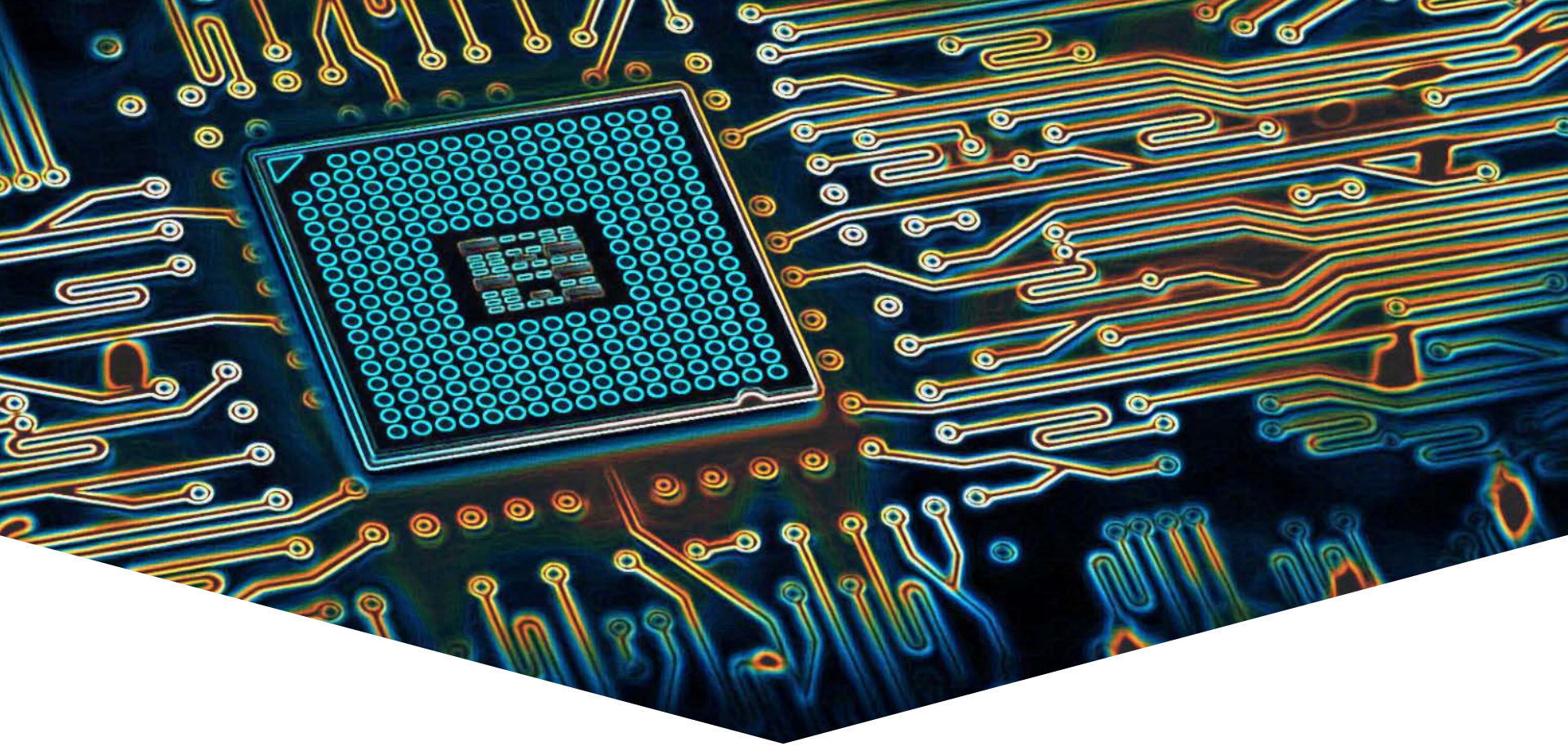


EMS20 Week 2 Lab 1

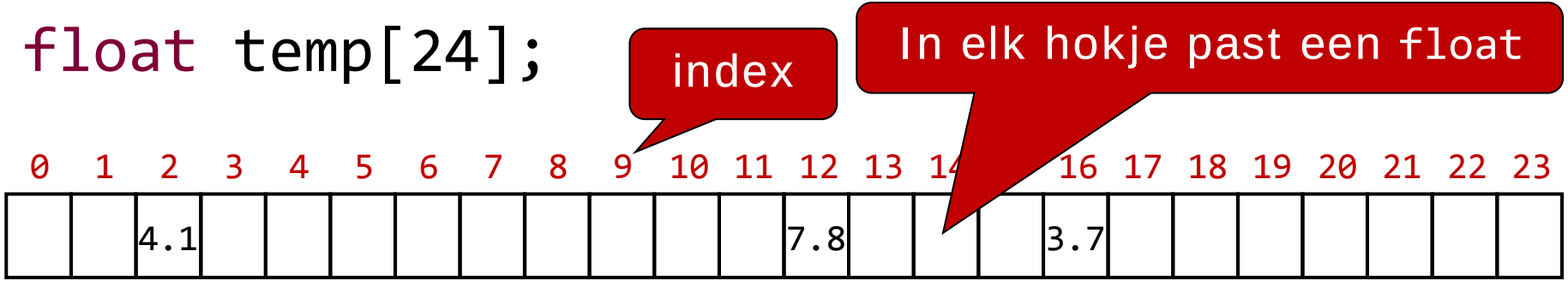
Leerdoelen Week 2 Lab 1

Leerdoelen week 2 lab 1. Je leert hoe je:

- meerdere variabelen van verschillende typen bij elkaar kunt opslaan in één samengestelde variabele. Zo'n samengestelde variabele wordt in C een **struct** genoemd;
- structs kunt opslaan in een array;
- structs kunt doorgeven als argument aan een functie en kunt teruggeven als returnwaarde vanuit een functie;
- een zogenoemd **enumeratie**-type kunt definiëren en gebruiken om je programma's beter leesbaar te maken;
- de compiler kunt vertellen welke versie van de C-standaard je wilt gebruiken.

Array in C (herhaling EMS10)


```
float temp[24];
```

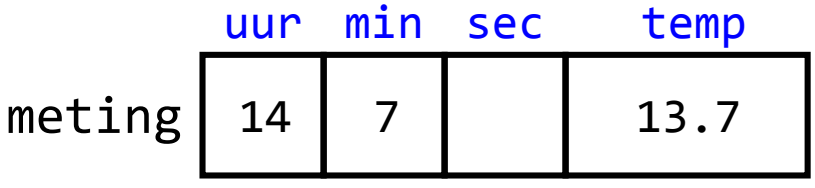


- Elk element heeft een index beginnend bij `0`.
- Elementen zijn van hetzelfde type.
- Element kan geselecteerd worden met indexoperator `[]`:
  - `temp[2] = 4.1;`
  - `temp[16] = 3.7;`
  - `temp[12] = temp[2] + temp[16];`

struct in C

Een struct variabele heeft een naam en een structuur:

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



- Elk element (member) heeft een naam (membername).
- Elementen kunnen van verschillende types zijn.
- Element kan gebruikt worden m.b.v. de selectie-operator .:
  - meting.temp = 13.7;
  - meting.uur = 14;
  - meting.min = meting.uur / 2

typedef

Als een array of `struct` type meerdere keren in een programma wordt gebruikt dan kan de declaratie van het type en de definitie van variabelen worden gesplitst met een `typedef`.

```
typedef double WeekTemp[7];
WeekTemp temperatuur;

typedef struct {
    int uur, min, sec;
    double temp;
} TempMeting;
TempMeting meting;
```

Statische datastructuren

`struct` kan array(s) bevatten.

```
typedef struct {
    char naam[80];
    int punten;
} Deelnemer;
```

Array kan `structs` bevatten.

```
typedef struct {
    Deelnemer speler[100];
    int aantalSpelers;
} Stand;
```

Voorbeeld

Gegeven variabele:

```
Stand s; /* zie vorige sheet */
```

Neem aan dat deze variabele gevuld is met data.

```
s = vulStand();
```

Schrijf een functie om de naam van de speler(s) met de meeste punten af te drukken.

```
printWinnaars(s);
```

Voorbeeld


```
void printWinnaars(Stand st) {
    if (st.aantalSpelers == 0) {
        printf("Er is geen winnaar.\n");
    }
    else {
        int i, max;
        max = st.speler[0].punten;
        for (i = 1; i < st.aantalSpelers; i++) {
            if (st.speler[i].punten > max) {
                max = st.speler[i].punten;
            }
        }
        printf("De winnaar(s) is(zijn):\n");
        for (i = 0; i < st.aantalSpelers; i++) {
            if (st.speler[i].punten == max) {
                printf("%s\n", st.speler[i].naam);
            }
        }
    }
}
```

Zie: https://bitbucket.org/HR_ELEKTRO/ems20/raw/master/Voorbeelden/printWinnaars.c

Enumeratie

Variabelen van een `enum` type kunnen een vast aantal waarden aannemen. Denk aan lijsten als:

- dag van de week (maandag, dinsdag, … , zondag);
- maand van het jaar (januari, februari, … , december);
- richting (noord, oost, zuid, west).

Een `enum` heeft tot doel een programma leesbaarder te maken.

enum voorbeelden


```
enum dag {MA, DI, WO, DO, VRIJ, ZAT, ZON};
```

- Een element uit een enum is een integer constante met een naam.
- De elementen krijgen standaard opvolgende nummers toegewezen: dus *MA* = 0 en *ZON* = 6.
- Tenzij anders aangegeven.

```
enum richting {NOORD, OOST=90, ZUID=180, WEST=270};
```

- Vaak worden elementen uit een enum met HOOFDLETTERS gespeld om verwarring met variabelen te voorkomen.

typedef

Een enum wordt vaak gecombineerd met een typedef:

```
typedef enum dag {MA, DI, WO, DO, VRIJ, ZAT, ZON} Dag;

    Dag afspraak;
    afspraak = DO;
    switch(afspraak)
    {
        case MA: //...
        case DI: //...
```

Een enum element is een integer

Een enum element is een integer en geen string!

```
//printf("%s", afspraak); // Kan niet!
printf("%d", afspraak); // Print een getal.
```

enum ➔ string


```c
#include <stdio.h>

typedef enum {MA, DI, WO, DO, VRIJ, ZAT, ZON} dag;
const char * dagnaam[] = {
    "maandag", "dinsdag", "woensdag",
    "donderdag", "vrijdag", "zaterdag", "zondag"
};

int main(void)
{
    dag afspraak = DO;

    printf("De afspraak is op %s!", dagnaam[afspraak]);

    return 0;
}
```

Zie: https://bitbucket.org/HR_ELEKTRO/ems20/raw/master/Voorbeelden/enum2string.c

Volgende lab…

MSP430 Assembler.

Neem je EXP430G2 LaunchPad met de MSP430G2553 microcontroller mee!

Aan de slag!

Aan de slag met Opdrachten_Week_2_Lab_1.pdf