

Tentamen Programmeren in C (EE1400)

28 jan 2011, 9.00 – 12.00

- Zet op elk antwoordblad je naam en studienummer.
- Beantwoord alle vragen zo nauwkeurig mogelijk.
- Wanneer C code gevraagd wordt:
 - o Denk aan de declaraties en initialisaties.
 - o Gebruik een overzichtelijke indentering.
 - o Wellicht is het een goed idee om de code eerst uit te werken in klad en daarna het eindresultaat op het antwoordblad te zetten.
- Bij elke vraag staat aangegeven hoeveel punten maximaal met die vraag zijn te behalen. In totaal zijn maximaal 90 punten te behalen.

Tijdens het tentamen mag je het boek "A Book on C" (of een ander boek over C) erbij houden en ook de collegeslides en practicumhandleiding deel 1 en deel 2. Je mag NIET erbij houden: minitoetsen, practicumuitwerkingen en elektronische apparatuur (mobieltjes, rekenmachines, etc.).

Opgave 1 (24 punten)

Wat zal afgedrukt worden bij het uitvoeren van de volgende C code:

- a.

```
int a, b;
a = 1;
b = --a + 2;
printf ("%d %d\n", a, b);
```
- b.

```
int a = 5, b, c;
c = a++;
b = c + a++;
printf ("%d %d\n", a, b);
```
- c.

```
int a, b;
a = 10 / 4 + 6;
b = 10 % (2 + 2);
printf ("%d %d\n", a, b);
```
- d.

```
int a, b;
a = 0444;
b = 0xff;
printf ("%d %d\n", a, b);
```

- e.

```
int a = 2, b = 5;
a = a & b;
b = (~b) & (b << 2);
printf ("%d %d\n", a, b);
```
- f.

```
char i, j, k;
i = 'a' + 2;
j = 'p';
k = 't';
printf ("%c %d\n", i, k - j);
```

Opgave 2 (8 punten)

Wat zal afgedrukt worden bij het uitvoeren van de volgende C code:

- a.

```
int i, z;
z = 0;
for (i = 0; i < 6; i++) {
    z = z + 2;
}
printf ("%d\n", z);
```
- b.

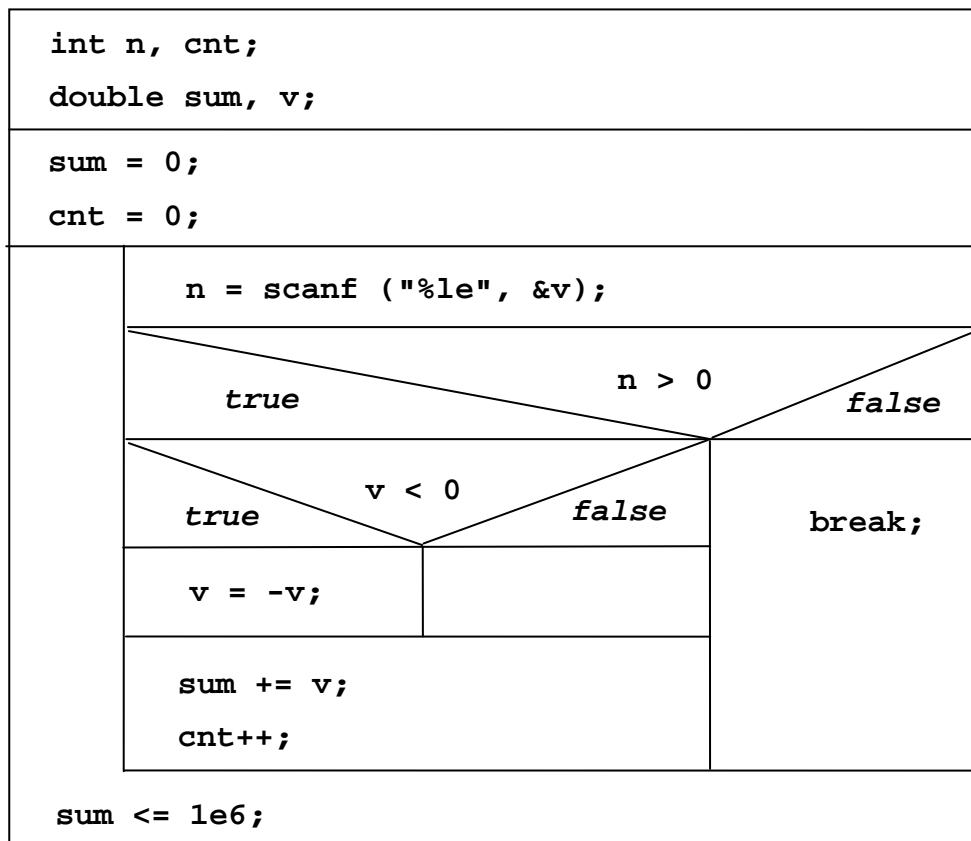
```
int i, z;
z = 0;
for (i = 0; i < 6; i++) {
    if (i == 3) continue;
    z = z + 2;
}
printf ("%d\n", z);
```
- c.

```
int i, z;
z = 0;
for (i = 0; i < 6; i++) {
    if (z > 2) break;
    z = z + 2;
}
printf ("%d\n", z);
```
- d.

```
int i, z;
i = z = 0;
while (i < 6 || z < 9) {
    z = z + i;
    i++;
}
printf ("%d\n", z);
```

Opgave 3 (10 punten)

Gegeven het volgende Nassi-Schneiderman diagram. Geef de bijbehorende C code.



Opgave 4 (8 punten)

Gegeven de opzet van de volgende functie:

```
char ** find_words (char line[])
{
    static char * words[MAXWORDS + 1];

    .....

    return words;
}
```

De array `line[]` bevat een regel van woorden. De woorden bestaan uit de karakters a-z en A-Z en worden gescheiden door één of meerdere spaties. Het eerste karakter van de regel is ook altijd een spatie. De regel wordt afgesloten met het karakter '`\n`'.

Maak de functie `find_words()` af zodanig dat hij een array teruggeeft met pointers naar het begin van de woorden die voorkomen in `line[]` (d.w.z. pointers naar elk eerste karakter van elk woord). De pointer waarden in het array moeten worden afgesloten met een waarde NULL. Vergeet niet de gebruikte variabelen te declareren.

Opgave 5 (10 punten)

Gegeven een polynoomfunctie

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_3 x^3 + a_2 x^2 + a_1 x + a_0$$

De waarden a_n zijn gegeven in de array **a[]** zodanig dat a_n correspondeert met **a[n]** en a_0 met **a[0]**. Schrijf een C functie welke $f(x)$ uitrekenet zonder gebruik te maken van standaard bibliotheekfuncties (Wanneer je gebruik maakt van bibliotheekfuncties is het maximaal aantal te behalen punten 6). Gebruik als functie header:

```
double polynomial (int n, double a[], double x)
```

Opgave 6 (8 punten)

Gegeven de volgende recursieve functie:

```
int mdiv2 (int a)
{
    int v;

    if (a <= 1)
        v = 0;
    else
        v = 1 + mdiv2 (a / 2);

    return v;
}
```

Schrijf een iteratieve versie van deze functie. Hint: analyseer het gedrag en schrijf hiervoor C code waarin gebruik wordt gemaakt van een lus.

Opgave 7 (7 punten)

Er wordt een binaire boom geconstrueerd om de namen van verschillende vruchten alfabetisch te sorteren. Teken het resultaat wanneer achtereenvolgens de volgende namen worden ingelezen.

banaan, dadel, appel, vijg, pruim

Aangezien banaan de eerste naam is die wordt ingelezen, zal deze in de root van de boom komen. Geef duidelijk aan of verbindende lijnen schuin naar links of naar rechts lopen!

Opgave 8 (15 punten)

Om de concurrentie voor te blijven ontwikkelt TNT Post een nieuwe sorteermachine. De software voor de machine wordt in C geschreven en daarbij wordt de volgende structure gebruikt om een brief te modelleren:

```
struct brief {
    int hoogte;
    int breedte;
    float gewicht;
    char postcode[7];
    struct brief * next;
};
```

De members hoogte en breedte worden gebruikt om de hoogte en breedte (in mm) van de brief te specificeren. Member gewicht geeft het gewicht van de brief in grammen en member postcode geeft de postcode als een string van 4 cijfers en 2 letters (alleen hoofdletters worden gebruikt). Het member next wordt gebruikt om een lineaire lijst van structures van het type brief te maken (laatste uit de lijst heeft een 'next' met waarde NULL). Zo'n lijst wordt gebruikt om een groot aantal brieven te representeren.

- a. Schrijf een functie waarmee gecontroleerd wordt of de afmetingen van één brief (hoogte bij breedte of breedte bij hoogte) niet groter zijn dan 250 mm bij 400 mm. Wanneer de brief te groot is moet de waarde 0 worden teruggegeven, anders een waarde ongelijk 0. Gebruik als functie header:

```
int brief_OK (struct brief * b)
```

- b. Schrijf een functie om het totale gewicht in kg (!) van alle brieven uit de lijst te berekenen. Gebruik als functie header:

```
double totaal_gewicht (struct brief * brieven_lijst)
```

- c. Neem aan dat de lijst (alfanumeriek) gesorteerd is op postcode (2628CD staat voor 2628CF, en 2628CD voor 2629CD). Schrijf een functie waarmee een nieuwe brief structure aan de lijst toegevoegd kan worden zodanig dat de lijst gesorteerd blijft. Je mag aannemen dat de nieuwe brief die wordt toegevoegd nooit de eerste brief uit de lijst zal worden. Verder mag je gebruik maken van de standaard bibliotheekfunctie `strcmp()`. Gebruik als functie header:

```
void brief_erbij (struct brief * brieven_lijst,
                 struct brief * b)
```

***** EINDE *****