



Academie voor Technology, Innovation &
Society Delft
Academie voor ICT & Media

Gestructureerd programmeren in C

GESPRG: 2 dimensionale array's, foutcontrole en strings

DE HAAGSE
HOGESCHOOL

2 dimensionale array

- Array van array's

```
int i = 31;
```

i

31

```
int a[4] = {7, 23, 6, 2};
```

a

0	1	2	3
7	23	6	2

```
int m[2][4] = { {1, 1, 61, 3}, {4, 23, 7, 0} };
```

m

	0	1	2	3
0	1	1	61	3
1	4	23	7	0

2 dimensionale array

- Array van array's

```
int m[2][4] = { {1, 1, 61, 3}, {4, 23, 7, 0} };
```

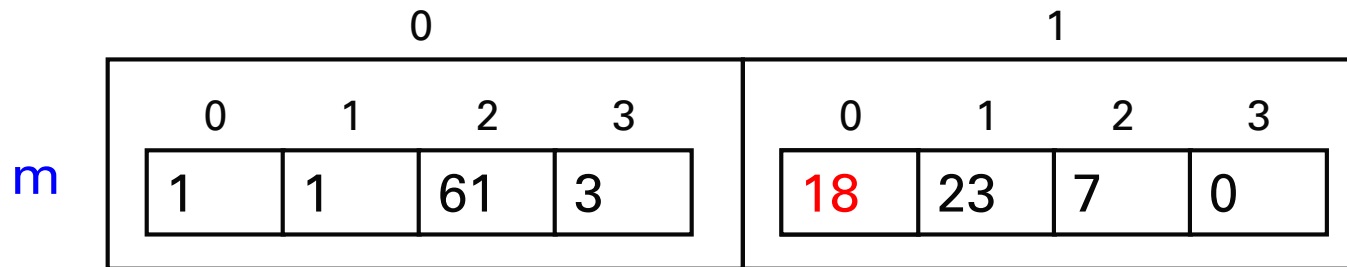


Diagram illustrating the array `m` in row-major order, showing the row and column indices.

	0	1	2	3
0	1	1	61	3
1	18	23	7	0

`m[1][0] = 18;`

Row-major order

Sudoku

8	6			2				
			7				5	9
				6		8		
	4							
		5	3					7
	2					6		
		7	5		9			

```
int sudoku[9][9] = {  
    {8 ,6 ,0 ,0 ,2 ,0 ,0 ,0 ,0},  
    {0 ,0 ,0 ,7 ,0 ,0 ,0 ,5 ,9},  
    {0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0},  
    {0 ,0 ,0 ,0 ,6 ,0 ,8 ,0 ,0},  
    {0 ,4 ,0 ,0 ,0 ,0 ,0 ,0 ,0},  
    {0 ,0 ,5 ,3 ,0 ,0 ,0 ,0 ,7},  
    {0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0},  
    {0 ,2 ,0 ,0 ,0 ,0 ,6 ,0 ,0},  
    {0 ,0 ,7 ,5 ,0 ,9 ,0 ,0 ,0}  
};
```

Alvast wat huiswerk (deel 1)

- Schrijf een functie `areRowsValid` die controleert of in een matrix van 9 bij 9 niet meerdere malen hetzelfde cijfer op dezelfde **regel** voorkomt. Een lege plaats wordt aangegeven met de waarde `0`.
 - De functie geeft `1` terug als er niet meerdere malen hetzelfde cijfer op dezelfde rij voorkomt.
 - De functie geeft `0` terug als er meerdere malen hetzelfde cijfer op dezelfde rij voorkomt.



Domein (van een functie)

- Faculteit (met `int`)
 - Domein = $[0..12]$
- n boven k (met `int`'s)
 - Domein versie met faculteit = $[0..12]$
 - Domein recursieve versie = $[0..33]$

Ligt een argument in het domein?

- Aanroeper (caller)

- De **aanroeper** moet voor aanroep van `faculteit(i)` controleren of `i` in het domein van de functie valt.
 - Wat moet de functie doen als `i` niet in het domein valt?

Niets! Misschien wel tijdens ontwikkelfase...
Gebruik `assert` (zie volgende sheet).

- Aangeroepeene (callee)

- De **functie** moet na aanroep van `faculteit(i)` controleren of `i` in het domein van de functie valt.
 - Wat moet de functie doen als `i` niet in het domein valt?

Foutmelding geven? Waar (denk aan embedded systeem)?

Foutcode teruggeven? **Caller moet checken!**

assert

```
#include <assert.h>
```

```
int faculteit(int n) {  
    int i, res = 1;  
    assert(n >= 0 && n <= 12);  
    for (i = 1; i <= n; i = i + 1) {  
        res = res * i;  
    }  
    return res;  
}
```

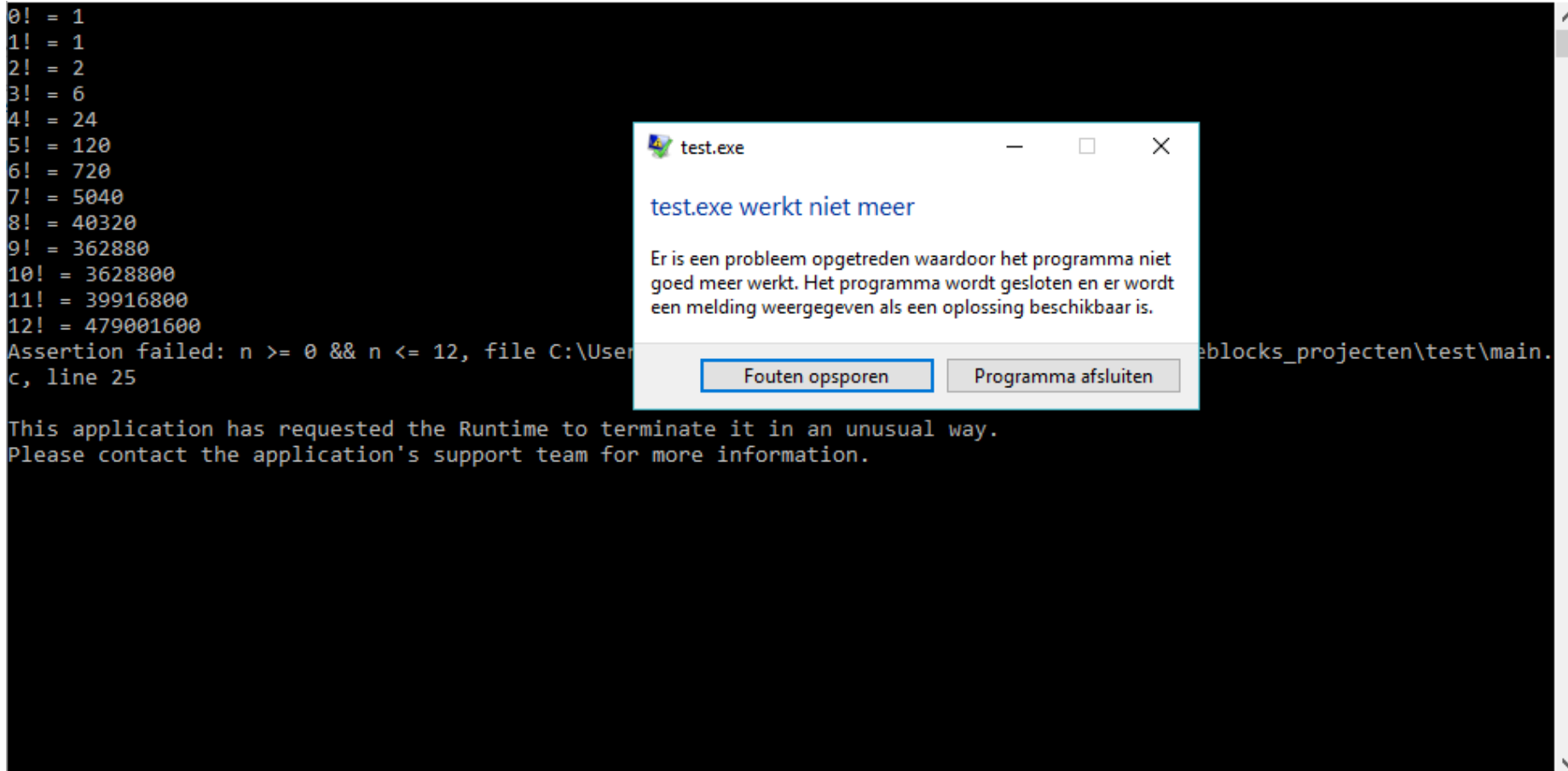
Ik beweer: $0 \leq n \leq 12$



assert tijdens ontwikkelen

- Het aanroepen van functie `faculteit` met argument 13 geeft deze uitvoer:

```
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
11! = 39916800
12! = 479001600
Assertion failed: n >= 0 && n <= 12, file C:\User
c, line 25
This application has requested the Runtime to terminate it in an unusual way.
Please contact the application's support team for more information.
```



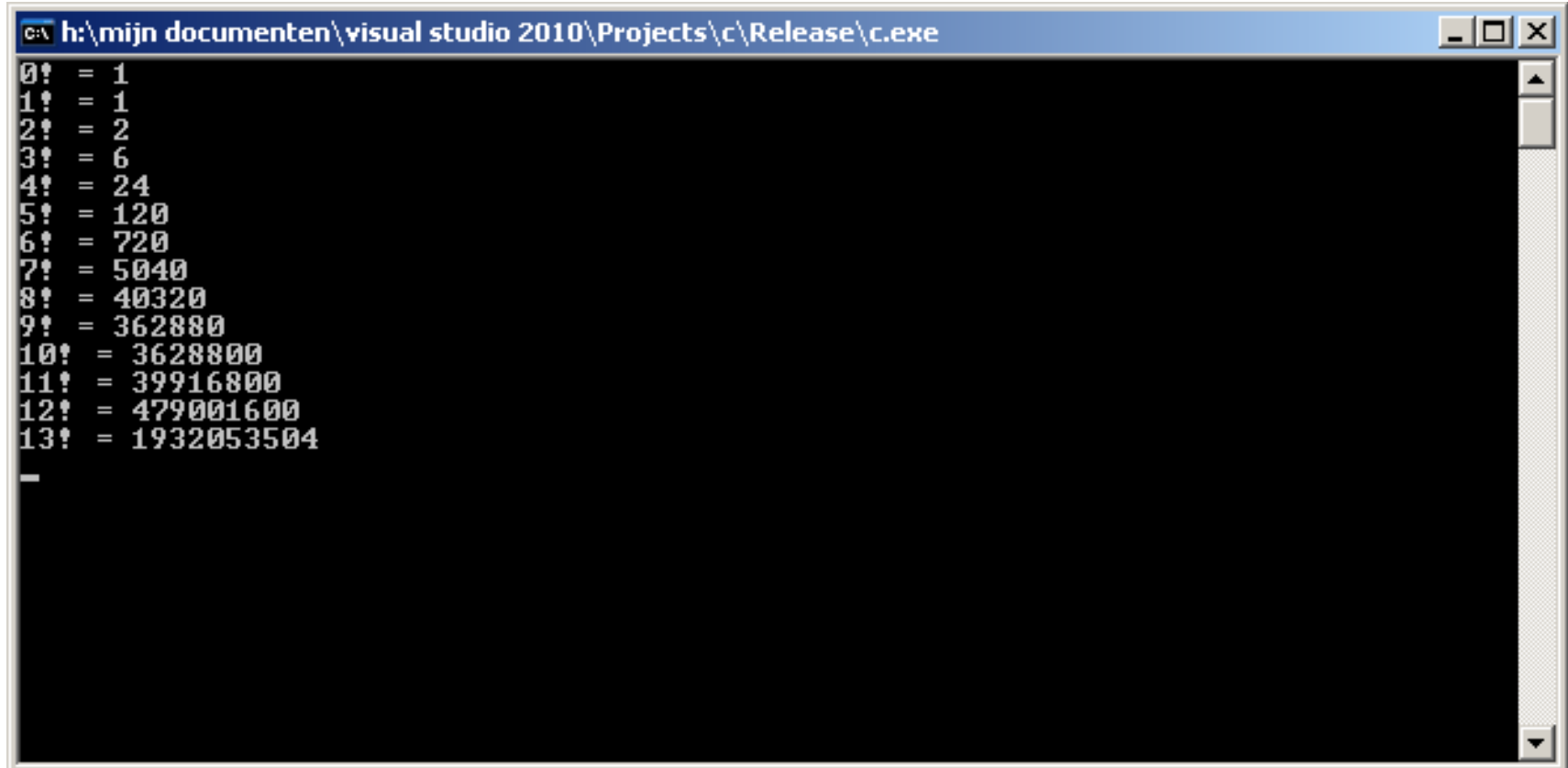
test.exe werkt niet meer

Er is een probleem opgetreden waardoor het programma niet goed meer werkt. Het programma wordt gesloten en er wordt een melding weergegeven als een oplossing beschikbaar is.

Fouten opsporen Programma afsluiten

assert in Release mode

- In software die je uitgeeft (release) wil je niet deze foutmelding hebben.
- Gebruik `#define NDEBUG` om dit te voorkomen.



```
C:\h:\mijn documenten\visual studio 2010\Projects\c\Release\c.exe
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
11! = 39916800
12! = 479001600
13! = 1932053504
-
```

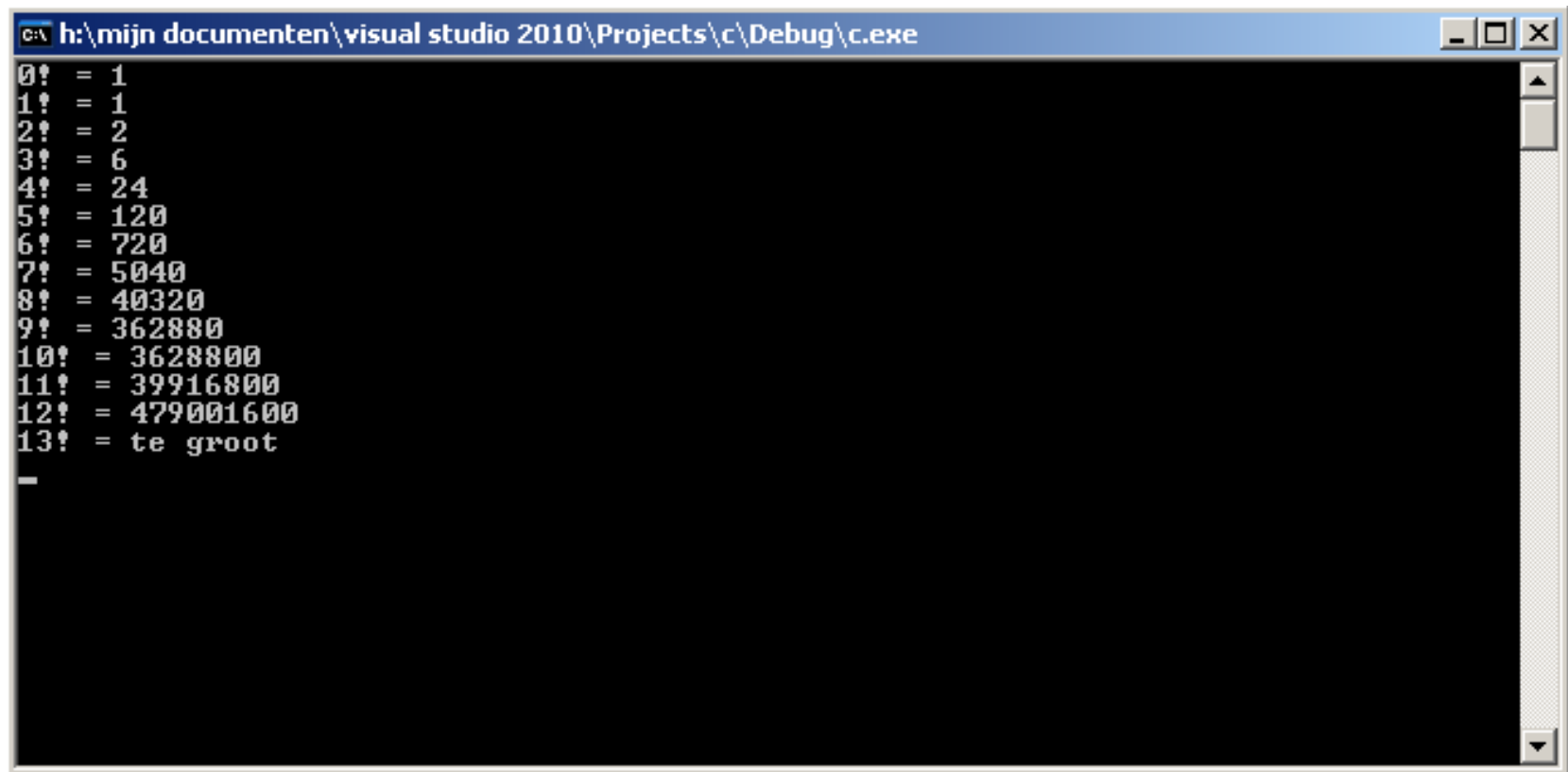
Foutcode

```
int faculteit(int n) {  
    int i, res = 1;  
    if (n >= 0 && n <= 12) {  
        for (i = 1; i <= n; i = i + 1) {  
            res = res * i;  
        }  
    }  
    else {  
        res = -1;  
    }  
    return res;  
}
```

Foutcode controleren

```
int main(void) {
    int n, fac;
    for (n=0; n<=13; n=n+1) {
        fac = faculteit(n);
        if (fac != -1) {
            printf("%d! = %d\n", n, fac);
        }
        else {
            printf("%d! = te groot\n", n);
        }
    }
    getchar();
    return 0;
}
```

Uitvoer



```
h:\mijn documenten\visual studio 2010\Projects\c\Debug\c.exe
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
11! = 39916800
12! = 479001600
13! = te groot
_
```

Karakters

Klassificatie van karakters:

```
#include <ctype.h>
```

```
isalpha(c)    /* is een letter */  
isdigit(c)    /* is een cijfer */  
isalnum(c)    /* is een letter of cijfer */  
isspace(c)    /* is een spatie, tab, newline */  
islower(c)    /* is een kleine letter (lower case) */  
isupper(c)    /* is een hoofdletter (upper case) */  
  
tolower(c)    /* maak kleine letter */  
toupper(c)    /* maak hoofdletter */
```

Karakters

```
#include <stdio.h>
#include <ctype.h>
```

Bibliotheek met karakter functies
<http://en.wikipedia.org/wiki/Ctype.h>

```
int main(void) {
    char c;
    printf("Type een karakter: ");
    scanf("%c", &c);
    if (isalpha(c)) {
        printf("Dat is een letter.\n");
        if (islower(c)) {
            printf("De bijbehorende hoofdletter is: %c\n",
                toupper(c));
        }
    }
    else {
        printf("Dat is geen letter.\n");
    }
    fflush(stdin); getchar(); return 0;
}
```

Karakter variabele

Inlezen karakter variabele

Printen karakter
variabele

```
Type een karakter: h
Dat is een letter.
De bijbehorende hoofdletter is: H
```

Strings in C

- Een rij karakters (inclusief spaties) wordt een string genoemd.
- Een string in C is een array met char's:

```
char greeting[] = { 'H', 'e', 'l', 'l', 'o', '\0' };
```

- Dit mag in C korter worden opgeschreven als:

```
char greeting[] = "Hello";
```

Afsluitkarakter
'\0' = 0x00

- Situatie in het geheugen:

Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x23451	0x23452	0x23453	0x23454	0x23455	0x23456

C strings

```
#include <stdio.h>
#include <string.h>

strlen(s)          /* geef lengte string (zonder '\0') */
strcmp(s1, s2)     /* vergelijk s1 met s2 */
strcpy(s1, s2)     /* kopieer s2 naar s1 */
strcat(s1, s2)     /* plak s2 achter s1 */
strchr(s, c)       /* zoek c in string s (pointer terug) */
printf("%s", s)    /* schrijf string s */
```

Vergelijken twee strings:

```
if (strcmp(s1, s2) == 0) { ... }
```

< 0 als s1 kleiner dan s2
= 0 als s1 gelijk aan s2
> 0 als s1 groter dan s2

C strings

```
#include <stdio.h>
#include <string.h>
```

Bibliotheek met string functies

<http://en.wikipedia.org/wiki/String.h>

```
int main(void) {
    char s[] = "Hallo";
    char *p;
    printf("%s\n", s);
    printf("sizeof(s) = %d\n", sizeof(s));
    printf("strlen(s) = %d\n", strlen(s));
    p = strchr(s, 'a');
    if (p != NULL) {
        *p = 'e';
    }
    printf("%s\n", s);
    fflush(stdin);
    getchar();
    return 0;
}
```

s is een string variabele

Afdrukken string variabele

zoek eerste 'a'
in een string

string variabele wordt
afgesloten met '\0'

```
Hallo
sizeof(s) = 6
strlen(s) = 5
Hello
```

Strings

Bepalen lengte van een string:

```
/* Treat string as array */
int string_length_array(char s[]) {
    int len=0;

    while (s[len] != '\0') {
        len = len + 1;
    }

    return len;
}
```

Huiswerk

- Bestudeer boek/dictaat:
 - 6.11
 - 6.5, 8.6
- Maak opdrachten:
 - 18 en 30 van <https://www.w3resource.com/c-programming-exercises/array/index.php>
 - 2 en 5 van <https://www.w3resource.com/c-programming-exercises/string/index.php>



Academie voor Technology, Innovation &
Society Delft
Academie voor ICT & Media

Gestructureerd programmeren in C

GESPRG: Structs

DE HAAGSE
HOGESCHOOL

Een nieuwe type variabele: struct

- Variabele met een 'vaste' statische structurering
- Een struct bestaat uit één of meerdere variabelen
- De opbouw van een struct:

```
struct naam {  
    members  
};  
  
struct meting_struct {  
    int uur, min, sec;  
    double temp;  
};
```

struct in C

- Members **kunnen** van **verschillende** typen zijn.

- ```
struct meting_struct {
 int uur, min, sec;
 double temp;
} meting;
```

|        | uur | min | sec | temp |
|--------|-----|-----|-----|------|
| meting | 14  |     |     | 13.7 |

- Elk member heeft een **naam** (membername).
- Een member kan benaderd worden met de selectie operator .
  - `meting.temp = 13.7;`
  - `meting.uur = 14;`



# struct in C

---

- Hoe maak je een instantie van een struct declaratie?

*Code:*

```
int main(void) {

 struct student {
 int nummer;
 char naam[20];
 float cijfer;
 };

 struct student s1, s2;

 s1.nummer = 13;
}
```



# struct in C

---

- Hoe maak je een instantie van een struct declaratie?
  - Alternatieve versie

*Code:*

```
int main(void) {

 struct student {
 int nummer;
 char naam[20];
 float cijfer;
 } s1, s2;

 s1.nummer = 13;
}
```

# Structs met array's

- Een struct kan ook array's als member hebben

```
struct Books {
 char title[50]; /* string van 49 karakters */
 char author[50];
 char subject[100];
 int book_id;
};
```

```
int main() {
 struct Books Book1, Book2;
 ...
 return 0;
}
```

# Structs en functies

- Structs kunnen net zoals andere variabelen doorgegeven worden aan functies:

```
struct Books {
 char title[50];
 char author[50];
 char subject[100];
 int book_id;
};
```

```
int main() {
 struct Books Book1, Book2;

 strcpy(Book1.title, "C Programming");
 strcpy(Book1.author, "Nuha Ali");
 strcpy(Book1.subject, "C Programming Tutorial");
 Book1.book_id = 6495407;
 /* Zie volgende slide */
```

# Structs en functies

```
/* Dit is een vervolg van de code die in de main staat */
```

```
printBook(Book1);
```

```
 return 0;
```

```
}
```

```
void printBook(struct Books book) {
 printf("Book title : %s\n", book.title);
 printf("Book author : %s\n", book.author);
 printf("Book subject : %s\n", book.subject);
 printf("Book book_id : %d\n", book.book_id);
}
```

# Structs en functies

- Nu dezelfde functie met call by reference:

*/\* Dit is een vervolg van de code die in de main staat \*/*

```
printpBook(&Book1);
```

```
return 0;
```

```
}
```

adres van Book1

-> Is een speciale operator die gebruikt kan worden bij pointers naar structs

```
void printpBook(struct Books *book) {
 printf("Book title : %s\n", book->title);
 printf("Book author : %s\n", book->author);
 printf("Book subject : %s\n", book->subject);
 printf("Book book_id : %d\n", book->book_id);
}
```

# Array van structs

- Een array kan ook uit structs bestaan:

```
struct Books {
 char title[50]; /* string van 49 karakters */
 char author[50]; /* string van 49 karakters */
 char subject[100]; /* string van 99 karakters */
 int book_id;
};

int main() {
 struct Books boeken[250]; /* maximaal 250 boeken */
 ...
 return 0;
}
```

# Array van structs met array's

- Elk boek is apart te benaderen:

```
strcpy(boeken[0].title, "C Programming");
strcpy(boeken[0].author, "Nuha Ali");
strcpy(boeken[0].subject, "C Programming Tutorial");
boeken[0].book_id = 6495407;
```

```
strcpy(boeken[1].title, "Wiskunde 1");
strcpy(boeken[1].author, "Kees de Joode");
strcpy(boeken[1].subject, "Wiskunde basis");
boeken[1].book_id = 12345;
```

```
c = boeken[1].title[0]; /* c == 'W' */
```

# Huiswerk

- Bestudeer boek/dictaat:
  - 11.3, 11.4
  - 10.1 t/m 10.6
- Maak opdracht:
  - Schrijf een functie die twee complexe getallen bij elkaar optelt. De twee complexe getallen moeten als parameter aan deze functie worden doorgegeven. Een complex getal kan opgeslagen worden in een struct instantie. Een oplossing is te vinden op: <https://www.programiz.com/c-programming/examples/complex-number-add>



# De selectie operator .

---

- In C zijn er twee selectie operatoren
- We gebruiken voor structs de 'direct member selector' .
  - Deze heeft de vorm:
    - `Naam_struct.identificier`
- Er is nog een andere operator die gebruikt wordt i.c.m. structs en pointers
  - De 'indirect member selector' operator ->
  - Deze heeft de vorm:
    - `Naam_pointer->identificier`
- Stel we hebben een struct genaamd **structuur** en een pointer naar deze structuur genaamd **pstructuur**. De struct **structuur** heeft een member genaamd **lid**.
- Het benaderen van **lid** gaat met:
  - `Structuur.lid = 22;`
- Maar kan ook met ->:
  - `Pstructuur->lid = 22;`
  - Let op de *identificier* moet overeenkomstig zijn met één van de leden van de struct!