

ProgKI oefententamen 1 nakijkvel
(31 punten)

(10 punten) Menselijke compiler

0. (1 punt) b
1. (1 punt) d
2. (1 punt) b
3. (1 punt) a
4. (1 punt) d
5. (1 punt) b
6. (1 punt) c
7. (1 punt) e
8. (1 punt) d
9. (1 punt) b

(3 punten) Algoritmes

10. (1 punt) d
11. (1 punt) a
12. (1 punt) d

(7 punten) Cijfers

Syntactische fouten slaan we hier grotendeels over. Behalve als we door zo'n fout niet kunnen beoordelen of de gewenste kennis aanwezig is. Bijvoorbeeld: een missende ; gewoon negeren, een loop met compleet andere opbouw ala `for (i = 0 to 9)` is fout.

Als het werkt is het goed! Kijk hier niet naar design ;)

Let op, niet alle onderstaande oplossingen hoef je te kennen. Ze zijn enkel ter referentie.

13. Oplossingen zijn hier bijvoorbeeld:

```
// deze
float gemiddelde(float deelcijfers[], int n)
{
    float som = 0;
    for (int i = 0; i < n; i++)
    {
        som += deelcijfers[i];
    }
    return som / n;
}
```

```
// maar ook
float gemiddelde(float deelcijfers[], int n)
{
    float som = 0;
    while (n > 0)
    {
        n -= 1;
        som += deelcijfers[n];
    }
    return som / n;
}
```

```
// of
float gemiddelde(float deelcijfers[], int n)
{
    float gem = 0;
    for (int i = 0; i < n; i++)
    {
        gem += deelcijfers[i] / n;
    }
    return gem;
}
```

- (a) (1 punt) Idee is correct (een loop, een variabele om in op te tellen, een berekening om het gemiddelde te berekenen)
- (b) (1 punt) Types kloppen (geen tussentijdse afronding!)
- (c) (1 punt) Code is correct (compileert minus syntactische fouten, en zou de juiste uitkomsten genereren)

14. Oplossingen zijn hier bijvoorbeeld:

```
// deze
int geslaagde_studenten(int m, int n, float cijfers[m][n])
{
    int teller = 0;
    for (int i = 0; i < m; i++)
    {
        if (gemiddelde(cijfers[i], n) >= 5.5)
        {
            teller++;
        }
    }
    return teller;
}

// maar ook
int geslaagde_studenten(int m, int n, float cijfers[m][n])
{
    int teller = 0;
    for (int i = 0; i < m; i++)
    {
        int gem = 0;
        for (int j = 0; j < n; j++)
        {
            gem += cijfers[i][j] / n;
        }
        if (gem >= 5.5)
        {
            teller++;
        }
    }
    return teller;
}

// of
int geslaagde_studenten(int m, int n, float cijfers[m][n])
{
    int teller = 0;
    while (m > 0)
    {
        teller += gemiddelde(cijfers[--m], n) >= 5.5;
    }
    return teller
}
```

- (a) (1 punt) Idee is correct (een loop (of twee), een variabele om te tellen)
- (b) (1 punt) Twee dimensionale array wordt correct gebruikt (juiste indices)
- (c) (1 punt) Correct gebruik gemaakt van of een geneste loop of de functie `gemiddelde`
- (d) (1 punt) Code is correct (compileert minus syntactische fouten, en zou de juiste uitkomsten genereren)

(4 punten) Gokverslaving

Kijk ook hier niet naar design. Als het probleem wordt opgelost is het goed!

15. Er zijn hier twee fouten. De eerste vind je op regel 10, daar wordt een random getal gegenereerd tussen de 0 en `MAX_INT`, terwijl dit tussen de 0 en 99 moet zijn. Een oplossing hiervoor is regel 10 vervangen door

```
int antwoord = rand() % 99;
```

De tweede fout vind je op regel 16, daar gaat de do-while loop door zolang `gok` gelijk is aan `antwoord`. Dat is precies omgekeerd! Je moet kunnen blijven gokken als `gok` niet gelijk is aan `antwoord`. Een oplossing hiervoor is regel 16 vervangen door

```
} while(gok != antwoord);
```

- (a) (2 punten) Per correct gespotte fout + regelnummer 1 punt
(b) (2 punten) Per correcte oplossing 1 punt (dit mag pseudocode of enkel uitleg zijn)

(4 punten) Errors

16. De error geeft aan dat `printf()` niet gedefinieerd is. Om te zorgen dat dit wel gebeurt moet de bijbehorende library worden ge-include. `#include <stdio.h>` toevoegen bovenaan het bestand zou het probleem oplossen.
- (a) (1 punt) De fout wordt correct uitgelegd (oorzaak is genoeg)
(b) (1 punt) Er wordt een correcte oplossing aangedragen (dit mag pseudocode of enkel uitleg zijn)
17. De error geeft aan dat `add()` niet gedefinieerd is. Echter staat de definitie wel in de code. Helaas wordt de code van boven naar beneden gecompileerd, en hierbij komt de compiler eerder de aanroep naar `add()` tegen dan de definitie. Een prototype ala `int add(int a, int b);` toevoegen bovenaan het bestand zou het probleem oplossen. Een andere oplossing is de hele functie `add()` naar boven `main()` verplaatsen.
- (a) (1 punt) De fout wordt correct uitgelegd (oorzaak is genoeg)
(b) (1 punt) Er wordt een correcte oplossing aangedragen (dit mag pseudocode of enkel uitleg zijn)

(3 punten) Lastig lezen

18. (1 punt) Enkel deze regels zijn genoemd: 9, 10, 14 t/m 24
19. (2 punten) Per ander verbeterpunt 1 punt (dit mag niet indentatie zijn en ze moeten van elkaar verschillen). Hier zijn veel verschillende antwoorden mogelijk, denk aan:
- (a) let op spaties rondom operatoren
(b) voeg comments toe
(c) voeg een header comment toe
(d) verzin een betere naam dan `x`