

1 4 9 9 9

3 2 0 6 1

2 7 6

## “14999 32061 276?”

CPU's worden sneller, geheugenchips groter en programmeertalen krachtiger. Bij gelijkblijvende prijs! Moore voorspelde al de regelmaat hiervan ... 40 jaar geleden. Je zou haast vergeten dat snelle, efficiënte algoritmes en compacte en slimme coderingstechnieken nog altijd cruciaal zijn in moderne ICT.

In deze challenge word JIJ uitgedaagd ... kun je het docententeam van HBO-ICT NSE verslaan door hun score te verbeteren? Duik jij onder deze getallen?

## “Bring it on... wat moet ik doen?”

Centraal in deze coding challenge staat deze puzzel:

VVD + D66 + CDA + CU = REG

Verskillende letters zijn verschillende decimale cijfers. De puzzel kent twee varianten: In variant A moet je de cijfers 1 t/m 8 gebruiken. In variant B de cijfers 1 t/m 9, maar niet de 6.

In deze challenge moet je programma deze puzzel zo snel mogelijk oplossen (doorlooptijd) en daarnaast zo compact mogelijk zijn. Nu zijn er alleen al qua hardware honderd configuraties denkbaar, laat staan het aantal programmeertalen. Om appels met appels te kunnen vergelijken zijn er een aantal spelregels:

- Het programma moet geschreven zijn in Atmega32 Assembly of C
- De doorlooptijd wordt gemeten a.d.h.v. het aantal gebruikte cycles in de Atmega simulator; De doorlooptijd van A en B samen is bepalend.
- De omvang van de code wordt bepaald aan de hand van de build output van je oplossing.

De snelste docentenoplossing is 14999(A) + 32061(B) cycles, gebruik makend van hetzelfde algoritme en dus dezelfde code. Op een Atmega32-16 Mhz betekent het een doorlooptijd van 2,9 ms (omgerekend naar een 3Ghz machine: 0,016 ms) De omvang van de code is 276 bytes. Versla jij het docententeam door minder cycles of minder bytes te gebruiken?

Een aantal randvoorwaarden aan de opdracht:

- Je programma moet generiek genoeg zijn **om beide varianten van de puzzel op dezelfde wijze** op te lossen (uiteraard mag je de definitie van de toegestane cijfers voor het runnen aanpassen. Maar ook niet meer dan dat. Een grote *if* of *rjmp*, direct aan het begin, wordt beschouwd als niet-generiek)
- Je mag **geen** gebruik maken van voorkennis over de oplossing van de puzzel. Er zijn vier oplossingen voor variant A en 16 voor variant B. Je programma mag dus bijv. niet afbreken sec en alleen omdat het vier oplossingen gevonden heeft. Kortom: het programma moet de gehele “oplossingsruimte” doorzoeken.
- Uiteraard mag je op voorhand wel wiskunderegels en -logica toepassen. Via wat logisch redeneren kun je bijv. afleiden dat de V niet een 8 kan zijn. Alles wat op die manier **afleidbaar en geldig is voor beide varianten** mag je gebruiken.
- Het programma moet de oplossingen in het geheugen plaatsen zodat deze controleerbaar zijn.

## Prijzen

Zoals je mag verwachten zijn aan deze challenge prijzen verbonden. Een appeltaart voor de snelste, werkende, studentenoplossing. Een appeltaart voor de meest compacte, werkende, studentoplossing. Mocht je oplossing sneller of compacter zijn dan die van het docententeam dan krijg je een bonustaart en... misschien wel het belangrijkste: eeuwige roem dat je het NSE-docententeam hebt verslagen! Dus in totaal liggen er vier appeltaarten én eeuwige roem in het verschiet!

## Logistiek geneuzel

Om deze challenge soepel te laten verlopen nog wat logistieke zaken:

- De challenge staat open tot 7/11/2018, 23:59 CET
- Uiteraard lever je alleen eigen werk in. Meedoen in teams mag: je (team) moet de code kunnen uitleggen en demo-en
- Alleen studenten van de opleiding HBO-ICT van de Hanzehogeschool kunnen meedoen
- Je code (.asm en/of .c) met commentaar dien je te uploaden via het inleverpunt op de BB-course van Thema2.1 NSE/SE. Direct aan het begin van je code dien je ook kort aan te geven welke wiskunde/logica regels je hebt gehanteerd en waarom dat versnellend werkt.

## Oh ja...

<preachmode>Eeuwige roem is natuurlijk leuk... en appeltaart ook. Je studie is echter 'n stuk belangrijker. </preachmode>