

## Algemene opmerkingen (lees dit aandachtig voor je begint)

1. Vul je voornaam, naam en school in, **alleen op het eerste blad**. Op alle andere bladzijden mag je enkel schrijven in de daarvoor **voorzijene kaders**. Als je door een fout toch buiten de antwoordkaders moet schrijven, moet je op hetzelfde blad papier verderschrijven. Anders kunnen we je antwoord niet verbeteren.
2. Je mag alleen iets om te schrijven bij je hebben. Rekentoestel, GSM, ... zijn **verboden**.
3. Schrijf je antwoorden met blauwe of zwarte **pen of balpen**. Laat geen antwoorden staan in potlood. Als je kladbladen wilt, vraag ze dan aan een toezichthouder.
4. Je **moet** op de vragen antwoorden in **pseudo-code** of in één van de **toegestane programmeertalen** (Java, C, C++, Pascal, Python, PHP). We trekken geen punten af voor syntaxfouten. Je mag geen voorgedefinieerde functies gebruiken, behalve diegenen die op de volgende bladzijde staan.
5. Je mag **op geen enkel moment met iemand communiceren**, behalve met de organisatoren of toezichthouders. Alle technische en inhoudelijke vragen mag je enkel stellen aan de organisatoren. Logistieke vragen mogen gesteld worden aan de toezichthouders.
6. Je mag je **plaats niet verlaten** tijdens de proef, bijvoorbeeld om naar het toilet te gaan of te roken. Afhankelijk van de plaats waar je de proef aflegt, kan het ook verboden zijn om te eten of te drinken in het lokaal.
7. Je krijgt **exact 1 uur en 40 minuten** de tijd om op de vragen te antwoorden. Een **overzicht** over pseudo-code en de afspraken en conventies die we hanteren staat op de volgende bladzijde.

**Succes !**

## Overzicht pseudo-code

In de vragenlijst kom je opgaves tegen waarbij je een stuk code moet aanvullen. We vragen je om dat te doen met ofwel een *expressie*, ofwel een *instructie*. Expressies berekenen een waarde, instructies voeren een actie uit.

Expressies bezitten een waarde en worden gebruikt in berekeningen, om de waarde van een variabele te veranderen, en in condities. Sommige expressies hebben een numerieke waarde (een geheel of reëel getal), anderen een booleaanse waarde (**true** of **false**). Instructies stellen acties voor die uitgevoerd moeten worden. Een algoritme is een opeenvolging van instructies.

De meest eenvoudige expressies zijn gehele getallen (zoals 3, -12...) en variabelen (zoals  $x$ ,  $sum$ ...). We kunnen een expressie ook samenstellen met operatoren zoals de volgende tabel toont.

operatie	notatie	voorbeelden	
gelijkheid	=	$3 = 2$ is <b>false</b>	$1 = 1$ is <b>true</b>
verschil	$\neq$	$1 \neq 4$ is <b>true</b>	$3 \neq 3$ is <b>false</b>
vergelijking	$>, \geq, <, \leq$	$1 > 3$ is <b>false</b>	$5 \leq 5$ is <b>true</b>
optellen en aftrekken	$+, -$	$3 + 2$ is 5	$1 - 9$ is -8
vermenigvuldigen	$\times$	$2 \times 7$ is 14	$-2 \times 3$ is -6
gehele deling	/	$10/3$ is 3	$9/3$ is 3
rest van de gehele deling	%	$10\%3$ is 1	$9\%3$ is 0
logische "niet"	<b>not</b>	<b>not</b> $3 = 2$ is <b>true</b>	<b>not</b> $1 = 1$ is <b>false</b>
logische "en"	<b>and</b>	$3 = 3$ <b>and</b> $5 \leq 9$ is <b>true</b>	$1 \geq 3$ <b>and</b> $2 = 2$ is <b>false</b>
logische "of"	<b>or</b>	$3 = 3$ <b>or</b> $12 \leq 9$ is <b>true</b>	$1 \geq 3$ <b>or</b> $2 \neq 2$ is <b>false</b>
toegang tot het element van een array	[ ]	gegeven $arr$ is $[1, 2, 3]$ , dan $arr[1]$ is 2	
functie oproepen die een waarde teruggeeft		$size(list)$	$min(12, 5)$

De functies die jullie mogen gebruiken zijn, naast diegenen die in de opgave worden gedefinieerd en diegenen in de tabel hieronder, ook de functies  $\max(a, b)$ ,  $\min(a, b)$  en  $\text{pow}(a, b)$ . Zij berekenen respectievelijk het grootste van twee getallen, het kleinste van twee getallen, of de machtsverheffing ( $a^b$ ).

De meest eenvoudige instructies staan in deze tabel samengevat :

operatie	notatie	voorbeeld
toekenning	$\leftarrow$	$x \leftarrow 20 + 11 \times 2$
teruggave	<b>return</b>	<b>return</b> 42
functie oproepen (die niets teruggeeft)	naam van de functie	$sort(list)$

Daarnaast hebben we ook *controle-instructies* waarvan we er 3 gebruiken : **if-else**, **while** et **for**. Deze instructies zullen een groep andere instructies uitvoeren, afhankelijk van een conditie. De notatie **for** ( $i \leftarrow a$  **to**  $b$  **step**  $k$ ) { [...] } staat voor een lus die herhaald wordt zolang  $i \leq b$ , waarbij  $i$  begint vanaf  $a$  en telkens verhoogd wordt met  $k$  aan het eind van elke herhaling.

We kunnen variabelen gebruiken om een waarde in op te slaan, en *arrays* om meerdere waarden op te slaan. Een array  $arr$  van grootte  $n$  wordt geïndexeerd van 0 tot  $n - 1$ . De notatie  $arr[i]$  geeft toegang tot het  $(i + 1)$ -ste element van de array. Zo is het eerste element van die array  $arr[0]$ . We kunnen een nieuwe array  $arr$  aanmaken van grootte  $n$ , waarvan alle elementen geïnitieerd zijn op 0, met de notatie  $arr \leftarrow \text{newArray}(n)$ . We kunnen array  $arr$  van grootte  $n$  kopiëren naar een nieuwe array  $newArr$  kopiëren met de notatie  $newArr \leftarrow arr$ .

## Over de kost van operaties

Bij sommige vragen moet je een zo efficiënt mogelijk algoritme schrijven om de maximumscore te kunnen halen.

Als we de tijdscomplexiteit van een programma willen kennen (een schatting van de tijd die het zal kosten om uit te voeren), is één van de mogelijke manieren het tellen van het aantal elementaire operaties die het algoritme uitvoert. Alle basisinstructies laten we meetellen als één tijdseenheid. Het aanmaken van een array van grootte  $n$ , of het kopiëren van zo'n array, kost  $n$  tijdseenheden. Voor de **if-else**, **while** en **for** instructies, tellen we één tijdseenheid voor het uitrekenen van de conditie. Daarna tellen we alle instructies die worden uitgevoerd.