

Proeftoets INFORM2-T1

Structuren

Opgave 1

Een geautomatiseerd weerstation slaat elk uur een aantal belangrijke weersinvloeden op:

1. Temperatuur (1 decimaal significant)
2. Luchtvochtigheid (geheel getal als %)
3. Windrichting (in gehele graden)

De tijd van de meting (als een struct met jaar, uur, minuut als 3 gehele getallen)

Creëer een typedef voor de bovenstaande struct waarin de gemeten waarden kunnen worden opgeslagen.

Maak in main() een declaratie naar een array met **24** elementen van de bovenstaande struct, zodat een geheel etmaal aan gegevens kan worden opgeslagen.

Opgave 2

Maak een programma waarmee de bovenstaande array (je mag ervan uitgaan dat de array al met meetwaarden is gevuld) onder elkaar wordt afgedrukt, zie voorbeeld

Meting 1

Tijd: 2012 00:01

Temp: 21.7

Luchtvochtigheid: 68

Windrichting: 254

Meting 2

Tijd: 2012 01:00

Temp: 20.3

Luchtvochtigheid: 69

Windrichting: 256

enzovoorts

Pointers

Opgave 3

Gegeven: de onderstaande declaraties:

```
int vrz[100], teller, *v;
```

Maak een programma waarmee de gehele array iteratief wordt gevuld met de waarde 23.

Doe dit m.b.v. array indexering.

Opgave 4

Vul de gehele array nu met de waarde 76.

Maar maak nu gebruik van een pointer variabele.

Bitwise operatoren

Opgave 5

Gegeven: de onderstaande declaratie:

```
unsigned char a = 140;
```

Bepaal m.b.v. bitwise operatoren of de bits 3 en 7 zijn “geset” (een 1 bevatten), begin te tellen vanaf 0 én vanaf rechts.

Opgave 6

“Set” vervolgens de bits 5 en 6 zonder de andere bits aan te tasten.

Gelinkte Lijst

Opgave 7

Verander opdracht 1 van deze toets nu zodat er geen array maar een gelinkte lijst wordt gebruikt. Je mag zelf bepalen of je de nieuwe nodi voor- of achteraan de lijst plakt. Indien noodzakelijk mag je de struct aanpassen en zo nodig ook variabelen toevoegen.

Opgave 8

Druk de inhoud van de gelinkte lijst af zodat dit identiek is aan het voorbeeld in opgave 1, d.w.z. de lijst wordt afgedrukt van meting 1 tot en met meting 24, dus NIET achterstevoren !

Als je daarvoor extra variabelen nodig hebt mag je die extra declareren, maar geen onzin variabelen toepassen !

Uitwerking opgaven 1 + 2

```
/* Proeftoets

    Opgave 1 + 2
*/

#include <stdio.h>

typedef struct
{
    unsigned jaar;
    unsigned uren;
    unsigned minuten;
} Datum;

typedef struct
{
    Datum datum;
    float temperatuur;
    unsigned luchtvocht;
    unsigned graden;
} Meting;

int main(void)
{
    Meting meetwaarden[24];
    int teller;

    for(teller=0; teller<24; teller++)
    {
```

```
printf("+++++\n");

printf("meting %d\n", teller+1);

printf("Tijd %4d %2d:%2d\n", meetwaarden[teller].datum.jaar,
meetwaarden[teller].datum.uren, meetwaarden[teller].datum.minuten);

printf("Temperatuur: %4.1f\n",
meetwaarden[teller].temperatuur);

printf("Luchtvochtigheid: %2d \n",
meetwaarden[teller].luchtvocht);

printf("Windrichting: %3d\n", meetwaarden[teller].graden);

}

return 0;

}
```

Uitwerking opgaven 3 + 4

/* Proeftoets

Opgave 3 +4

Gegeven: de onderstaande declaraties:

```
int vrz[100], teller, *v;
```

Maak een programma waarmee de gehele array iteratief wordt gevuld met de waarde 23.

Doe dit m.b.v. array indexering.

```
*/
```

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

```
int main(void)
```

```
{
```

```
    int vrz[100], teller, *v;
```

```
    // Opgave 3
```

```
    for(teller=0; teller<100; teller++)
```

```
        vrz[teller] = 23;
```

```
    // Opgave 4
```

```
    v=vrz; // let op ! arraynaam is al een pointerreferentie
```

```
    for(teller=0; teller<100; teller++)
```

```
        *(v+teller) = 76;
```

```
    return 0;
```

```
}
```

Uitwerking opgaven 5 + 6

```
/*
```

```
Opgave 5
```

```
Gegeven: de onderstaande declaratie:
```

```
unsigned char a = 140;
```

```
Bepaal m.b.v. bitwise operatoren of de bits 3 en 7 zijn "geset" (een 1 bevatten), begin te tellen vanaf 0 én vanaf rechts.
```

```
"Set" vervolgens de bits 5 en 6 zonder de andere bits aan te tasten.
```

```
*/
```

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

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    unsigned char a = 140;
```

```
// Opgave 5
```

```
    // creeer het masker
```

```
    unsigned char masker3 = 1<<3;
```

```
    unsigned char masker7 = 1<<7;
```

```
    if(a & masker3)
```

```
        printf("bit 3 is een 1\n");
```

```
    else
```

```
        printf("bit 3 is een 0\n");
```

```
    if(a & masker7)
```

```
        printf("bit 7 is een 1\n");
```

```
    else
```

```
printf("bit 7 is een 0\n");

// Opgave 6

unsigned char masker5 = 1<<5;
unsigned char masker6 = 1<<6;

a |= (masker5 | masker6);

printf("a heeft de waarde %d", a);

return 0;
}
```

Uitwerking opgaven 7 + 8

/*

Verander opdracht 1 van deze toets nu zodat er geen array maar een gelinkte lijst wordt gebruikt. Je mag zelf bepalen of je de nieuwe nodi voor- of achteraan de lijst plakt. Indien noodzakelijk mag je de struct aanpassen en zo nodig ook variabelen toevoegen.

Opgave 8

Druk de inhoud van de gelinkte lijst af zodat dit identiek is aan het voorbeeld in opgave 1, d.w.z. de lijst wordt afgedrukt van meting 1 tot en met meting 24, dus NIET achterstevoren !

Als je daarvoor extra variabelen nodig hebt mag je die extra declareren, maar geen onzin variabelen toepassen !

*/

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

```
#include <stdlib.h>
```

```
#include <malloc.h>
```

```
typedef struct
```

```
{
```

```
    unsigned jaar;
```

```
    unsigned uren;
```

```
    unsigned minuten;
```

```
} Datum;
```

```
typedef struct meetwaarden
```

```
{
```

```
    Datum datum;
```

```
    float temperatuur;
```

```
    unsigned luchtvocht;
```

```
    unsigned graden;
```

```
    struct meetwaarden *volgende;
```

```
} Meting;
```



```

int main(void)
{
    Meting *kop, *loper, *temp;

    int teller;

    // Creeer eerst een LEGE lijst;
    kop=NULL;

    // Voeg 24 elementen toe aan de lijst
    for(teller=0; teller<24; teller++)
    {
        // Alloceer geheugen voor een element
        temp=(Meting *)malloc(sizeof(Meting));

        // Sluit element af
        temp->volgende=kop;

        // Vul meetwaarden in
        temp->temperatuur=18.7;

        // enz.

        // Plak aan het begin van de lijst
        kop=temp;
    }

    // omdat we lui zijn geweest met invullen van de lijst moeten we deze nu
    inlezen in een array

    // en vervolgens omgekeerd afdrukken

    // declareer array
    Meting arr[24];

```

```

// Bewaar het begin van de lijst, ZEER ESSENTIEEL !!!

loper = kop;

for(teller=0; teller<24; teller++)
{
    // Lees de handel in de array, LET OP arr puntjes gebruiken, lijst
    pijltjes gebruiken !

    arr[teller].temperatuur = loper->temperatuur;

    // enz.

    // Schuif door naar het volgende element in de lijst

    // Dit gaat goed omdat we zeker weten dat we 24 elementen hebben, zo
    niet dan moeten

    // we controleren of we het einde van de lijst hebben bereikt !

    loper=loper->volgende;
}

// Druk nu in omgekeerde volgorde af

for(teller=23; teller>=0; teller--)
{
    printf("+++++\n");

    printf("meting %d\n", 24-teller);

    printf("Tijd %4d %2d:%2d\n", arr[teller].datum.jaar,
arr[teller].datum.uren, arr[teller].datum.minuten);

    printf("Temperatuur: %4.1f\n", arr[teller].temperatuur);

    printf("Luchtvochtigheid: %2d \n", arr[teller].luchtvocht);

    printf("Windrichting: %3d\n", arr[teller].graden);
}

return 0;
}

```