

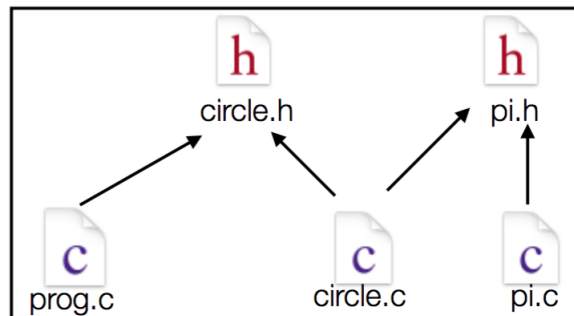
# Structuur van Computerprogrammas II: Week 3

Elisa Gonzalez Boix

Maarten Vandercammen

## 1 Makefiles

1. Download het bestand `main.c` op Canvas. Schrijf een simpele makefile, bestaande uit slechts één regel, om een executable voor dit programma te maken.
2. Download `circle_example.zip` op Canvas. Schrijf een makefile voor dit programma, met een aparte regel voor elke object file die moet gecompileerd worden. De dependencies die voor deze oefening belangrijk zijn zien er uit als volgt:



## 2 Control flow in C

3. (**Dodona**) Een klassieke programmeeroefening is de FizzBuzz-oefening. Hierin is het de bedoeling dat je voor alle getallen tussen 1 en 100 "Fizz" in de console print als dit getal deelbaar is door 3, "Buzz" als het deelbaar is door 5 en "FizzBuzz" als het deelbaar is door 3 en door 5. Het restant van de deling van `a` door `b` vind je via de expressie `a % b`. Implementeer de functie `void fizzbuzz(int n)` die voor alle getallen van 1 tot en met `n` het volgende print: ofwel `i: FizzBuzz\n`, ofwel `i: Fizz\n`, ofwel `i: Buzz\n`, waarbij `i` het huidige getal voorstelt, afhankelijk van of `i` deelbaar is door 3 en door 5, enkel deelbaar is door 3, of enkel deelbaar is door 5. Als `i` door geen van beide getallen deelbaar is, wordt er niets geprint.

4. (**Dodona**) De cosinus van een getal  $x$  (uitgedrukt in radialen) kan benaderd worden via de volgende serie:  $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$ . Implementeer een functie `double calc_cosine(double x, int n)` die de cosinus van  $x$  tot op  $n$  termen berekent. Zo zou `calc_cosine(x, 1)` enkel de term 1 moeten teruggeven, `calc_cosine(x, 2)` het resultaat  $1 - \frac{x^2}{2!}$  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 van het vorige WPO hergebruiken. Als je deze oefening op Dodona maakt, mag je ervan uitgaan dat zowel `pow` als `fac` al gedefinieerd zijn.
5. (**Dodona**) Maak een functie `void print_christmas_tree(int n)` die een kerstboom van hoogte  $n$  in je console print. `print_christmas_tree(4)` zou bv. het volgende moeten printen:

```

      *
    ***
  *****
*****
      *
      *
```

6. (**Dodona**) Een simpele manier om na te gaan of een getal  $x$  een priemgetal is of niet, is door alle getallen tussen 2 en  $x$  af te lopen en te kijken of het restant van de deling van  $x$  met een van deze getallen gelijk is aan nul. Schrijf een functie `void print_primes(int x)` die alle priemgetallen tussen 2 en  $x$  uitprint, van elkaar gescheiden door een newline (het karakter 'n').
7. (**Dodona**) Schrijf een functie `void print_perfect_numbers(int n)` die alle *perfecte* getallen, van elkaar gescheiden door een newline ('n'), print tussen 1 en  $n$ . In de wiskunde wordt een natuurlijk getal perfect genoemd als de som van al diens 'echte' factoren (t.t.z., alle factoren van het getal behalve het getal zelf) gelijk is aan dat getal. 6 is bv. een perfect getal, want zijn 'echte' factoren zijn 1, 2 en 3, en de som van deze factoren is ook gelijk aan 6.

### 3 Extra

8. (**Dodona**) Schrijf een functie `int convert_hex_to_dec(char hex[])` die, gegeven als argument een string die de hexadecimale representatie van een getal voorstelt, de decimale representatie van dit getal teruggeeft. `convert_hex_to_dec("5F")` zou bv. het getal 95 moeten teruggeven. Om deze oefening op te lossen zal je waarschijnlijk de functie `strlen` moeten gebruiken die, gegeven een string als argument, de lengte van deze string

teruggeeft. Om deze functie te gebruiken moet je `#include <string.h>` bovenaan je codefile toevoegen. Het n-de karakter van een string inlezen doe je via `string[n]`.

9. (**Dodona**) Implementeer het insertion-sort algoritme voor strings om een gegeven string alfabetisch te sorteren. Twee karakters vergelijken kan je gewoon doen via o.a. de operator `<`. Je zou dit algoritme dan moeten kunnen oproepen via bv.:

```
char string[] = "randomstring";  
insertion_sort(string);
```

waarna de karakters in `string` alfabetisch gesorteerd zouden moeten zijn en `string` dus als waarde "adgimnnorrst" zou moeten hebben.

$$\cos(x) = \frac{x^0}{0!} - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$