

Naam: Klas: Studentnummer:

Instructie

Voor je ligt de tweede deeltoets van **CPL01**. Bij deze toets moeten enkele vragen worden uitgewerkt in **Flowgorithm** en **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. Dit kun je doen in de map:

`..\inlever\ELEKTROTECHNIEK\2015-2016\CPL01\Deeltoets2`

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

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: Flowchart (LD1: 30 punten)

Gegeven een array van 100 integers. Laat in de vorm van een flowchart zien hoe je de grootste (maximale) waarde binnen deze array kunt vinden. De flowchart hoeft niet werkend gemaakt te worden in flowgorithm maar een functionerende flowchart verifieert wel de werking van je algoritme.

De aanroep `random(100)` in flowgorithm zorgt voor een willekeurig getal tussen de 0 en 100 waarmee je de array kunt vullen om je flowchart eventueel te testen.

Toetsvraag 2: multiplyArray (LD2,3,4: 35 punten)

Wanneer elektronische schakelingen worden doorgerekend wordt er veel gebruik gemaakt van stelsels vergelijkingen. Deze vergelijkingen kunnen vervolgens in matrix formaat worden opgelost zoals je geleerd hebt bij de elektronica vakken.

Het doel van deze opdracht is een algemene functie te schrijven die een 1×3 vector kan vullen met een 3×3 matrix. Er wordt hierbij gebruik gemaakt van call-by-reference en van arrays.

Voor een algemene vector \times matrix vermenigvuldiging:

$$\begin{bmatrix} x_{uit} \\ y_{uit} \\ z_{uit} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (1)$$

geldt:

$$x_{uit} = x \cdot a_{11} + y \cdot a_{12} + z \cdot a_{13} \quad (2)$$

$$y_{uit} = x \cdot a_{21} + y \cdot a_{22} + z \cdot a_{23} \quad (3)$$

$$z_{uit} = x \cdot a_{31} + y \cdot a_{32} + z \cdot a_{33} \quad (4)$$

Schrijf een functie met het volgende prototype:

```
void multiplyArray(double *x, double *y, double *z, double matrix[3][3]);
```

Dit is een algemene functie om de 1×3 vector (x, y, z) te vermenigvuldigen met een 3×3 matrix volgens de formules uit vergelijkingen (2) tot (4). De vector (x, y, z) wordt m.b.v. pointers aan deze functie doorgegeven.

Als test kan je de volgende vermenigvuldiging gebruiken. Roep de functie aan in main, geef de juiste waarden mee en kijk of het resultaat overeenkomt.

$$\begin{bmatrix} 1.4 \\ 3.2 \\ 5.0 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \end{bmatrix} \quad (5)$$

Toetsvraag 3: dagnaam2EnumDag (LD2,3,4: 35 punten)

Op de netwerkschijf staat het onvolledige programma opdr3.c:

```
... \2015-2016\CPL01\Toetsen\Deeltoets2\opdr3.c.
```

Dit programma moet de naam van een dag kunnen inlezen in een string. Deze string moet vervolgens worden omgezet naar een enumeratie. Afhankelijk van de enumeratie wordt vervolgens een bepaalde uitvoer geprint.

De functie dagnaam2EnumDag moet nog afgemaakt worden. Deze functie moet de string dagnaam omzetten naar de enumeratie `enum dag`. Let op! De naam van de dag mag uit kleine letters, uit hoofdletters of uit een combinatie van hoofdletters en kleine letters bestaan. Tip: m.b.v. de functie `tolower` kun je een hoofdletter omzetten naar een kleine letter.

Hieronder is een testrun van een goed werkend programma gegeven. De input is onderstreept weergegeven.

Geef de naam van een dag: MaAnDaG

MaAnDaG is wasdag.