

Structuur van Computerprogrammas II: Week 2

Elisa Gonzalez Boix

Maarten Vandercammen

1 Introductie

1. Maak je eerste Hello World programma via ofwel Eclipse ofwel CLion.

Als je **Eclipse** kiest, vind je op Canvas in het bestand **HelloWorld in Eclipse.pdf** instructies om dat te doen. Volg die instructies en voer het programma uit. Op de computers in de computerzalen open je Eclipse door naar het startmenu linksboven te gaan → Development → Eclipse.

Instructies om dit voor **CLion** te doen, vind je op Canvas in het bestand **HelloWorld in CLion.pdf**. Op de computers in de computerzalen kan je CLion openen door het commando `/usr/local/clion/bin/clion.sh` in de terminal uit te voeren.

2. In C kan je de `printf` functie gebruiken om tekst in je console te printen. Deze functie neemt minstens 1 argument, een string die de tekst voorstelt die geprint zal worden, gevolgd door een variabel aantal andere argumenten. Hieronder vind je enkele voorbeelden van hoe deze functie gebruikt kan worden:

- `printf("Hallo iedereen!\n")` → print "Hallo iedereen!", gevolgd door een newline
- `printf("5 + 5 = %i", 10)` → print "5 + 5 = 10"
- `printf("Structuur %i wordt gegeven door %s", 2, "Elisa")` → print "Structuur 2 wordt gegeven door Elisa"

Zoals je ziet kan het eerste argument van deze functie speciale karakters bevatten, zoals `%i` of `%s`. Deze tekens worden *format specifiers* genoemd. Op het moment dat de tekst geprint wordt, wordt elke specifier vervangen door een van de extra argumenten aan `printf` wiens positie overeenkomt met de positie van die specifier. Deze format specifiers kunnen verschillende vormen aannemen, afhankelijk van het *type* van het argument dat geprint moet worden. Zo wordt bv. een `%i` gebruikt om integers te printen, een `%s` om strings te printen en een `%g` om floating-point getallen te printen¹. Experimenteer nu eens met het printen van verschillende soorten expressies.

¹Het volledige overzicht van alle format specifiers vind je op <http://www.cplusplus.com/reference/cstdio/printf/>

Wat gebeurt er als je een floating-point getal als een integer probeert te printen? Of een integer als een string? Wat gebeurt er als je minder/meer format specifiers meegeeft dan er argumenten zijn?

3. (**Dodona**) Maak een functie om de faculteit van een getal te berekenen. In Scheme kan je die functie bv. als volgt definiëren:

```
(define (fac n)
  (if (< n 2)
      1
      (* n (fac (- n 1)))))
```

of in Python:

```
def fac(n):
    if n < 2:
        return 1
    else:
        return n * fac(n - 1)
```

Vertaal deze functie nu naar C. Laat de functie een integer-type als argument nemen en een integer-type teruggeven: implementeer m.a.w. de functie `int fac(int n)`.

Zoals je hebt gezien in de tweede oefening, kan je het resultaat van een oproep naar `fac` in de console printen via `printf("%i", fac(n))`.

4. (**Dodona**) De grootste gemeenschappelijke deler (ggd) van twee getallen a en b is gedefinieerd als het grootst mogelijke getal dat zowel a als b deelt. Euclides ontdekte dat $\text{ggd}(a, b)$ gelijk is aan a als b gelijk is aan 0, en anders dat $\text{ggd}(a, b)$ gelijk is aan $\text{ggd}(b, c)$, waar c de rest was na deling van a door b . Maak hiervan gebruik om de `int ggd(int a, int b)` functie te definiëren. In C bereken je een deling van a door b via a / b . De rest van zo'n deling bereken je via $a \% b$.
5. (**Dodona**) Implementeer een functie `void vkv(float a, float b, float c)` om de oplossingen van een vierkantsvergelijking (vergelijkingen van de vorm van de vorm $ax^2 + bx + c$, waarbij a , b , en c kommagetallen kunnen zijn) te berekenen. Deze vergelijkingen hebben nul, een of twee oplossingen, afhankelijk van of D (waarbij D gelijk is aan $b^2 - 4ac$) kleiner dan, gelijk aan of groter dan 0 is. Als D groter is dan 0, zijn die oplossingen gelijk aan $\frac{-b-\sqrt{D}}{2a}$ en $\frac{-b+\sqrt{D}}{2a}$. Een vierkantswortel berekenen kan je doen met de `sqrt` functie uit de `math.h` library.

Laat je functie afhankelijk van het resultaat ofwel `Geen oplossingen`, ofwel `Oplossing = X`, ofwel `Oplossing 1 = X en oplossing 2 = Y` printen via de `printf` functie, waar X en Y kommagetallen zijn. Aangezien de functie niets hoeft terug te geven, kan je het return-type van deze functie specificeren als `void`.

6. (**Dodona**) Schrijf een functie `void collatz(int n)` om de Collatz-sequentie voor het getal `n` uit te printen. Een Collatz-sequentie begint steeds bij een natuurlijk getal `n`. Als `n` even is, gaat de sequentie verder met `n / 2`. Als `n` oneven is, gaat de sequentie verder met `n * 3 + 1`. Het einde van de sequentie wordt bereikt wanneer `n` gelijk is aan 1. `collatz(5)` aanroepen zou dus het volgende moeten printen:

```
5
16
8
4
2
1
```

7. (**Dodona**) De sinus van een getal `x` (uitgedrukt in radialen) kan benaderd worden via de volgende serie: $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$. Implementeer een functie `double calc_sine(double x, int n)` die de sinus van `x` tot op `n` termen berekent. Zo zou `calc_sine(x, 1)` enkel de term `x` moeten teruggeven, `calc_sine(x, 2)` de som $x - \frac{x^3}{3!}$ enz. Een machtsverheffing a^b kan berekend worden via de `pow` functie uit de `math` library: `pow(a, b)`. Om de faculteit te berekenen, kan je de functie `fac` uit oefening 3 hergebruiken. Als je deze oefening op Dodona maakt, mag je ervan uitgaan dat zowel `pow` als `fac` al gedefinieerd zijn.

2 Extra

8. (**Dodona**) Implementeer een functie `long ackermann(short m, short n)` om de *Ackermann* functie te implementeren. De Ackermann functie is een dubbel-recursieve, wiskundige functie die gedefinieerd is als:

$$A(m, n) = \begin{cases} n + 1 & \text{if } m = 0 \\ A(m - 1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m - 1, A(m, n - 1)) & \text{if } m > 0 \text{ and } n > 0 \end{cases} \quad (1)$$

Je kan de waarden uit de tabel op https://en.wikipedia.org/wiki/Ackermann_function#Table_of_values gebruiken om te controleren of je oplossing juist is.

9. (**Dodona**) Schrijf een functie `void print_difference(int h1, int m1, int s1, int h2, int m2, int s2)` om het verschil (in absolute waarde) in uren, minuten en seconden tussen twee tijdstippen uit te printen. Laat de functie zes parameters nemen: het uur, de minuut, en de seconde van beide tijdstippen. De functie oproepen als `print_difference(10, 0, 0, 11, 2, 3)` zou bv. het volgende moeten printen:

```
1 uren, 2 minuten, 3 seconden
```