

Programmeren KI

Oefententamen 1

November 2017

Naam:

Studentnummer:

Tijd: 2 uur

Maximaal aantal punten: 31

Menselijke compiler (10 punten)

0. (1 punt) Stel, je studeert cum laude af als je gemiddelde tenminste een 8 is, en je nooit lager dan een 7 hebt gehaald voor een vak. Welke C-expressie defineert cum laude afstuderen?

- (a) `!(laagste_cijfer > 7) && gemiddelde >= 8`
- (b) `!(laagste_cijfer < 7) && gemiddelde >= 8`
- (c) `laagste_cijfer > 7 && gemiddelde >= 8`
- (d) `laagste_cijfer >= 7 && gemiddelde > 8`

1. (1 punt) Wat zijn de waarden van de variabelen `x` en `y` nadat het volgende stukje code is uitgevoerd?

```
int x = 10;
int y = x / 5 + x / 6;
x = x / 0.5;
```

- (a) `x = 0, y = 3`
- (b) `x = 5, y = 3.66666`
- (c) `x = 20.0, y = 5.25`
- (d) `x = 20, y = 3`

2. (1 punt) Wat zal de waarde van `c` worden na het uitvoeren van dit stukje code?

```
int a = 1;
int b = 2;
int c = 3;
if (a < b)
{
    c = b;
}
else if (a > b)
{
    c = a;
}
a = 4;
b = 5;
```

- (a) `c = 1`
- (b) `c = 2`
- (c) `c = 3`
- (d) `c = 4`
- (e) `c = 5`

3. (1 punt) Welke van de volgende waarden voor de variabelen `a`, `b` en `c` zullen ervoor zorgen dat bij het uitvoeren van het volgende stuk code de variabele `answer` dezelfde waarde heeft als variabele `c`? Alle variabelen zijn van het type integer.

```
int answer;
if (a >= b) {
    if (b > c || a == b) {
        answer = a;
    }
    else {
        answer = c;
    }
}
else {
    answer = b;
}
```

- (a) `a=3, b=2, c=2`
 - (b) `a=0, b=1, c=2`
 - (c) `a=2, b=2, c=3`
 - (d) `a=2, b=1, c=0`
4. (1 punt) Wat is de uitkomst ofwel het waarschijnlijke doel van het volgende stukje code?

```
int numbers[] = {10, 3, 5, -4, -1, -2};
int n = 6;
int result = 0;
for (int i = 0; i < n; i++) {
    int number = numbers[i];
    if (number < 0) {
        number = 0;
    }
    result += number;
}
printf("%d\n", result);
```

- (a) Om de positieve getallen in de array te tellen
- (b) Om alle negatieve getallen uit de array te halen
- (c) Om alle negatieve getallen in de array naar 0 te veranderen
- (d) Om de som van de positieve getallen in de array te berekenen

5. (1 punt) Wat is de uitkomst van het volgende stukje code?

```
printf("%d", 'c' - 'd');
```

- (a) "d - a"
- (b) -1
- (c) 0
- (d) nothing, compile error

6. (1 punt) Wat is de uitkomst van het volgende stukje code?

```
char s[] = "Jesse";  
for (int i = 0, n = strlen(s) - 1; i < n; i++)  
{  
    if (s[i] == 'e' && s[i + 1] == 's') {  
        s[i] = 'x';  
    }  
}  
printf("%s", s);
```

- (a) x
- (b) Jxssx
- (c) Jxsse
- (d) nothing, compile error

7. (1 punt) Wat is de uitkomst van het volgende stukje code?

```
int x = 3 * 3;  
if (x > 5) {  
    int y = 5;  
}  
printf("%d", x + y);
```

- (a) "x + y"
- (b) 11
- (c) 8
- (d) 14
- (e) nothing, compile error

8. (1 punt) Wat mag er in plaats van de ... staan? Selecteer het meest volledige antwoord.

```
... divide(int a, float b)
{
    return a / b;
}
```

- (a) void
- (b) int
- (c) float
- (d) zowel int als float
- (e) zowel int als float als void

9. (1 punt) Wat print het volgende stukje code?

```
#include <stdio.h>
void foo(int a)
{
    if (a <= 0)
    {
        return;
    }
    printf("%d", a);
    foo(a - 1);
}
int main(void)
{
    foo(3);
}
```

- (a) 3
- (b) 321
- (c) 3210
- (d) 123
- (e) nothing, compile error

Algoritmes (3 punten)

10. (1 punt) Wat is de runtime-complexiteit van bubble sort voor een array van grootte n in het slechtste geval?

- (a) $O(n)$
- (b) $O(2 * n)$
- (c) $O(2^n)$
- (d) $O(n^2)$
- (e) $O(n * \log(n))$

11. (1 punt) Wat is de runtime-complexiteit van linear search voor een array van grootte n in het beste geval?

- (a) $\Omega(1)$
- (b) $\Omega(\log(n))$
- (c) $\Omega(n)$
- (d) $\Omega(n^2)$

12. (1 punt) Wat is de runtime-complexiteit van het volgende stukje code in het slechtste geval?

```
string name = get_string();
for (int i = 0, n = strlen(name); i < n; i++)
{
    for (int j = 0; j < n; j++) {
        printf("%c", name[j]);
    }
    printf("\n");
}
```

- (a) $O(1)$
- (b) $O(n)$
- (c) $O(2^n)$
- (d) $O(n^2)$
- (e) $O(n * \log(n))$

Cijfers (7 punten)

Stel dat een student slaagt voor een vak als haar gemiddelde hoger of gelijk is aan een 5.5. Voor de administratie wil de universiteit weten hoeveel studenten een vak hebben gehaald. In plaats van alle deeltijfers te tellen en een heel hoop handwerk, kunnen we hier een programma voor schrijven.

13. (3 punten) Allereerst hebben we een functie nodig die het gemiddelde cijfer van een student berekent. Maak de functie hieronder af. Deze accepteert twee argumenten: `deeltijfers` een array van cijfers (1.0 t/m 10.0), en `n` de lengte van de array `deeltijfers`. De functie moet het gemiddelde cijfer berekenen en `returnen`.

```
float gemiddelde(float deeltijfers[], int n)
{
```

14. (4 punten) Nu kunnen we een functie schrijven die telt hoeveel studenten het vak hebben gehaald. Alle behaalde deelcijfers van een vak zitten in een twee dimensionale array genaamd `cijfers` van grootte `m x n`. Waar `m` het aantal studenten is dat het vak volgt, en `n` het aantal deelcijfers van het vak. Ter verduidelijking: `cijfers[2][1]` geeft van de derde student het tweede deelcijfer.

Maak de functie hieronder af. Deze accepteert drie argumenten: `m` het aantal studenten dat het vak volgt, `n` het aantal deelcijfers van het vak, en `cijfers` een twee dimensionale array (`m x n`). De functie moet het aantal studenten dat het vak heeft gehaald als `int` `return`en. Je mag bij deze vraag aannemen dat de functie `gemiddelde` van vraag 13 bestaat, is geïmporteerd, en werkt.

```
int geslaagde_students(int m, int n, float cijfers[m][n])
{
```


Gokverslaving (4 punten)

Hieronder staat code voor een interactief gokspel. De gebruiker moet een getal tussen de 0 en 99 raden. Als de gebruiker het getal niet raadt, mag zij blijven raden tot ze het getal heeft geraden. Helaas werkt het niet helemaal.

```
loterij.c
1) #include <stdlib.h>
2) #include <stdio.h>
3) #include <cs50.h>
4) #include <time.h>
5)
6) int main(void)
7) {
8)     int gok;
9)     srand((unsigned) time(NULL));
10)    int antwoord = rand();
11)
12)    do
13)    {
14)        printf("Gok een getal tussen de 0 en 99: ");
15)        gok = get_int();
16)    } while (gok == antwoord);
17)
18)    printf("Je hebt het geraden!\n");
19)    printf("Jouw gok was %d en het correcte antwoord was %d!\n", gok, antwoord);
20) }
```

Bij het spelen van het spel gaan er een aantal dingen mis. Dit kun je zien in de volgende test van de code:

```
~/workspace/ $ ./loterij
Gok een getal tussen de 0 en 99: 3
Je hebt het geraden!
Jouw gok was 3 en het correcte antwoord was 794882860!
```

15. (4 punten) Spoor de twee fouten in de code op. Meld het regelnummer, de fout, en hoe je deze fout zou oplossen. Voor de volledigheid, de functie `rand()` genereert een pseudorandom getal tussen de 0 en de maximale waarde van een integer. De functie `srand()` seed de pseudorandom generator.

Errors (4 punten)

Het volgende programma compileert niet:

```
hello.c
1) int main(void)
2) {
3)     printf("Hello, world!");
4) }
```

De compiler geeft de volgende error bij het compileren:

```
~/workspace/ $ make hello
clang hello.c -Werror -lcrypt -lcs50 -lm -o hello
hello.c:3:5: error: implicitly declaring library function 'printf'
with type 'int (const char *, ...)' [-Werror]
printf("Hello, world!");
^
make: *** [hello] Error 1
```

16. (2 punten) Leg uit wat er fout gaat, en hoe je dit kan oplossen.

Het volgende programma compileert niet:

```
foo.c
1) int main(void)
2) {
3)     add(3, 5);
4) }
5)
6) int add(int a, int b)
7) {
8)     return a + b;
9) }
```

De compiler geeft de volgende error bij het compileren:

```
~/workspace/ $ make foo
clang foo.c -Werror -lcrypt -lcs50 -lm -o foo
foo.c:3:5: error: implicit declaration of function 'add' is invalid in C99
[-Werror,-Wimplicit-function-declaration]
    add(3, 5);
    ^
1 error generated.
make: *** [foo] Error 1
```

17. (2 punten) Leg uit wat er fout gaat, en hoe je dit kan oplossen.

Lastig lezen (3 punten)

Onderstaande oplossing voor mario's piramide is bijna niet door te komen!

```
mario.c
1) #include <stdio.h>
2) #include <cs50.h>
3)
4) int main(void)
5) {
6)     int x = 0;
7)     do
8)     {
9)         printf("Hoogte: ");
10)        x=get_int();
11)    } while (x <0 || x> 23);
12)
13)    for (int i = 0; i <x; i++)
14)    {
15)        for (int j = 0, n = x -i- 1; j < n; j++)
16)        {
17)            printf(" ");
18)        }
19)        for (int j = 0, n = i + 2; j < n; j++)
20)        {
21)            printf("#");
22)        }
23)        printf("\n");
24)    }
25)    return 0;
26) }
```

18. (1 punt) Het inspringen gaat volgens de styleguide op meerdere regels verkeerd. Geef alle regelnummers waar verkeerd is ingesprongen.

19. (2 punten) Geef nog twee andere verbeterpunten met betrekking tot stijl.

Kladpapier

Wij kijken niet naar wat je op dit vel schrijft!