

Structuur van Computerprogrammas II: Week 8

Elisa Gonzalez Boix Maarten Vandercammen Robbe De Greef

1 Geheugenallocatie

1. Hoeveel geheugen moet er minstens statisch gealloceerd worden voor de functie `read_chess_pieces` uit het bestand `oefeningen.c`? Hoeveel geheugen wordt er dynamisch gealloceerd tijdens de executie van die functie als de gebruiker n schaakstukken wil inlezen?

2 Command-line argumenten

2. (**Dodona**) Schrijf een programma dat drie *command-line argumenten* neemt: een getal, een operator karakter ('+', '-', 'X' of '/') en een tweede getal. Het programma print dan het resultaat van de operator toegepast op de twee getallen. Om een string (nl. het command-line argument) om te zetten in een getal, kan je bv. de `atoi` functie uit `<stdlib.h>` gebruiken¹. Zie ook de laatste pagina van deze opgaven voor instructies om command-line argumenten mee te geven via Eclipse of CLion.

3 Function pointers

3. (**Dodona**) Schrijf een functie `int *filter(int *array, int *p_length, int (*f)(int))` die een array van integers filters. De functie neemt als argumenten een array van integers, een **pointer naar** de lengte van deze array, en een pointer naar een functie die één **integer** als argument neemt en een **integer** teruggeeft.

`filter` maakt een nieuwe array aan die enkel elementen bevat uit de meegegeven array waarvoor de functie `f` géén 0 teruggaf. Daarnaast slaat `filter` ook de lengte van de gefilterde array op in de variabele waar `p_length` naar wijst.

¹https://www.tutorialspoint.com/c_standard_library/c_function_atoi.htm

4 Pointers naar pointers

4. (**Dodona**) Op Dodona en in `oefeningen.c` vind je de definitie van een `struct Student` die een student met een studenten-id en een puntengemiddelde voorstelt. Implementeer een functie `Student **read_students(int n)` die automatisch deze gegevens voor `n` studenten uit de console inleest en een array teruggeeft bestaande uit pointers naar deze structs. Als de gebruiker de input `56789 99.9 59876 11.1` ingeeft in de console, zou `read_students` dus een array van twee pointers naar studenten moeten teruggeven. De eerste student heeft als id `56789` en een gemiddelde van `99.9` en de tweede student heeft als id `59876` en een gemiddelde van `11.1`.

5 Datastructuren

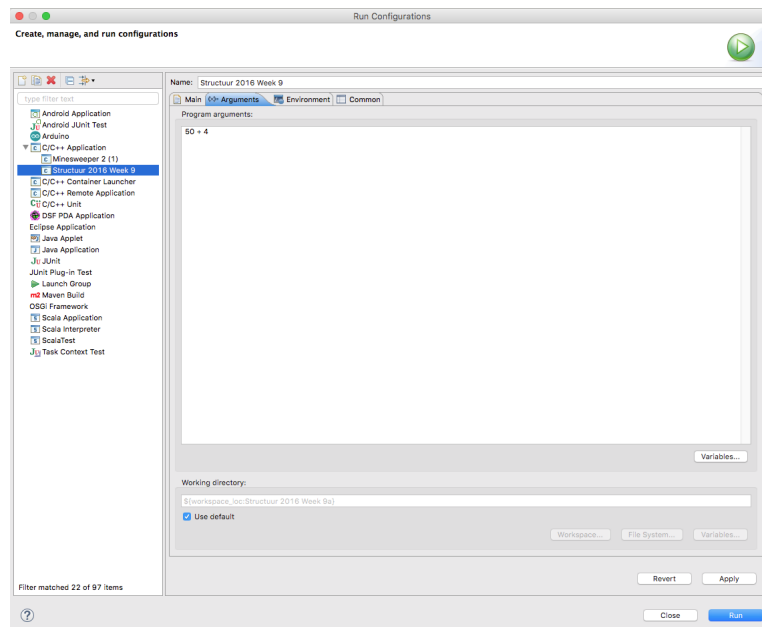
5. (**Dodona**) Implementeer de functie `Node *insert_node(Node *start, int value)` die een nieuwe node met waarde `value` op de juiste plaats in de gegeven lijst toevoegt zodat in deze lijst alle waarden gesorteerd van klein naar groot worden bijgehouden. De functie geeft de (mogelijk nieuwe) start van de gesorteerde lijst terug.

6 Extra

6. (**Dodona**) Een double linked list is een datastructuur waarin elke node niet alleen zijn volgende, maar ook zijn vorige buur bijhoudt. De definitie van die struct wordt al gegeven op Dodona en in `oefeningen.c`. Implementeer nu een double linked list van integers met de volgende functies:
 - (a) `Double_Node *insert_before(int value, Double_Node *node)`: deze functie neemt een node en een integer als input en voegt de integer toe aan de lijst vóór deze node. Als resultaat geeft de functie de nieuw aangemaakte node terug.
 - (b) `Double_Node *insert_after(int value, Double_Node *node)`: deze functie neemt een node en een integer als input en voegt de integer toe aan de lijst ná deze node. Als resultaat geeft de functie de nieuw aangemaakte node terug.
 - (c) `Double_Node *delete_node(int value, Double_Node *start)`: deze functie neemt de startnode van de lijst en een integer als input en verwijdert de eerste node die overeenkomt met de gegeven integer. Als resultaat geeft de functie de (nieuwe) start van de lijst terug. Let op voor memory leaks.

Let er bij het implementeren van elk van deze functies op dat de `node` of `start` parameter een `NULL` pointer kan zijn.

Een command-line argument meegeven in Eclipse doe je door bv. eerst je programma gewoon uit te voeren, en daarna te navigeren naar **Run** → **Run Configurations...** → **C/C++ application**, dan op de naam van je project te klikken en dan op de **Arguments** tab. Daarna kan je in het **Program Arguments** tekstveld de argumenten schrijven, zoals te zien is in de afbeelding hieronder.



In CLion voeg je command-line argumenten toe door naar Run → Edit Configurations te gaan. Klik op de applicatie die overeenkomt met je project en voeg je argumenten toe aan het **Program arguments** veld.

