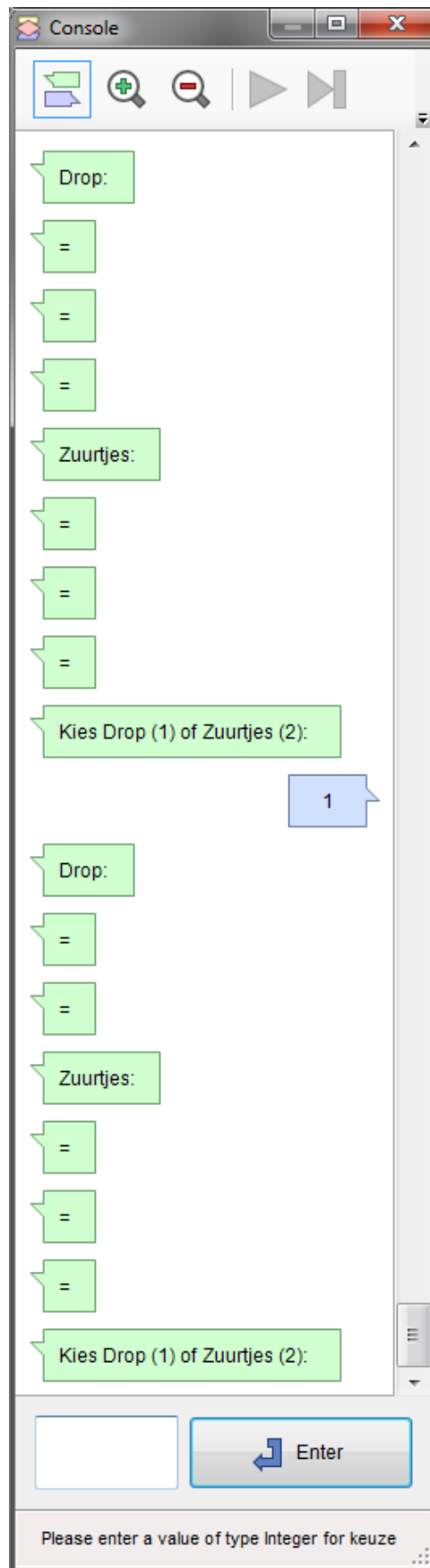
Toetsvraag 2: Programma (LD2,LD3,LD4: 50 punten)

Implementeer je oplossing in C met behulp van Code::Blocks. De opmaak van de uitvoer van het programma moet er exact zo uitzien als in figuur 2 gegeven is.

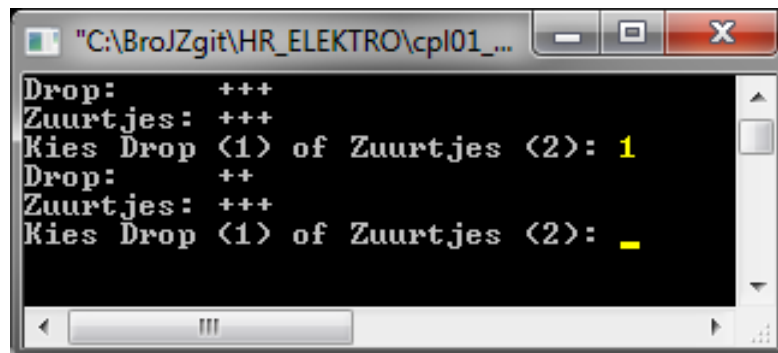
Toetsvraag 3: Uitbreiding (LD2,LD3,LD4: 20 punten)

Geef nu ook aan de gebruiker door wat het percentage zakjes drop en het percentage zakjes zuurtjes is dat er nog in de machine zit. Er van uitgaande dat de machine in de beginsituatie voor 100% gevuld is. Geef het eindresultaat weer met één cijfer achter de decimale punt.

De opmaak van de uitvoer van het programma moet er exact zo uitzien als in figuur 3 gegeven is.

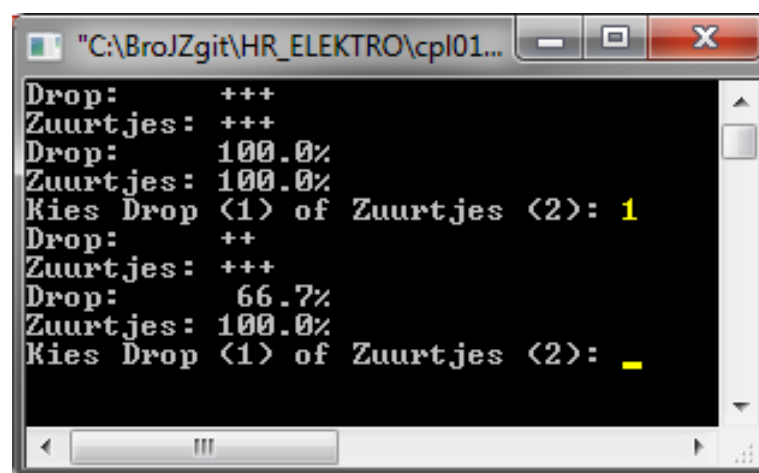


Figuur 1: Een mogelijk 'uitvoer' van Flowgorithm.



```
"C:\BroJZgit\HR_ELEKTRO\cpl01_...  
Drop:      +++  
Zuurtjes: +++  
Kies Drop <1> of Zuurtjes <2>: 1  
Drop:      ++  
Zuurtjes: +++  
Kies Drop <1> of Zuurtjes <2>: _
```

Figuur 2: Een mogelijk run van het C-programma van toetsvraag 2.



```
"C:\BroJZgit\HR_ELEKTRO\cpl01...  
Drop:      +++  
Zuurtjes: +++  
Drop:      100.0%  
Zuurtjes: 100.0%  
Kies Drop <1> of Zuurtjes <2>: 1  
Drop:      ++  
Zuurtjes: +++  
Drop:      66.7%  
Zuurtjes: 100.0%  
Kies Drop <1> of Zuurtjes <2>: _
```

Figuur 3: Een mogelijk run van het C-programma van toetsvraag 3.

Naam: Klas: Studentnummer:

Instructie

Voor je ligt de eerste deeltoets van **CPL01**. Bij deze toets moeten enkele vragen worden uitgewerkt in **Flowgorithm** en **Code::Blocks**. De toets neemt **90 minuten** in beslag. Tijdens de toets mag je het boek “C How to Program” gebruiken. Internet, andere documenten en andere programma’s dan **Code::Blocks** en **Flowgorithm** zijn niet toegestaan en zullen worden opgemerkt door de **CPL01 logger**.

Als het klopt heb je Flowgorithm open met een bestand genaamd `jouwklas_jouwnaam_jouwstudnr.fprg` en heb je in Code::Blocks een project geopend. De source file in dit project heet `jouwklas_jouwnaam_jouwstudnr.c` en niet `main.c`.

Het `jouwklas_jouwnaam_jouwstudnr.c` bestand van dit project en de flowchart `jouwklas_jouwnaam_jouwstudnr.fprg` moeten worden ingeleverd in de inlevermap van CPL01: `..\inlever\ELEKTROTECHNIEK\2016-2017\CPL01\Deeltoets1`. **Let op: de bestanden in deze map slepen, en NIET direct vanuit een programma opslaan!**

Na het inleveren van je `.c` en `.fprg` bestand dien je dit opgaveblad, met je naam, klas en studentnummer ingevuld, terug te geven aan de docent. Zonder ingeleverd opgaveblad wordt je opdracht niet nagekeken!

Indien je vragen of problemen hebt die niet aan de programmeeropdracht zijn gerelateerd is de docent bereid te helpen. Succes met de toets!

Probleembeschrijving

We willen een programma maken dat een drankautomaat simuleert. De gebruiker moet een flesje water of flesje frisdrank uit de automaat kunnen halen.

In de beginsituatie van de automaat zitten er drie flesjes water en drie flesjes frisdrank in de machine.

- Het programma moet telkens afdrukken `Water:` met daarna voor elk nog aanwezig flesje water het teken `#`. Daarna moet worden afgedrukt `Frisdrank:` met daarna voor elk nog aanwezig flesje frisdrank het teken `#`.
- Vervolgens moet aan de gebruiker worden gevraagd: `Kies Water (1) of Frisdrank (2):`.
- Daarna moet een getal worden ingelezen.
- Als de gebruiker het getal 1 heeft ingevoerd, dan moet het aantal aanwezige flesjes water met één worden verlaagd.

- Als de gebruiker het getal 2 heeft ingevoerd, dan moet het aantal aanwezige flesjes frisdrank met één worden verlaagd.
- Als de gebruiker een ander getal heeft ingevoerd, dan moet dit genegeerd worden.
- In het geval dat er bijvoorbeeld geen flesjes water meer aanwezig zijn, maar nog wel flesjes frisdrank, dan kun je uiteraard nog wel een flesje frisdrank uit het apparaat halen. De keuze voor een flesje water wordt in dit geval genegeerd.
- Na het verwerken van de invoer begint het programma weer van voor af aan.
- Als er geen enkel flesje water en geen enkel flesje frisdrank meer in het apparaat aanwezig is, dan moet het programma beëindigd worden.

Toetsvraag 1: Flowchart (LD1: 30 punten)

Maak van bovenstaande probleembeschrijving met behulp van **Flowgorithm** een flowchart. Hierbij gaat het er vooral om dat de structuur van het programma duidelijk wordt weergegeven. Het is niet nodig dat je de flowchart kunt 'uitvoeren' in Flowgorithm, maar dat mag natuurlijk wel. Als de flowchart 'uitgevoerd' wordt in Flowgorithm, dan zou dat eruit moeten zien zoals figuur 1.

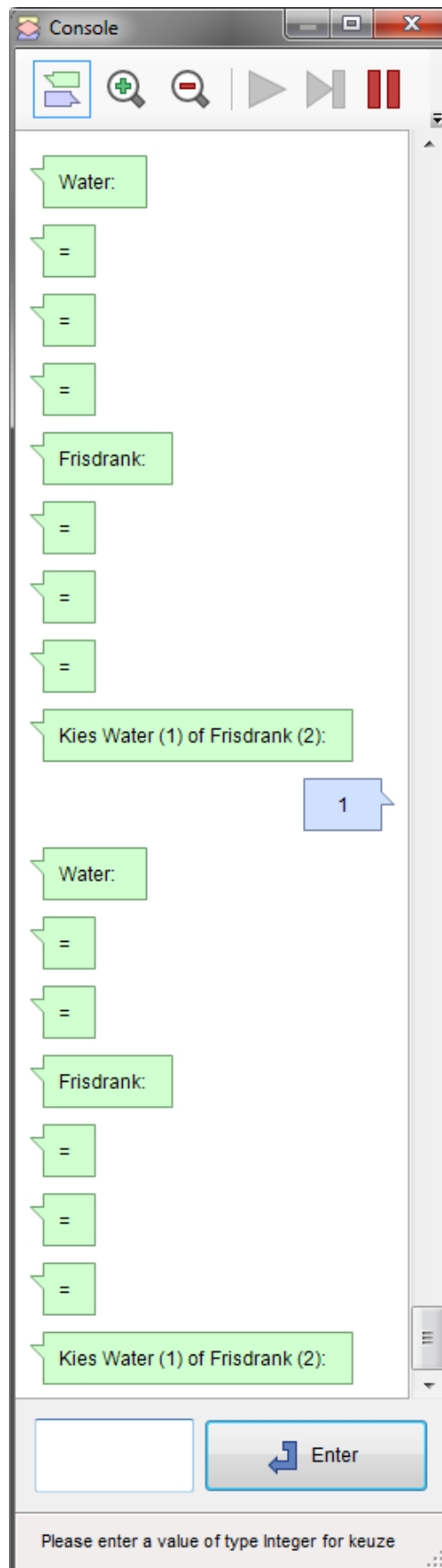
Toetsvraag 2: Programma (LD2,LD3,LD4: 50 punten)

Implementeer je oplossing in C met behulp van Code::Blocks. De opmaak van de uitvoer van het programma moet er exact zo uitzien als in figuur 2 gegeven is.

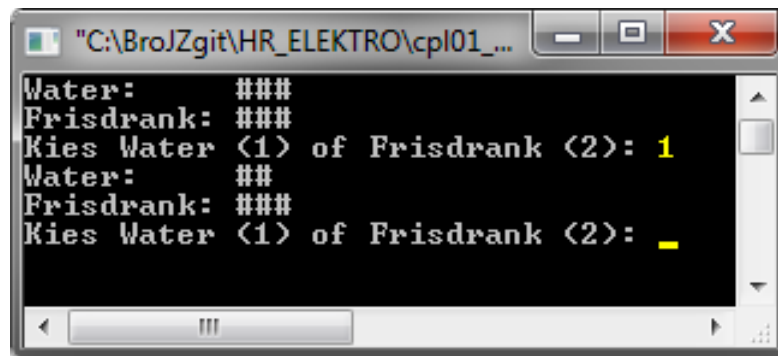
Toetsvraag 3: Uitbreiding (LD2,LD3,LD4: 20 punten)

Geef nu ook aan de gebruiker door wat het percentage flesjes water en het percentage flesjes frisdrank is dat er nog in de machine zit. Er van uitgaande dat de machine in de beginsituatie voor 100% gevuld is. Geef het eindresultaat weer met drie cijfers achter de decimale punt.

De opmaak van de uitvoer van het programma moet er exact zo uitzien als in figuur 3 gegeven is.

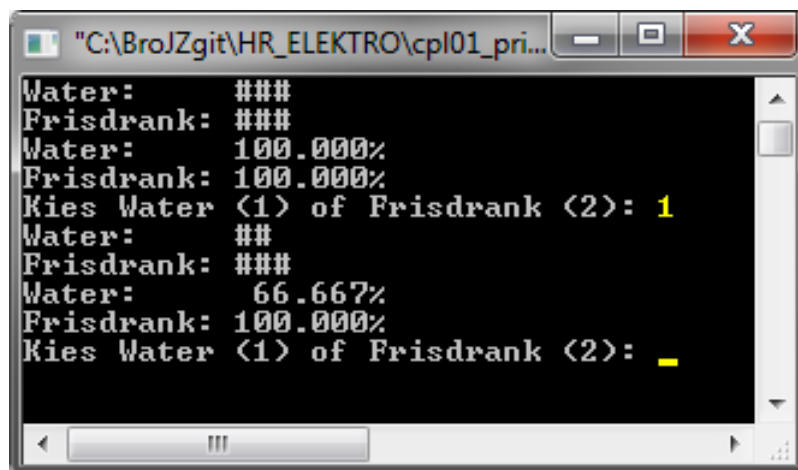


Figuur 1: Een mogelijk 'uitvoer' van Flowgorithm.



```
"C:\BroJZgit\HR_ELEKTRO\cpl01_...  
Water:   ###  
Frisdrank: ###  
Kies Water <1> of Frisdrank <2>: 1  
Water:   ##  
Frisdrank: ###  
Kies Water <1> of Frisdrank <2>: _
```

Figuur 2: Een mogelijk run van het C-programma van toetsvraag 2.



```
"C:\BroJZgit\HR_ELEKTRO\cpl01_pri...  
Water:   ###  
Frisdrank: ###  
Water:   100.000%  
Frisdrank: 100.000%  
Kies Water <1> of Frisdrank <2>: 1  
Water:   ##  
Frisdrank: ###  
Water:   66.667%  
Frisdrank: 100.000%  
Kies Water <1> of Frisdrank <2>: _
```

Figuur 3: Een mogelijk run van het C-programma van toetsvraag 3.