

## Tentamen Programmeren in C (EE1400)

3 feb. 2012, 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 Functies uit de standard C library mogen gebruikt worden, tenzij anders vermeld.
  - o Tenzij er expliciet om gevraagd wordt, hoeven er geen header files van de standard C library (bijv. stdio.h) te worden opgenomen.
  - 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), de collegeslides en de practicumhandleiding erbij houden. NIET toegestaan zijn oude toetsen en tentamens, en practicumuitwerkingen. Een eenvoudige, niet programmeerbare, rekenmachine is toegestaan: TI-30 en Casio FX-82, maar geen andere elektronische apparatuur.

### Opgave 1 (24 punten)

Wat zal afgedrukt worden bij het uitvoeren van de volgende C code (let goed op de format specifiers):

- a. 

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

```
int a, b, c = 10;
a = --c;
b = --c + --a;
printf ("%d %d\n", a, b);
```
- c. 

```
long a, b;
a = 4.3 * 3;
b = 8 % 3 + 1;
printf ("%ld %ld\n", a, b);
```
- d. 

```
int a, b;
a = 0x22;
b = a + 16;
printf ("%d %x\n", a, b);
```

- e. `int a = 21, b = 19;`  
`a = a & b;`  
`b = (a | b) << 2;`  
`printf ("%d %d\n", a, b);`
- f. `char a, b;`  
`a = 3 + 'k';`  
`b = (a == 'k') || !(a == '3');`  
`printf ("%c %d\n", a, b);`

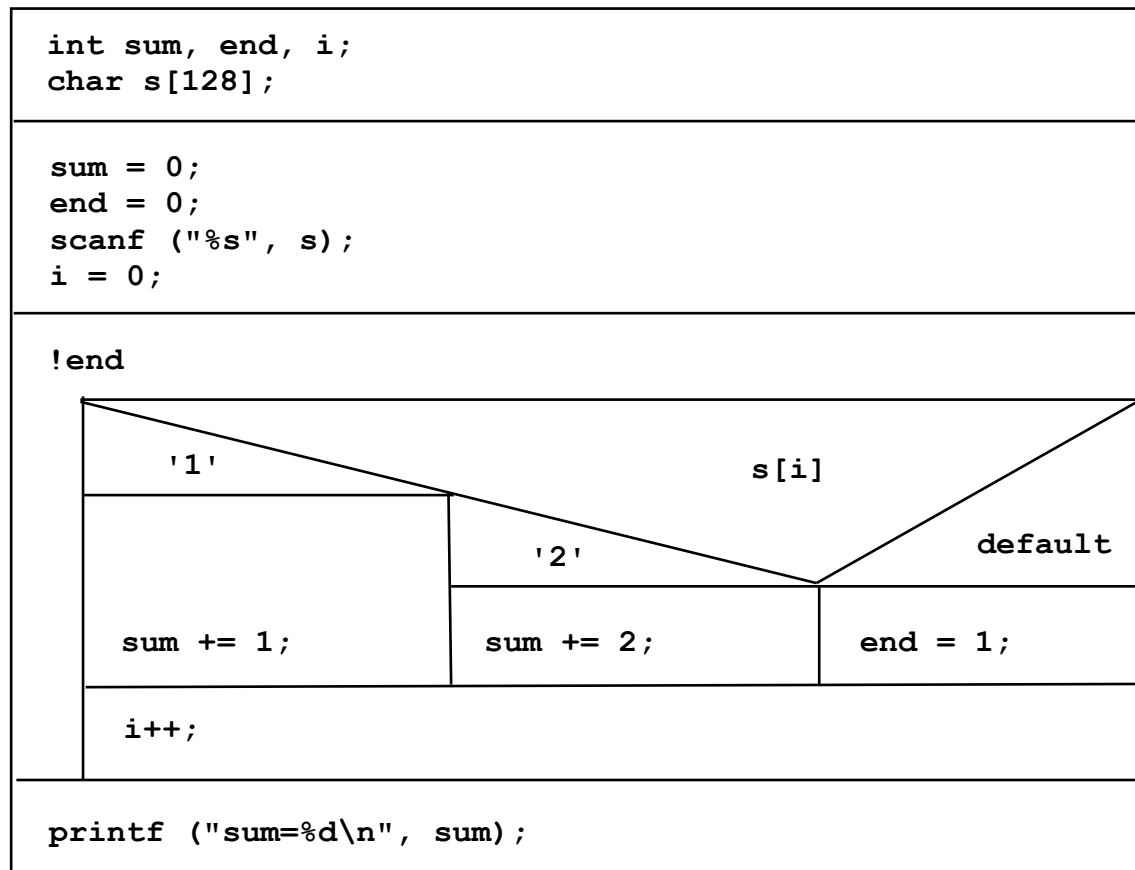
## Opgave 2 (8 punten)

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

- a. `int i, z;`  
`z = 1;`  
`for (i = 1; i < 6; i++) {`  
`z = z - 2;`  
`}`  
`printf ("%d\n", z);`
- b. `int i, z;`  
`z = 1;`  
`for (i = 1; i < 6; i++) {`  
`if (i != 4) {`  
`z = z + 1;`  
`}`  
`else`  
`continue;`  
`z = z + 2;`  
`}`  
`printf ("%d\n", z);`
- c. `int i, z;`  
`z = 1;`  
`for (i = 1; i < 6; i++) {`  
`z = z * 3;`  
`if (z > 30) break;`  
`}`  
`printf ("%d\n", z);`
- d. `int i, z;`  
`i = z = 1;`  
`do {`  
`z += 5;`  
`} while (i++ <= 11 && z < 100);`  
`printf ("%d\n", z);`

### Opgave 3 (10 punten)

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



### Opgave 4 (6 punten)

Gegeven de volgende recursieve functie:

```
int func (int x)
{
    int a;

    if (--x < 1)
        a = 4;
    else
        a = 2 * func (x);

    return a;
}
```

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

### Opgave 5 (8 punten)

Schrijf een functie:

```
int str2int (char *s)
```

om een string zoals "6723" te converteren naar de corresponderende integer waarde. Je kunt daarbij gebruik maken van het feit dat de integer waarde van een karakter  $c$  ('0', '1', '2', ... '9') kan worden gevonden m.b.v. de expressie  $c - '0'$ . Er kan verder van worden uitgegaan dat de string alleen bestaat uit cijfers en er hoeft niet gecheckt te worden op een overflow.

### Opgave 6 (8 punten)

De populatievariantie  $\sigma^2$  van een populatie  $x_1, x_2, x_3, \dots, x_n$  met gemiddelde  $u$  is gelijk aan

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - u)^2$$

Schrijf nu een functie:

```
double variance (double x[], int n, double u)
```

om een populatievariantie  $\sigma^2$  te berekenen, waarbij de array  $x[1], x[2], x[3], \dots, x[n]$  de populatie representeert, en  $n$  en  $u$  respectievelijk het aantal en het gemiddelde.

### Opgave 7 (8 punten)

De overheid heeft een programma geschreven waarmee gegevens over burgers worden bijgehouden. Het programma maakt gebruik van een linear linked list waarin elke burger wordt gerepresenteerd met een structure (voor het laatste element uit de lijst geldt dat de pointer 'volgend' gelijk is aan NULL):

```
struct burger {
    char *voornaam;
    char *achternaam;
    int sofi_nummer;
    struct burger *volgend;
};
```

Men wil graag weten hoeveel personen er zijn met een bepaalde voornaam. Aan u de taak om een functie te schrijven welke als argumenten mee krijgt het begin van de lijst met alle burgers, en de voornaam welke geteld moet worden. De functie dient dan als return waarde te geven het aantal burgers met de gespecificeerde voornaam. Gebruik de volgende functieheader:

```
int tel_namen (struct burger *lijst, char *naam)
```

### Opgave 8 (8 punten)

Gegeven een matrix (2 dimensionaal array)  $a[M][N]$ , waarbij element  $a[i][j]$  het aantal exemplaren representeert dat een klant  $i$  van product  $j$  gekocht heeft ( $i = 0 \dots M-1, j = 0 \dots N-1$ ). Gevraagd wordt om een functie te schrijven welke het aantal verkochte exemplaren voor elk product berekent, en dit op te slaan in één dimensionale array  $b[N]$ , waarbij  $b[j]$  ( $j = 0 \dots N-1$ ) het totaal aantal exemplaren voor product  $j$  aangeeft. De functie dient bovendien het totaal van alle aantallen  $b[j]$  te retourneren. Gebruik de volgende functie header:

```
int calculate_totals(int a[M][N], int b[N])
```

waarbij  $M$  en  $N$  als constanten zijn gedefinieerd met een `#define`. Ga ervan uit dat de array  $b$  bij het begin van de aanroep een willekeurige inhoud kan hebben.

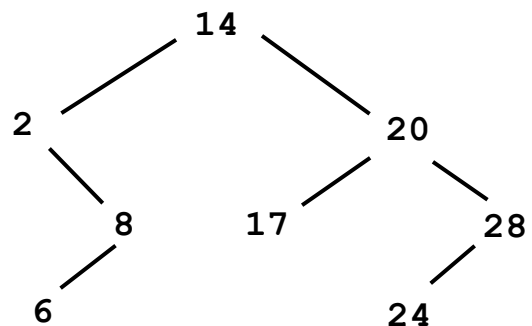
### Opgave 9 (10 punten)

Gegeven een gesorteerde binary tree datastructuur waarin de nodes van de tree worden gerepresenteerd met de volgende structure:

```
struct node {
    int data;
    struct node *left;
    struct node *right;
};
```

Member `left` verwijst naar de top van de subtree waarin alle nodes zitten met een kleinere waarde dan de waarde gegeven door 'data', en member `right` verwijst naar de top van de subtree waarin alle nodes zitten met een grotere waarde. Wanneer er geen subtree is links of rechts, dan heeft het desbetreffende member de waarde `NULL`.

- Gegeven de nevenstaande gesorteerde binary tree. Neem de tree over op je antwoordblad en geef duidelijk aan (gaat de verbindende lijn schuin naar links of schuin naar rechts?) waar nieuwe nodes met de getallen 3 en 19 geplaatst moeten worden
- Schrijf een functie waarmee de som van alle waarden in de tree wordt bepaald. Gebruik als functieheader:



```
int tree_sum (struct node * n)
```

waarbij  $n$  de root van de tree is (hint: gebruik recursie).

\*\*\*\*\* EINDE \*\*\*\*\*