

Naam: Klas: Studentnummer:

Instructie

Voor je ligt de Herkansingstoets 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” gebuiken. 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\Herkansing
```

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

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)

Het spel ‘raad het getal’ is een simpel spelletje. Speler 1 voert een getal in en vervolgens moet speler 2 dit getal raden. Nadat speler 1 een getal heeft ingevoerd, vraagt het spel continu aan speler 2 om een getal in te voeren. Zodra het ingevoerde getal van speler 1 gelijk is aan de input van speler 2, dan is de uitvoer op het scherm bijvoorbeeld: Dit is het juiste getal! Hierna stopt het spel.

Laat in de vorm van een flowchart zien hoe dit spel eruit ziet. De flowchart wordt bij voorkeur gemaakt met behulp van Flowgorithm, maar mag op papier worden ingeleverd. De flowchart hoeft niet werkend gemaakt te worden in Flowgorithm.

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

Het spel paardenrace wordt gespeeld met 3 paarden (deelnemers). Het aantal stappen dat een paard per beurt mag zetten, wordt bepaald door het gooien van een dobbelsteen. Dus als iemand 6 gooit mag zijn paard 6 stappen zetten. Het parcours is 40 stappen lang. Wie als eerste aan of over de finish is, heeft gewonnen.

Dit spel ga je programmeren in C. Een deel van dit programma is al gegeven in het bestand `paardenrace.c`.

Toetsvraag 3: Race-statistieken (LD2,3,4: 35 punten)

We hebben van drie formule 1 raceauto's de gereden tijden van 10 rondes bijgehouden over twee Grand Prix races. Nu zijn we per Grand Prix benieuwd welke auto de beste tijd heeft gereden in welke ronde. We gaan hiervoor een functie schrijven.

In het programma `autorace.c` is de Grand Prix data te vinden en is er een lege functie gemaakt:

```
int snelsteRonde(double gp[3][10], double *snelsteTijd, int *snelsteRonde);
```

Vul de functie `snelsteRonde` in, om de snelste ronde te bepalen, de tijd die hierbij hoort en de raceauto die deze gereden heeft. De functie geeft het nummer van de raceauto die de snelste ronde heeft gereden terug (*return value*). Daarnaast stopt de functie de bijbehorende ronde en rondetijd in de *call-by-reference* parameters `snelsteRonde` en `snelsteTijd` om deze zo terug te geven aan `main`.

De matrices `monaco` en `hockenheim` in `main` bestaan uit drie rijen (Raceauto 1, Raceauto 2, Raceauto 3) en tien kolommen (Ronde 1, Ronde 2, ..., Ronde 10) met de rondetijden voor de circuits van Monaco en Hockenheim.

In `main` wordt de functie `snelsteRonde` tweemaal aangeroepen (voor beide Grand Prix een keer) waarna (eveneens in `main`) het nummer van de raceauto met de snelste ronde en snelste rondetijd met 2 cijfers achter de komma wordt afgedrukt zoals in onderstaand voorbeeld.

Monaco: In ronde 5 heeft auto 1 de snelste ronde gereden in 80.04 seconden!

Hockenheim: In ronde 7 heeft auto 2 de snelste ronde gereden in 60.39 seconden!