

# Algoritmen en datastructuren

test 1: controlestructuren (eenvoudig en genest)



Naam:

datum:

reeks:

... / 2

## 1. Code begrijpen (15 à 20 min)

- Wat doet volgend stukje code. Omschrijf de functionaliteit dus WAT de code doet, niet hoe de functie dit doet!
- Geef een gepaste boodschap op de plaats van de puntjes.
- Test aan de hand van volgend voorbeeld

getal1 = 456789

getal2 = 987321

```
var getal1 = parseInt(prompt("geef een getal aub"));
var getal2 = parseInt(prompt("geef een getal aub"));
var uitGetal= 0;
var m = 1;

while (getal1 !=0 && getal2 != 0)
{
    if(getal1%10 < getal2%10)
    {
        uitGetal= uitGetal + getal1%10 * m;
    }
    else
    {
        uitGetal= uitGetal + getal2%10 * m;
    }
    m*= 10;
    getal1 = (getal1 - getal1%10) / 10;
    getal2 = (getal2 - getal2%10) / 10;
}
confirm(".... = " + uitGetal);
```

# Algoritmen en datastructuren

test 1: controlestructuren (eenvoudig en genest)



Naam:

datum:

reeks:

... / 2

## 2. Code aanvullen (15 à 20 min)

- Vervolledig volgend stukje code waarin het product van het grootste en het kleinste even getal wordt berekend:  
(**TIP:** als je het niet meteen ziet, maak dan een analyse en schrijf je eigen versie van het algoritme... probeer dit dan in te passen in onderstaande code!)

```
// programma voor het berekenen van het product van
// het kleinste en het grootste even cijfer in een getal
var getal = parseInt(prompt("geef een getal aub"));
var produkt = 0;
var kleinste = .....;
var grootste = .....;
while (.....)
{
    if(.....)
    {
        if(..... < kleinste)
        {
            kleinste = .....;
        }
        if(..... > grootste)
        {
            grootste = .....;
        }
    }
    getal = .....;
}
produkt = kleinste * grootste;
confirm("het product van het kleinste en het grootste even cijfer = " +
produkt);
```

## Algoritmen en datastructuren

test 1: controlestructuren (eenvoudig en genest)



Naam:

datum:

reeks:

... / 2

### 3. Code schrijven (50 à 60 min)

- Schrijf een stukje JavaScript code voor volgende probleemstelling:
- Begin met een analyse
  - input (wat heb je nodig, variaties?)
  - output

```
// Na de examens worden de punten van een student gewogen en
// samengeteld om de student een rapport te kunnen geven met een
// totaal aantal punten
// Alle examens staan op 20 punten, de weging gebeurt op basis van
// het aantal credits voor dat vak.
// Zorg ervoor dat de student op basis van punten en credits een correct
// eindpercentage (op 100) krijgt op zijn rapport.

// Schrijf een algoritme dat de gebruiker om de juiste input vraagt
// (punten, credits, ...) en dat het gemiddelde van alle punten als output
// terug geeft.
```