# Oefententamen 2

C

**Naam:**

**Studentnummer:**

Tijd: 2 uur

Maximaal aantal punten: 30

## Menselijke compiler (10 punten)

0. (1 punt) Stel, je haalt het tentamen als je tenminste een 5.5 gemiddeld hebt gehaald voor beide deeltentamens of tenminste een 5.5 voor het hertentamen. Welke C-expressie defineert het tentamen halen?

   (a) `!(hertentamen < 5.5) || (deeltentamen2 && deeltentamen1 >= 5.5)`

   (b) `!(hertentamen <= 5.5) || (deeltentamen1 + deeltentamen2 >= 11.0)`

   (c) `deeltentamen1 + deeltentamen2 / 2.0 || hertentamen >= 5.5`

   (d) `(deeltentamen1 + deeltentamen2) / 2.0 >= 5.5 || hertentamen >= 5.5`

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

```
int x = 10;
float y = x * 2 / 3;
x = x % 2;
```

   (a) `x = 0, y = 6.66666`

   (b) `x = 8, y = 7.0`

   (c) `x = 1, y = 6.0`

   (d) `x = 0, y = 6.0`

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 (true)
{
    c = a;
}
else
{
    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 waarden voor de variabelen `a`, `b` en `c` zorgen ervoor dat na het uitvoeren van het stuk code de variabele `answer` dezelfde waarde heeft als variabele `c`? Alle variabelen zijn van het type integer.

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

(a) `a=1, b=1, c=0`

(b) `a=3, b=2, c=1`

(c) `a=2, b=3, c=2`

(d) `a=1, b=2, c=3`

4. (1 punt) Wat is de uitkomst ofwel het waarschijnlijke doel van het volgende stukje code?

```
int numbers[] = {4, 6, 7, 3, 1};
int n = 5;
for (int i = 1; i < n; i++)
{
    numbers[i] = numbers[i - 1] + numbers[i];
}
printf("%d\n", numbers[n - 1]);
```

(a) Om het laatste getal uit `numbers` uit te printen.

(b) Om de som van de getallen in `numbers` uit te printen.

(c) Om elk ander getal in `numbers` te verwisselen en de resulterende array uit te printen.

(d) Om de getallen in `numbers` cumulatief op te tellen en de resulterende array uit te printen.

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

```
printf("%c", 'c' - 'a' + 'A');
```

   (a) `"c - a + A"`
   (b) `2`
   (c) `c`
   (d) `C`
   (e) nothing, compile error

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

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

   (a) `xbbxxbbx`
   (b) `abxxxbxa`
   (c) `abxxabxa`
   (d) `abxxabxx`
   (e) nothing, compile error

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

```
#include <stdio.h>
int foo()
{
    int x = 3;
    printf("%d : ", x);
}

int main(void)
{
    foo();
    print("%d", x * 2);
}
```

   (a) `3 : x * 2`
   (b) `3 : 6`
   (c) `36`
   (d) nothing, compile error

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

```
... letter_a()
{
    return 'a';
}
```

   (a) `void`
   (b) `int`
   (c) `char`
   (d) zowel `int` als `char`
   (e) zowel `int` als `char` als `void`

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

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

}
int main(void)
{
    bar(3);
}
```

   (a) `3`
   (b) `321`
   (c) `123`
   (d) `0123`
   (e) `3210`
   (f) nothing, compile error

## Algoritmes (3 punten)

10. (1 punt) Wat is de runtime-complexiteit van insertion 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 binary 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; n = n / 2)
{
    for (int j = 0, m = strlen(name); j < m; j++) {
        printf("%c", name[j]);
    }
    printf("\n");
}
```

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

## Ontcijferen (6 punten)

In dit vak geven we vier deelcijfers voor de problem sets: scope, correctness, design en style. Uiteindelijk moet dit voor de universiteit cijfers tussen de 1 en 10 worden. Om dit te doen gebruiken we de volgende formule:

```
cijfer = (scope * (3 * correctness + 2 * design + style)) / 150 * 9 + 1
```

Natuurlijk berekenen we niet elk cijfer met de hand. In plaats daarvan maken we gebruik van een beschikbare hulpbron, ~~de computer~~ jou!

13. (2 punten) Implementeer de functie `cijfer` hieronder. Deze functie accepteert 4 deelcijfers en moet het bijbehorende cijfer op de nederlandse schaal van 1.0 t/m 10.0 `return`-en. Pas op! De functie `return`-ed een float, zorg dat je niet tussentijds afrond.

    ```
    float cijfer(int scope, int correctness, int design, int style)
    {
    ```

Gedurende het vak willen we graag weten wie er hoge cijfers halen voor de opdrachten. In plaats van dat wij door grote spreadsheets moeten speuren, implementeer jij de volgende functie :)

14. (4 punten) Implementeer de functie `hoge_cijfers()` hieronder. Deze functie accepteert als argumenten een array van `namen`, een 2D array van `deelcijfers` en een integer `n`. De functie moet alle namen uitprinten die als cijfer een 8 of hoger hebben gehaald op basis van de bijbehorende deelcijfers. De array `namen` is van lengte `n` en de array `deelcijfers` heeft de dimensies `n` x 4. Ter verduidelijking, `deelcijfers[3][2]` geeft van de vierde student het derde deelcijfer (design). De deelcijfers staan in de volgorde scope, correctness, design en style. De deelcijfers van de naam op index `x` in `namen` vind je op index `x` in `deelcijfers`. Je mag aannemen dat de functie `cijfer()` bestaat en werkt.

```
void hoge_cijfers(int n, string namen[], float deelcijfers[n][])
{
```

## Het werkt bijna, maar... (4 punten)

Hieronder staat code voor het omgedraaid uitprinten van een door de gebruiker ingevoerde string. Het werkt net niet helemaal.

reverse.c

```
1)  #include <cs50.h>
2)  #include <string.h>
3)  #include <stdio.h>
4)
5)  int main(void)
6)  {
7)      string s = get_string();
8)      for (int i = strlen(s) - 1; i > 0; i--)
9)      {
10)         printf("%c", s[i]);
11)     }
12)     printf("\n");
13) }
```

Bij het runnen van het programma gaan er een paar dingen mis. Dit kun je zien in de volgende test van de code:

```
~/workspace/ $ ./reverse
hello
olle
```

15. (2 punten) Spoor de fout in de code op. Meld het regelnummer, de fout, en hoe je deze fout zou oplossen.

Hieronder staat code voor het tellen van het aantal woorden in een door de gebruiker ingevoerde string. We nemen aan dat alle woorden gescheiden worden door een spatie. Je mag aannemen dat de gebruiker nooit te veel spaties invult. Het programma werkt net niet helemaal.

word_count.c

```
1)  #include <cs50.h>
2)  #include <string.h>
3)  #include <stdio.h>
4)
5)  int main(void)
6)  {
7)      string s = get_string();
8)      int count = 0;
9)      for (int i = 0, n = strlen(s); i < n; i++)
10)     {
11)         if (s[i] == ' ')
12)         {
13)             count++;
14)         }
16)     }
17)     printf("%d\n", count);
18) }
```

Bij het runnen van het programma gaan er een paar dingen mis. Dit kun je zien in de volgende tests van de code:

```
~/workspace/ $ ./word_count
hello world
1
~/workspace/ $ ./word_count
hello
0
~/workspace/ $ ./word_count

0
```

**Let op**! Bij de derde test heeft de gebruiker niks ingevuld, ook geen spatie.

16. (2 punten) Spoor de fout in de code op. Meld het regelnummer, de fout, en hoe je deze fout zou oplossen.

## Errors (4 punten)

Het volgende programma zou een vierkant van #jes moeten uitprinten, maar het programma compileert niet.

square.c

```
1)  int main(void)
2)  {
3)      int n = 5;
4)      for (int i = 0; i < n; i++)
5)      {
6)          for (int i = 0; i < n; i++)
7)          {
8)              printf("#");
9)          }
10)         printf("\n");
11)     }
12) }
```

De compiler geeft de volgende error bij het compileren:

```
~/workspace/ $ make square
clang square.c -Werror -lcrypt -lcs50 -lm -o square
square.c:6:18: error: declaration shadows a local variable [-Werror,-Wshadow]
        for (int i = 0; i < n; i++)
                 ^
1 error generated.
make: *** [square] Error 1
```

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

Het volgende programma zou alle getallen tot 10 op een nieuwe regel moeten printen, maar het programma compileert niet.

count.c

```
1)  #include <stdio.h>
2)  int main(void)
3)  {
4)      for (int i = 0, n = 10, i < n, i++)
5)      {
6)          printf("%d\n", i);
7)      }
8)  }
```

De compiler geeft de volgende errors bij het compileren:

```
~/workspace/ $ make count
clang count.c -Werror -lcrypt -lcs50 -lm -o count
count.c:4:29: error: declaration shadows a local variable [-Werror,-Wshadow]
    for (int i = 0, n = 10, i < n, i++)
                            ^
count.c:4:14: note: previous declaration is here
    for (int i = 0, n = 10, i < n, i++)
             ^
count.c:4:29: error: redefinition of 'i'
    for (int i = 0, n = 10, i < n, i++)
                            ^
count.c:4:14: note: previous definition is here
    for (int i = 0, n = 10, i < n, i++)
             ^
count.c:4:31: error: expected ';' in 'for' statement specifier
    for (int i = 0, n = 10, i < n, i++)
                              ^
count.c:4:31: error: expected expression
4 errors generated.
make: *** [count] Error 1
```

18. (2 punten) Leg uit wat er fout gaat, en hoe je dit kan oplossen.

## Lastig lezen (3 punten)

Onderstaande oplossing voor de caesar encryptie is bijna niet door te komen!

caesar.c

```
1)  #include <stdio.h>
2)  #include <ctype.h>
3)  #include <cs50.h>
4)  #include <string.h>
5)  #include <stdlib.h>
6)
7)  int main(int argc, string argv[])
8)  {
9)      if (argc != 2)
10)     {
11)         printf("Usage: ./caesar k\n");
12)         return 1;
13)     }
14)     int k = atoi(argv[1]);
15)     printf("plaintext: ");
16)     string p = get_string();
17)     printf("ciphertext: ");
18)     if (p != NULL) {
19)         for (int i = 0, n = strlen(p); i < n; i++)
20)         {
21)             char x = p[i];
22)             if (!isalpha(x))
23)             {
24)                 printf("%c", x);
25)             }
26)
27)
28)
29)             else {
30)                 if (isupper(x)) {
31)                     x = ((x - 'A') + k) % 26 + 'A';
32)                     printf("%c", x);
33)                 }
34)             else {
35)                 if (islower(x))
36)                 {
37)                     x = ((x - 'a') + k) % 26 + 'a';
38)                     printf("%c", x);
39)                 }
40)     }
41)     }
42)     }
43)     }
44)     printf("\n");
45)     return 0;
46) }
```

19. (1 punt) Het inspringen gaat volgens de styleguide op meerdere regels verkeerd. Geef alle regelnummers waar verkeerd is ingesprongen.

20. (2 punten) Geef nog twee andere verbeterpunten met betrekking tot stijl.

## Kladpapier

Wij kijken niet naar wat je op dit vel schrijft!