

# Tentamen Programmeren in JavaScript

## Vrijdag 2 november 2001, 9.30–12.30 uur

### Universiteit Leiden — Informatica

Geef overal duidelijke uitleg. Veel succes!

#### 1. (2 punten)

- a. Geef een belangrijk verschil tussen Java en JavaScript.
- b. Wat wordt er in de alert-boxen afgedrukt door het volgende stukje JavaScript:

```
var x = 3; var y = 9;
alert ("Som: " + x + y + ".");
alert ("Som: " + parseInt(x) + parseInt(y) + ".");
alert ("Som: " + parseInt(x + y) + ".");
```

Leg uit waarom de functie `parseInt` vaak gebruikt wordt.

- c. Schrijf een JavaScript-functie `roter ( )` die twee even grote plaatjes (`een.jpg` en `twee.jpg`) afwisselend laat zien, elk gedurende 5 seconden.

#### 2. (2 punten)

We hebben een array `A` met `A.length` verschillende positieve gehele getallen.

- a. Geef een JavaScript-functie `kl (A)` die de *waarde* van het *kleinste* array-element van het array `A` retourneert.

- b. Geef een JavaScript-functie `eennakl (A)` die de *waarde* van het *op één na kleinste* array-element van het array `A` retourneert.

- c. Leg in *woorden* uit hoe met behulp van de twee functies van **a** en **b** een eenvoudige sorteermethode gemaakt kan worden. Geef in woorden aan welke aanpassingen eventueel nog aan deze functies moeten plaatsvinden. Doe dit voor de volgende situatie: er moet een aantal personen op volgorde van lengte gezet worden, en de functie-aanroep `kl (A)` levert de lengte van de kleinste persoon op, en de functie-aanroep `eennakl (A)` de lengte van de op één na kleinste persoon.

Als randvoorwaarde geldt dat de functies `kl` en `eennakl` beide hooguit  $n/2$  (zonodig naar beneden afgerond) keer gebruikt mogen worden.

#### 3. (2 punten)

- a. Bij een functie kun je te maken hebben met *locale* en *globale* variabelen. Verder heb je ook nog variabelen (parameters) die tussen haakjes worden meegegeven. Leg deze begrippen aan de hand van een *klein* voorbeeld duidelijk uit.

- b. Een zeker JavaScript-programma bevat de volgende functies:

```
function telop1 ( ) { var temp = 2*a + b; return temp; }
function telop2 (a,b) { var temp = 2*a + b; return temp; }
function telop3 ( ) { a = 2*a + b; return a; }
function telop4 (a,b) { a = 2*a + b; return a; }
```

Wat gebeurt er bij het volgende stukje JavaScript? NB `a` en `b` zijn globale variabelen.

```
a = 5; b = 3; alert (telop1 ( )); alert (a); alert (b);
a = 5; b = 3; alert (telop2 (a,b)); alert (a); alert (b);
```

```

a = 5; b = 3; alert (telop2 (b,a)); alert (a); alert (b);
a = 5; b = 3; alert (telop3 ( )); alert (a); alert (b);
a = 5; b = 3; alert (telop4 (a,b)); alert (a); alert (b);
a = 5; b = 3; alert (telop4 (b,a)); alert (a); alert (b);

```

Wat wordt er in de alert-boxen afgedrukt? Geef hierbij uiteraard uitleg.

**4. (2 punten)** Deze opgave gaat over een  $m$  bij  $n$  array  $A$  met gehele getallen tussen 0 en 9. Een voorbeeld, met  $m = 3$  rijen en  $n = 5$  kolommen:

```

1 2 3 4 5
0 0 8 3 5
3 0 3 9 1

```

**a.** Schrijf een JavaScript-functie `willekeur (n)` die een willekeurig (random) geheel getal tussen 1 en  $n$ , grenzen inbegrepen, oplevert. Gebruik `Math.floor (r)` (rond  $r$  naar beneden af) en `Math.random ( )` (geeft random getal groter dan of gelijk aan 0, en kleiner dan 1).

**b.** Schrijf een stukje JavaScript dat een  $m$  bij  $n$  array  $A$  maakt, en de array-elementen met willekeurige getallen tussen 0 (let op: 0!) en 9 vult. Gebruik de functie van **a**.

**c.** Schrijf een JavaScript-functie `hoeveel (A,i)` die de “gewone” getalswaarde oplevert van de cijfers uit de  $i$ -de rij van het  $m$  bij  $n$  array  $A$ . Met  $i=0$  zou het antwoord voor het voorbeeldarray 12345 zijn, met  $i=1$  835, en met  $i=2$  30391.

Opmerking: lukt het niet met algemene  $n$ , neem dan  $n = 5$ .

**5. (2 punten)**

Deze opgave gaat over onderstaande functie  $F$ ; de functie krijgt als argument een array met getallen.

```

function F (A) {
  var B = new Array ( );
  var j = 0; var i;
  for (i = 0; i < A.length; i++) {
    if (A[i] != 0) {
      B[j] = A[i];
      j++;
    }// if
  }// for
  return B;
}// F

```

**a.** Leg uit wat de functie  $F$  doet (wat zit er in het array  $B$  op het moment dat `return B`; wordt uitgevoerd?). Je mag dit uitleggen aan de hand van een voorbeeldarray dat als argument wordt meegegeven (bijvoorbeeld een array met array-elementen 1, 7, 0, 2, 5, 0, 6).

**b.** Wat had de functie gedaan als er `B[i]` had gestaan in plaats van `B[j]` (je mag weer een zelfde voorbeeldarray gebruiken in je uitleg)?

**c.** We vervangen de test `(A[i] != 0)` door `(i < A.length-1 && A[i] < A[i+1])`. Beantwoord vraag **a** voor deze nieuwe situatie, en leg hierbij uit waarom er `i < A.length-1` in de test staat.

**d.** Mogen de twee tests `i < A.length-1` en `A[i] < A[i+1]` in **c** omgewisseld worden?