

Structuur van Computerprogrammas II: Week 4

Elisa Gonzalez Boix Maarten Vandercammen Robbe De Greef

1 Debugging

1. Download `fib.c` van Canvas en gebruik een debugger (ofwel via de terminal ofwel via Eclipse of CLion) om door de executie van de `fib` functie te stappen en de waarden van de `fib_n` en `fib_n1` variabelen te kunnen inspecteren.

2 Control flow in C

2. (**Dodona**) Schrijf een functie `void convert_whitespace(char string[])` die in de gegeven string alle whitespace-karakters (t.t.z. de karakters '\t', '\n' en ' ') omzet naar het underscore karakter '_'.
Zie slide 7 uit hoofdstuk 2 voor een voorbeeld van hoe je C strings gebruikt. Het karakter met index `i` uit een string halen, kan je doen via `string[i]`. Dat karakter vervangen door bv. het underscore karakter kan je doen via `string[i] = '_'`. Een string eindigt altijd met het null karakter '\0'.
3. (**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 Pointers

4. (**Dodona**) Schrijf een functie `void rotate(int *pa, int *pb, int *pc)` die drie pointers naar integers `a`, `b` en `c` als input neemt en er vervolgens voor zorgt dat de waarde van `b` nu in `a` komt, de waarde van `a` in `c` komt, en de waarde van `c` in `b` komt. Als je bv. de volgende code uitvoert:

```
int a = 1, b = 2, c = 3;  
rotate(&a, &b, &c);
```

zou `a` hierna gelijk aan 2 moeten zijn, `b` gelijk aan 3 en `c` gelijk aan 1.

- (**Dodona**) Schrijf een functie `void do_arithmetic(float numbers[], int length, float *p_sum, float *p_product, int *p_contains_42)` die gegeven een array `numbers` van kommagetallen en de lengte `length` van die array, zowel de som als het product van die getallen berekent. Eens die getallen berekend zijn, worden ze opgeslagen in respectievelijk de `float` waar de pointers `p_sum` en `p_product` naar wijzen. Daarnaast controleert de functie ook of `numbers` het getal 42 bevat. Zo ja zorgt de functie ervoor dat de `int` waar `p_contains_42` naar wijst wordt geassigned aan 1. Zonnee wordt die `int` geassigned aan 0.
- (**Dodona**) In C bestaat er de `scanf`-functie, waarmee je getallen, karakters, strings enz. kan inlezen. Die functie kan bv. aangeroepen worden als volgt:

```
char c;
scanf("%c", &c);
```

In dit voorbeeld zal `scanf` wachten op de gebruiker om 1 karakter in de console in te geven en op enter te drukken. Het ingegeven karakter wordt dan opgeslagen in de variabele waarvan het geheugenadres werd meegegeven. Als de gebruiker bv. het karakter `a` ingaf, zou in dit voorbeeld de variabele `c` het karakter `'a'` bevatten.

Zoals je kan zien gebruikte `scanf` dezelfde *format specifiers* (die je hebt gezien in oefening 2 van het eerste WPO) als `printf` om te bepalen wat voor waarde (karakter, integer, string enz.) er moet ingelezen worden. Maak nu zelf een variant `void my_scanf(char string[], char *c)` op deze functie die de functionaliteit van `scanf` en `printf` combineert. Het is de bedoeling dat je deze functie als volgt zou kunnen gebruiken:

```
char c;
my_scanf("Enter a character: %c", &c);
```

Je variant zou dus niet alleen een karakter inlezen, maar ook de rest van de string (`Enter a char:`) meteen uitprinten. Het is voldoende om `my_scanf` enkel één karakter te laten inlezen, en geen strings of integers. Je kan de `getchar`-functie uit de `<stdio.h>` library gebruiken om een karakter in te lezen.

4 Extra

- (**Dodona**) Pas de `scanf`-functie uit oefening 6 aan naar een `void my_scanf_int(char string[], int *i)` zodat je integers kan inlezen. Let op: een integer kan bestaan uit meerdere cijfers. Je kan dus niet op voorhand weten hoeveel karakters je voor een bepaald getal moet inlezen. Wel kan je gebruik maken van het feit dat `getchar` het newline karakter `'\n'` teruggeeft als het einde van de input bereikt is.