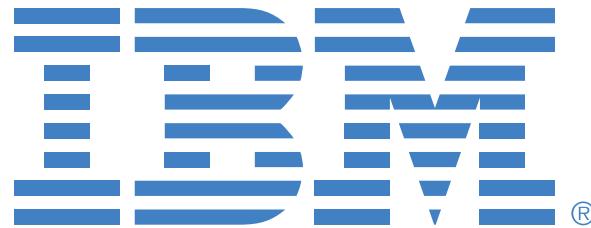


# 2<sup>de</sup> Olympiades in de Informatica

Sponsor :



Met de steun van :



2011

<http://www.be-oi.be>



# DE BELGISCHE OLYMPIADES IN DE INFORMATICA

De **Belgische Olympiades in de Informatica** (OI) zijn competities in programmeren, algoritmeek en logica, georganiseerd door de Belgische universiteiten en hogescholen waar opleidingen informatica gedoceerd worden. Er zijn twee wedstrijden: de eerste is voor leerlingen uit het middelbaar onderwijs. De beste kandidaten plaatsen zich voor de Internationale Olympiades in Informatica (IOI). De tweede wedstrijd focust op studenten uit het eerste jaar van het hoger onderwijs.

De Olympiades verlopen in twee rondes:

- **Halve Finales** in verschillende regionale centra;
- en **Finales** in Louvain-La-Neuve.

Alle informatie over de wedstrijd staat op de officiële website : <http://www.be-oi.be>.

## INTERNATIONALE OLYMPIADE IN DE INFORMATICA

De **Internationale Olympiade in de Informatica** (IOI) is een internationale informaticawedstrijd voor jongeren van het secundair onderwijs. Elk jaar vindt de wedstrijd plaats in een ander land. Vorig jaar was Canada het gastland, dit jaar is het de beurt aan Thailand. Meer dan 80 landen zijn vertegenwoordigd en sturen een nationale ploeg. De IOI is één van de wetenschappelijke olympiades die internationaal erkend zijn, naast de olympiades voor de disciplines wiskunde, biologie, chemie, fysica, aardrijkskunde, ...

België nam opnieuw deel aan de Internationale Olympiades in 2010, voor de eerste keer sinds de enige vorige deelname ooit in 1992. *De Belgische Olympiades in de Informatica* hebben onder andere als doel leerlingen uit het secundair onderwijs te selecteren voor het vormen van een Belgische ploeg voor de IOI 2011.

U kunt meer informatie over de IOI vinden op hun officiële website: <http://www.ioinformatics.org/> en over de editie 2011 op <http://www.ioi2011.or.th/>.



## REGLEMENT

- Art. 1. De Olympiades in de Informatica (OI) zijn wedstrijden in programmeren, algoritmeek en logica, georganiseerd door Belgische universiteiten en hogescholen waar opleidingen informatica gedoceerd worden.
- Art. 2. De OI worden gecoördineerd door een coördinatiecomité (CCOI) bestaande uit onderwijzend personeel van Belgische universiteiten en hogescholen. Het CCOI is verantwoordelijk voor het beheer van alle aspecten van deze competitie.
- Art. 3. Door hun inschrijving aanvaarden de kandidaten alle bepalingen van dit wedstrijdreglement.
- Art. 4. De officiële talen van de OI zijn het Nederlands en het Frans. De deelnemers moeten een taal kiezen (Frans of Nederlands) waarin ze aan de wedstrijd willen deelnemen.
- Art. 5. De kandidaten moeten in België verblijven en moeten ingeschreven zijn aan een middelbare school, hogeschool of universiteit gevestigd in België. De kandidaten moeten jonger zijn dan 25 jaar op de eerste mei van het jaar waarin de OI worden gehouden.
- Art. 6. De OI bestaan uit twee wedstrijden:
  - (a) De eerste wedstrijd is bestemd voor leerlingen van het algemeen, technisch en beroeps secundair onderwijs van België.
  - (b) De tweede wedstrijd is bestemd voor studenten in hun eerste jaar hoger onderwijs, d.w.z. zij die nog geen 60 credits van hun huidige bacheloropleiding hebben behaald.
- De organisatoren van de OI mogen de kandidaten verzoeken een attest van inschrijving in een school voor te leggen.
- Art. 7. De vragen worden door het CCOI opgesteld. Bij iedere vraag staat een indicatie van moeilijkheidsgraad. De vragen zijn verschillend voor elk van de wedstrijden van de OI. De halve finales zijn volledig schriftelijk. Voor de finales zijn er twee soorten vragen:
  - (a) Op het eerste soort vragen moeten de kandidaten schriftelijk antwoorden. De antwoorden mogen in pseudo-code of in één van de toegestane programmeertalen gegeven worden.
  - (b) Op het tweede soort vragen moeten de kandidaten antwoorden door een programma te schrijven op de computer in één van de toegestane programmeertalen.
- De toegestane programmeertalen zijn: Java, C, C++, Pascal, Python en PHP.
- Art. 8. De wedstrijdjury die door het CCOI is samengesteld verbetert de antwoorden van de kandidaten, maakt een rangschikking van de kandidaten en beslist wie de winnaars zijn van beide wedstrijden. Leden van deze jury zijn leden van universiteiten en hogescholen van België.
- Art. 9. De OI verlopen in twee rondes: halve finales en finales. De data van deze wedstrijden zijn door het CCOI vastgesteld en in de kalender gepubliceerd. De praktische organisatie van deze twee stappen wordt in specifieke reglementen\* beschreven.
- Art. 10. Maximum 60 kandidaten met de beste resultaten uit de halve finales van elke wedstrijd, genomen over alle regionale centra heen, zullen deelnemen aan de finale.
- Art. 11. De winnaars van de finales van de wedstrijden zullen prijzen krijgen.
- Art. 12. Minstens zes finalisten (drie Franstaligen en drie Nederlandstaligen) van de wedstrijd voor de leerlingen van het secundair onderwijs, worden door het CCOI gekozen om deel uit te maken van een pool van leerlingen, waaruit deelnemers van het Belgische nationale team voor de Internationale Olympiades in Informatica (IOI) zullen gekozen worden. De kandidaten hiervoor moeten ook aan de volgende bijkomende criteria voldoen: jonger zijn dan 20 jaar zijn op 1ste juli van het jaar waarin de IOI wordt gehouden, en de Belgische nationaliteit hebben. Het aantal deelnemers aan de pool wordt vastgelegd door het CCOI.
- Art. 13. Een kandidaat die weigert deel te nemen aan de IOI pool kan door een andere finalist(e) vervangen worden. Deze beslissing wordt door het CCOI genomen.
- Art. 14. In alle gevallen niet voorzien in bovenstaand reglement beslist het CCOI. Tegen beslissingen van het CCOI is geen beroep mogelijk.

\*Reglementen van de Halve Finales and Finales zijn beschikbaar op de website : <http://www.be-oi.be>.

worden. Schrijf een functie die de staat van de stock inleest en de nieuwe staat van de stock teruggeeft, na controle en eventuele vernietiging van sommige producten.

## Voor het hoger onderwijs

### 1. Programmeren (niveau gemakkelijk)

Schrijf een programma dat twee reeksen van karakters inleest van standaard input. De eerste reeks  $W$  is een woord bestaand uit letters ( $a-zA-Z$ ) en de tweede reeks  $P$  is een motiefje bestaand uit letters ( $a-zA-Z$ ) en het symbool  $_$ . Het programma moet testen of reeks  $P$  een subreeks is van reeks  $W$  (d.w.z.  $P$  komt voor in  $W$ ), als  $_$  eender welke letter kan voorstellen.

### 2. Algoritmiek (niveau gemiddeld)

Je hebt 2 functies `moveElem` en `minIndex`, waarvan de specificaties hieronder gegeven zijn. Schrijf een algoritme dat toestaat om een array van grootte  $n$  oplopend te sorteren, door enkel deze twee functies te gebruiken.

De functie `moveElem(arr, i, j)` laat toe de elementen op indices  $i$  en  $j$  van de array `arr` om te wisselen, waarbij  $i$  en  $j$  geldige indices zijn op `arr`. De functie `minIndex(arr, i)` geeft je de index van een kleinste element van de deel-array `arr[i:n-1]`; het geeft een index  $j$  terug zodat  $\forall k \in [i, n - 1] : arr[k] \geq arr[j]$ , waarbij index  $i$  een geldige index is op `arr`.

### 3. Algoritmiek (niveau gemiddeld)

**Context:** Een lijst kan gezien worden als bestaande uit twee delen: een kop ( $H$ ) en een staart ( $T$ ). Zo kunnen we een lijst voorstellen als  $L = H | T$ . De enige operaties die toegelaten zijn op de lijst zijn `head(L)` die je de kop teruggeeft, en `tail(L)` die je de staart geeft. De lege lijst wordt voorgesteld als `nil`.

**Voorbeeld:** Gegeven de lijst  $L = 1|2|3|4|5$ . De kop van de lijst is `head(L) = 1` en de staart ervan is `tail(L) = 2|3|4|5`. De kop van de staart is `head(tail(L)) = 2` en de staart daarvan is `tail(tail(L)) = 3|4|5`, etc.

**Opdracht:** Schrijf een algoritme dat test of de eerste en laatste elementen van een niet-lege lijst  $L$  van lengte  $n$ , identiek zijn of niet. Je mag de functies `head` en `tail` maximaal  $n$  keer aanroepen.

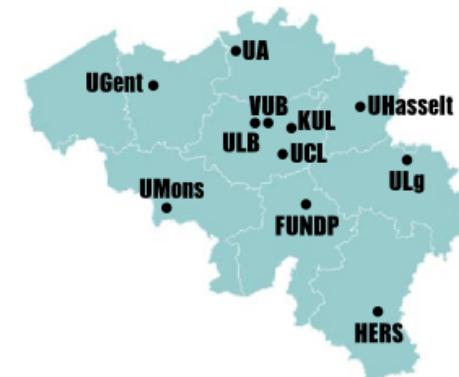
De oplossingen voor deze vragen zullen beschikbaar zijn op de officiële website. Je vindt er eveneens andere voorbeelden, alsook een introductie tot de algoritmiek en een gids over pseudo-code.

## ORGANISATIE VAN DE OLYMPIADES

De **halve finales** vinden plaats in meerdere regionale centra in België, op *woensdag 16 februari 2011 van 14u tot 17u*. Elk centrum kan het onthaal van de kandidaten slechts in één taal garanderen. De kandidaten mogen de proef wel afleggen in de taal die zij wensen, onafhankelijk van het gekozen centrum. De twee proeven, voor het secundair en het hoger onderwijs, zijn schriftelijk.

De 11 regionale centra zijn de volgende:

- Universiteit Antwerpen (UA);
- Vrije Universiteit Brussel (VUB);
- Université Libre de Bruxelles (ULB);
- Katholieke Universiteit Leuven (KUL);
- Universiteit Gent (UGent);
- Universiteit Hasselt (UHasselt);
- Haute École Robert Schuman de Libramont (HERS);
- Université de Liège (ULg);
- Université catholique de Louvain (UCL);
- Université de Mons (UMons);
- Facultés Universitaires Notre-Dame de la Paix de Namur (FUNDP).



Informatie over de exacte locatie van de verschillende centra en plattegronden bevindt zich op de officiële website. De **finales** worden georganiseerd aan de UCL te Louvain-La-Neuve, op *woensdag 20 maart 2011 van 14u tot 18u*. De proef zal 4 uur duren. De twee finalewedstrijden bestaan elk uit 2 delen: een deel op papier en een deel aan de computer. De proclamatie van de winnaars zal plaatsvinden op *zaterdag 2 april 2011*. Er zijn talrijke prijzen te winnen.

## OPLEIDINGEN

Tussen de proeven door worden er korte cursussen georganiseerd voor de leerlingen van het secundair onderwijs. De eerste cursus vindt plaats tijdens de krokusvakantie en duurt 2 dagen. De eerste dag is gewijd aan een introductie tot de algoritmiek en de tweede dag zal besteed worden aan meer gevorderde oefeningen. Deze opleidingen zijn optioneel.

Tenminste 6 laureaten van de finale voor het secundair onderwijs zullen het aanbod krijgen om deel uit te maken van een pool waaruit de leden van het Belgisch team voor de IOI geselecteerd zullen worden. De tweede cursus, die tijdens de paasvakantie zal plaatsvinden, is verplicht en is bestemd voor deze geselecteerde laureaten.

Meer informatie over deze cursussen zal aan de deelnemers worden meegegeven en wordt ook op de officiële website gezet.

## AGENDA

20 december 2010	Lancering van de OI en begin van de inschrijvingsperiode.
10 februari 2011	Uiterste datum om in te schrijven.
16 februari 2011	<b>Halve Finales</b> in regionale centra.
7-11 maart 2011	Tweedaagse cursus Algoritmiek (facultatief) voor leerlingen uit het middelbaar onderwijs (krokusvakantie).
30 maart 2011	<b>Finales</b> in Louvain-La-Neuve.
2 april 2011	Bekendmaking en prijsuitreiking.
11-24 april 2011	Vierdaagse cursus Algoritmiek voor leerlingen uit het middelbaar onderwijs geselecteerd voor de IOI pool (Paasvakantie, exacte data nog te bepalen).
22-29 juli 2011	23 <sup>de</sup> Internationale Olympiade in de Informatica (IOI) in Pattaya, Thailand ( <a href="http://www.ioi2011.or.th">http://www.ioi2011.or.th</a> ).

## VRAGEN

Tijdens de wedstrijd worden de deelnemers getest op hun kennis en vaardigheden in programmeren, algoritmiek en logica. De halve finales zijn volledig schriftelijk; de finales zijn deels schriftelijk en deels aan de computer. Hieronder staan enkele voorbeeldvragen voor het schriftelijk gedeelte. Op de website van de Belgische OI zijn meer voorbeeldvragen te vinden, alsook de vragen van vroegere edities.

### Voor het secundair onderwijs

#### 1. Logica (niveau gemiddeld)

Met behulp van twee zandlopers (één van 7 minuten en één van 4 minuten) moet je 9 minuten afmeten. Je mag de zandlopers al manipuleren voordat je de 9 minuten begint te tellen. Vind de oplossing die in totaal het minste tijd in beslag neemt.

#### 2. Algoritmiek (niveau gemakkelijk)

Gegeven een geheel getal  $n > 0$ . Schrijf een algoritme dat het aantal positieve gehele delers van  $n$  berekent.

#### 3. Algoritmiek (niveau gemiddeld)

Alice en Bob beschikken over een groot aantal (veronderstel oneindig veel) rechthoekige speelkaarten, allemaal even groot. Ze willen ze naast elkaar leggen, allemaal in dezelfde richting en zonder ze te overlappen of te snijden, om een vierkant te vormen.

Help hen door de minimale grootte van dit vierkant te vinden. Als  $L$  de lengte is van een kaart en  $H$  de hoogte (allemaal strikt positieve gehele waarden), vragen wij om een algoritme te schrijven dat  $X$  kan berekenen:  $X$  is het aantal kaarten dat nodig is om een vierkant te vormen.

#### 4. Algoritmiek (niveau moeilijk)

Je hebt de leiding over een magazijn waarin je  $X$  producten kan opslaan. Elke dag komt er een vrachtwagen die  $Y$  nieuwe producten brengt en de  $Y$  oudste producten uit de stock wegneemt. De producten aangeleverd door de vrachtwagen zijn altijd nieuw. De producten in stock en de producten in de vrachtwagen worden voorgesteld in lijsten:  $stock = \langle s_1, s_2, \dots, s_X \rangle$ ;  $vrachtwagen = \langle c_1, c_2, \dots, c_Y \rangle$ .

(a) Schrijf een algoritme dat de staat van de stock en de inhoud van de vrachtwagen inleest, en de staat van de stock aan het eind van de dag teruggeeft.

(b) Breid het algoritme uit zodat het ook overweg kan met vrachtwagens die producten van verschillende leeftijd leveren. Je kan de ouderdom van een product  $a$  te weten komen met de functie  $age(a)$ .

(c) Elke avond doet een werknemer de ronde om te controleren of er geen producten zijn die te oud zijn om verkocht te mogen worden. Een product dat ouder is dan 126 dagen (op het moment van controle) moet vernietigd