Technisch-Wetenschappelijke Software
Oefenzitting 1: Fortran 95, kennismaking

Peter Opsomer

6 oktober 2016

Tijdens de zitting

1. Schrijf een programma dat twee gehele getallen definieert en dat afdrukt of het eerste strikt groter is dan het tweede. Doe dit met een *single line if-statement* (pas dus de `hello_world` van je voorbereiding aan) en dan ook door het rechtstreeks afprinten van een (naamloze) logische variabele. Test dit op een compiler die je kiest.

2. Maak een programma dat een getal inleest (je mag aannemen dat het een integer is) en dan zoveel keer `Hello, world!` afdrukt. Voeg enkele extra controles toe zodat zinloze invoer of belachelijk grote getallen op een propere manier afgehandeld worden.

3. Plaats de code uit de vorige opgave als een subroutine zonder argumenten in een module `hello_module` die in het bestand `hello_module.f95` of `hello_module.f90` zit. Pas het vorige programma aan zodat het deze subroutine gebruikt. De structuur van deze programma-eenheden staat in de 'cheatsheet'.

   Maak het programma door eerst `hello_module` te compileren naar object code, het programma te compileren naar object code en daarna het uitvoerbaar bestand samen te stellen.

   Probeer eens om het programma te maken met slechts één van de twee object bestanden om een idee te krijgen van de foutmeldingen die daardoor veroorzaakt worden. Wat is het effect als je programma probeert te compileren naar een object bestand, in afwezigheid van `hello_module.mod`?

4. Haal de `Makefile` af van Toledo, verwijder eventuele haakjes in de naam van het bestand en compileer het programma uit de vorige opgave ermee door middel van `make hello_world`. Probeer uit te vissen hoe dit systeem de onderlinge afhankelijkheden in de `Makefile` gebruikt om zo weinig mogelijk te compileren. Kijk na wat het effect is van een verandering in de programmacode als je daarna opnieuw `make hello_world` uitvoert of als je enkel de `linking` doet.

5. Schrijf een programma dat de tekenreeks 'Here's to the crazy ones. The misfits. The rebels.' op drie manieren gaat printen: links uitgelijnd, gecentreerd en rechts uitgelijnd. Neem als schermbreedte 80 karakters. Een mogelijke uitvoer is alsvolgt:

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
Here's to the crazy ones. The misfits. The rebels.
              Here's to the crazy ones. The misfits. The rebels.
                                  Here's to the crazy ones. The misfits. The rebels.
```

6. Creëer een programma dat een `recursive` functie bevat die de dubbele faculteit kan berekenen voor alle gehele getallen. Deze functie wordt hier op de volgende manier gedefinieerd

$$n!! = \begin{cases} n \cdot (n-2)!! & \text{als } n > 0 \\ 1 & \text{als } n = 0 \text{ of } n = -1 \\ 0 & \text{als } n < -1. \end{cases}$$

   Gebruik een `select case` blok. Evalueer deze functie voor $n = -2$ tot $n = 29$. Wat zie je gebeuren?

Huistaak

Ontwikkel een Fortranprogramma `kennismaking.f90` dat de volgende subroutines en functie bevat, en test met je favoriete compiler:

1. Maak een subroutine `matrix_stats` die een aantal eigenschappen van een matrix van reële getallen (dit is het argument van de subroutine) print:
   - De afmetingen (aantal rijen, aantal kolommen) en het aantal elementen.
   - De kleinste en grootste waarde in de hele matrix.
   - De bereiken van de indices voor zowel de rijen als de kolommen. Dit doe je zowel in de subroutine als in het hoofdprogramma: bespreek het verschil.
   - De som van elke rij en de som van elke kolom.

   Zorg dat de routine overweg kan met elke grootte van matrix en test voor de volgende 2 matrices:

```
real, dimension(4,4) :: something_special = reshape( (/ &
   1.0000e+00,   1.2500e-01,   1.8750e-01,   8.1250e-01, &
   3.1250e-01,   6.8750e-01,   6.2500e-01,   5.0000e-01, &
   5.6250e-01,   4.3750e-01,   3.7500e-01,   7.5000e-01, &
   2.5000e-01,   8.7500e-01,   9.3750e-01,   6.2500e-02  &
   /), (/ 4,4 /), order=(/2,1/) )

real, dimension(-3:3,0:2) :: different_special = reshape( (/ &
   9.0909e-01,1.9091e+00,8.1818e-01,1.4545e+00,4.5455e-01, &
   1.2727e+00,1.8182e-01,2.7273e-01,3.6364e-01,6.3636e-01, &
   1.0000e+00,1.3636e+00,1.6364e+00,1.7273e+00,1.8182e+00, &
   7.2727e-01,1.5455e+00,5.4545e-01,1.1818e+00,9.0909e-02, &
   1.0909e+00 /), (/7,3/), order=(/1,2/) )
```

   Tip: dit is ook elektronisch beschikbaar op Toledo. Begin dit dus niet over te typen ...

2. Schrijf een subroutine die één karakter aanvaardt en aan de hand van een *select case* blok een zelf gekozen tekenreeks van 40 karakters links uitgelijnd print als het karakter 'l' is, 'c' voor gecentreerd en 'r' voor rechts. Als het karakter 'i' is, wordt eerst de tweede helft van je tekenreeks en dan de eerste helft afgeprint; het geheel is gecentreerd op één lijn. Neem als schermbreedte 80 karakters en test deze vier opties. Programmeer dit zo kort mogelijk en argumenteer.

3. Maak een `elemental` functie `graycode(n)` die voor gehele getallen n de decimale waarde van de Gray-code van n berekent via de formule (1) waarbij "$\oplus$" de bitsgewijze exclusieve OR-operatie voorstelt. Met "$\gg$" bedoelen we een shift van 1 bit naar rechts.

$$\text{GrayCode} = n \oplus (n \gg 1) \tag{1}$$

   Het is de bedoeling dat je functie `elemental` is en geen lussen gebruikt maar werkt met de beschikbare bit-manipulatie-functies. Test zowel met een scalar, een vector als een matrix. Laat dan het programma de eerste 20 graycode getallen berekenen en illustreer de typische structuur in het verschil tussen twee opeenvolgende getallen van deze reeks: bespreek.

Decimaal	0	1	2	3	4	5	6	...
n (binair)	0000	0001	0010	0011	0100	0101	0110	...
	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	
$n \gg 1$ (binair)	0000	0000	0001	0001	0010	0010	0011	...
	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	
$n \oplus (n \gg 1)$ (binair)	0000	0001	0011	0010	0110	0111	0101	...
Gray-code	0	1	3	2	6	7	5	...

Zet uitvoer van uw programma, samen met je antwoorden en andere gevraagde informatie (zoals naam, jaar en gevolgde studieprogramma(s): zie richtlijnen) als commentaar in de broncode en stuur dit `kennismaking.f90`-bestand door via e-mail naar `peter.opsomer@cs.kuleuven.be` vóór 19 oktober 14h. Volg hierbij strikt de richtlijnen voor het indienen van de huistaken zoals uitgedeeld en op Toledo-Oefenzittingen.

Het is de bedoeling dat je individueel aan deze gequoteerde huistaak werkt en normaal niet meer dan een dag. Vragen over Fortran, de oefenzittingen of deze opgave stel je op de discussieruimte `Oefenzittingen` van Toledo, niet via mail. Kijk ook in het cursusmateriaal en de Fortran standaard, die je nodig hebt voor de opgaves.