

Naam: ..... Klas: ..... Studentnummer: .....

## Instructie

Voor je ligt het tentamen van **CPL01**. Bij deze toets moeten **3 toetsvragen** worden uitgewerkt in **Flowgorithm** en/of **Code::Blocks**. De toets neemt **120 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**.

De bestanden `jouwklas_jouwnaam_jouwstudnr_opdr1.fprg`, `jouwklas_jouwnaam_jouwstudnr_opdr2.c` en `jouwklas_jouwnaam_jouwstudnr_opdr3.c` zullen na het afronden van de opdrachten moeten worden ingeleverd in de inlevermap van CPL01:

`..\inlever\ELEKTROTECHNIEK\2017-2018\CPL01\Tentamen`

**Let op: De bestanden in deze map slepen, en NIET direct vanuit een programma opslaan!**

**Let op: Sla je gemaakte flowchart eerst op en sluit Flowgorithm af voordat je je flowchart inlevert!**

Na het inleveren van je `.c` en `.fprg` bestanden dien je dit opgaveblad, met je naam, klas en studentnummer ingevuld, terug te geven aan de docent.

**Let op: 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!

## Toetsvraag 1: Dobbelen (LD1: 30 punten)

Een gewone dobbelsteen kan gebruikt worden om een willekeurige waarde tussen 1 en 6 (inclusief) te bepalen. Dit wordt gedaan door de dobbelsteen te werpen en het aantal ogen op de bovenliggende zijde te tellen. In een spel wordt een aantal dobbelstenen in een keer geworpen. Als het totaal van alle waarden van de geworpen dobbelstenen 42 bedraagt, heeft de speler het spel gewonnen. De speler kan bij elke worp kiezen hoeveel dobbelstenen hij wil gebruiken (minimaal 7 en maximaal 42). De waarden van de geworpen dobbelstenen worden in een array opgeslagen, Maak in Flowgorithm een flowchart die de volgende stappen uitvoert:

1. Vraag aan de gebruiker met hoeveel dobbelstenen geworpen wordt.
2. Controleer of dit aantal minimaal 7 en maximaal 42 is.
3. Herhaal de bovenstaande vraag totdat aan de bovenstaande voorwaarde is voldaan.
4. Om de worp te simuleren vul je de array met willekeurige waarden tussen 1 en 6 (inclusief). Je kunt hiervoor de aanroep `random(6)+1` gebruiken.
5. Bepaal **daarna** het totaal van de waarden van alle geworpen dobbelstenen.
6. Druk af of de speler gewonnen of verloren heeft.

## Toetsvraag 2: Aantal getallen tellen in een string (LD2, 3, en 4: 40 punten)

We willen van een willekeurige string weten hoeveel gehele getallen er in zitten. Schrijf daarom een functie genaamd `telGeheleGetallen` met een input parameter van het type C-string (een `char[]`) en een teruggeefwaarde (return value) van het type integer: De teruggeefwaarde van de functie is het aantal gehele getallen dat de string bevat.

Als voorbeeld:

De string: `Na 3 pogingen heb ik in 2017 uiteindelijk een 10 gehaald voor CPL01` bevat 4 gehele getallen. Namelijk 3, 2017, 10 en 01. De 01 in CPL01 telt namelijk ook als een geheel getal.

Het prototype van de functie is als volgt: `int telGeheleGetallen(char s[]);`

Je kunt de functie `int isdigit(int c);` uit `<ctype.h>` gebruiken om te kijken of het karakter `c` een cijfer is. Bedenk dat een geheel getal uit één of meer cijfers bestaat die zonder onderbreking achter elkaar staan.

Opdracht:

1. Implementeer de functie.
2. Schrijf een test voor de functie in de main.

## Toetsvraag 3: Omzetten coördinatenstelsels (LD2, 3 en 4: 30 punten)

In een bepaald systeem willen we poolcoördinaten  $(r, \theta)$  van een complex getal omzetten naar cartesische coördinaten  $(a + bj)$ . Hiervoor moet een functie geschreven worden met twee inputparameters en twee outputparameters. Gebruik variabelen van het type `double`. De functie maakt geen gebruik van de teruggeefwaarde (return value).

De formules om poolcoördinaten naar cartesische coördinaten te converteren zijn als volgt:

$$a = r \cdot \cos(\theta)$$

$$b = r \cdot \sin(\theta)$$

Opdracht:

- Kies een geschikte naam voor de functie.
- Schrijf een C-programma voor de bovengenoemde functie.
- Zorg dat je de functie op de juiste manier aanroept vanuit `main`, zodat je de functie kunt testen.
- De argumenten waarmee je de functie moet aanroepen zijn:  
 $r = 0.5$  en  $\theta = 3.14159265358979323846$  ( $\pi$ ).  
Je hoeft deze dus niet aan de gebruiker te vragen.
- Print het resultaat in de `main` functie.  
Het juiste resultaat is  $a = -0.500000$  en  $b = 0.000000$ .

## Einde Toets