

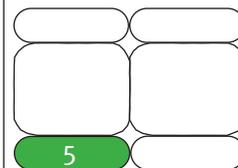


# OLIMPIADI DELL'INFORMATICA

## Giovani talenti cercansi

Le Olimpiadi in Informatica, sponsorizzate dall'UNESCO e riservate agli studenti delle scuole superiori, rappresentano la più prestigiosa gara internazionale di questo genere. Il MIUR ha affidato all'AICA il compito di gestire l'organizzazione e la partecipazione italiana a tali gare a partire dal 2000, istituendo inoltre le Olimpiadi Italiane dal 2002. Questo articolo illustra il processo di selezione e i contenuti dei temi di gara, commentando i risultati delle gare e la preparazione in informatica richiesta ai candidati.

**Bruno Fadini**  
**Roberto Grossi**



### 1. COSA SONO LE IOI

**L**e Olimpiadi Internazionali di Informatica (**IOI**, *International Olympiad in Informatics*) hanno l'obiettivo di scoprire, incoraggiare, far incontrare, mettere in competizione e dare riconoscimento ai giovani che nel mondo mostrano maggior talento in informatica nonché di promuovere l'amicizia fra studenti e docenti di diversa cultura. La prima edizione delle IOI si è svolta nel 1989 in Bulgaria, su iniziativa e con il patrocinio dell'UNESCO (*United Nations Educational, Scientific and Cultural Organization*) e si sono svolte con cadenza annuale; l'ultima edizione ha avuto luogo in Polonia mentre quella del 2006 si svolgerà in Messico. Esse rappresentano la più prestigiosa gara internazionale di informatica riservata agli studenti delle scuole superiori: vedono la partecipazione di oltre 80 Paesi, ciascuno con una squadra di al più quattro studenti. Gli studenti gareggiano individualmente in due giorni distinti (con tre problemi di natura algoritmica e di *problem solving* da risolvere in ciascun giorno, entro cinque ore), trascorrendo ulteriori cinque

giorni in attività socio-culturali. In coerenza con gli obiettivi della competizione, la premiazione avviene a fasce per cui la metà dei

#### Edizioni delle IOI

1. 1989 - Pravetz, Bulgaria, 16-19 maggio
2. 1990 - Minsk, Repubblica Bielorussa, Unione Sovietica, 15-21 luglio
3. 1991 - Atene, Grecia, 19-25 maggio
4. 1992 - Bonn, Germania, 11-21 luglio
5. 1993 - Mendoza, Argentina, 16-25 ottobre
6. 1994 - Haninge, Svezia, 3-10 luglio
7. 1995 - Eindhoven, Paesi Bassi, 26 giugno - 3 luglio
8. 1996 - Veszprém, Ungheria, 25 luglio - 2 agosto
9. 1997 - Cape Town, Repubblica del Sudafrica, 30 Novembre - 7 dicembre
10. 1998 - Setúbal, Portogallo, 5-12 settembre
11. 1999 - Antalya-Belek, Turchia, 9-16 ottobre
12. 2000 - Pechino, Cina, 23-30 settembre
13. 2001 - Tampere, Finlandia, 14-21 luglio
14. 2002 - Yong-In, Repubblica Coreana, 18-25 agosto
15. 2003 - Kenosha, Wisconsin, USA, 16-23 agosto
16. 2004 - Atene, Grecia, 11-18 settembre
17. 2005 - Nowy Sacz, Polonia, 18-25 agosto
18. 2006 - Mérida, Yucatán, Messico, 13-20 agosto

partecipanti ottiene una medaglia: precisamente, un dodicesimo dei partecipanti vince la medaglia d'oro, un sesto quella d'argento e un quarto la medaglia di bronzo. Il sito ufficiale delle IOI è [www.ioinformatics.org](http://www.ioinformatics.org).

L'Italia partecipa alle IOI a partire dal 2000 con il supporto scientifico ed economico da parte del MIUR (Ministero dell'Istruzione, dell'Università e della Ricerca) e dell'AICA (Associazione Italiana per l'Informatica e il Calcolo Automatico) che ne fu la promotrice, riconoscendo l'alto valore educativo di tale iniziativa come strumento di sviluppo dell'informatica. L'obiettivo primario affianca, agli obiettivi generali delle IOI, quello di stimolare l'interesse dei nostri giovani verso gli aspetti scientifici dell'Informatica e di contribuire alla sua conoscenza e diffusione nelle scuole italiane.

## 2. LE OLIMPIADI DI INFORMATICA IN ITALIA

Nel 2000, nel contesto di un accordo di collaborazione MIUR-AICA che prevedeva anche lo sviluppo dell'ECDL e uno studio sui programmi di informatica nelle scuole, il MIUR ha affidato all'AICA il compito di gestire l'organizzazione e la partecipazione italiana alle gare internazionali delle IOI. La gestione avviene tramite il **Comitato Olimpico**, una struttura paritetica MIUR-AICA, che definisce l'organizzazione e la sovrintende e il Gruppo dei Selezionatori Nazionali (ovvero gli "allenatori"), che si occupa della formazione scientifica e, in collaborazione con il Comitato, del processo di selezione degli studenti per la gara internazionale. Dal 2002, poi, il Comitato Olimpico ha istituito le "OII, Olimpiadi italiane di informatica" che si svolgono in concomitanza con il processo di selezione.

La partecipazione in sole sei edizioni ha prodotto un discreto numero di medaglie: 1 oro, 4 argenti e 8 bronzi (si veda il riquadro a p. 11). Il cammino verso la selezione e la preparazione della squadra italiana si svolge durante tutto il corso dell'anno ed è organizzato in quattro fasi di selezione:

- **Scolastica:** vengono scelti un migliaio di studenti tra gli oltre 10.000 partecipanti.
- **Regionale:** vengono scelti 80 studenti tra il migliaio a disposizione.
- **Nazionale:** viene svolta la gara nazionale,

### Comitato Olimpico Italiano e Gruppo dei Selezionatori Nazionali

#### Presidente:

- Prof. Fadini - Università "Federico II" di Napoli

#### Esponenti AICA:

- Dott.ssa Genovì De Vita
- Prof. Casadei - Università di Bologna
- Prof. Italiani - Università di Milano
- Prof. Marchetti Spaccamela - Università "La Sapienza" di Roma

#### Esponenti MIUR:

- Dott.ssa Davoli - Dir. Gen. Ordinamenti Scolastici
- Preside Affronti - Liceo Classico V. Emanuele II - Palermo
- Preside Calabrò - ITIS Panella - Reggio Calabria
- Preside Di Rienzo - ITC Tosi - Busto Arsizio (VA)

#### Selezionatori Nazionali:

- Dott. Chierichetti - Università "La Sapienza" di Roma
- Prof. Grossi - Università di Pisa (coordinatore)
- Prof. Laura - Università "La Sapienza" di Roma
- Prof. Rizzi - Università di Udine
- Prof. Santini - Università di Milano

#### Leader e Deputy Leader alle IOI:

- Prof. Boldi - Università di Milano (IOI 2000-2003)
- Prof. Grossi - Università di Pisa (IOI 2004-2005)
- Prof. Rizzi - Università di Udine (IOI 2005)
- Prof. Vigna - Università di Milano (IOI 2000-2004)

denominata Olimpiadi Italiane di Informatica, individuando i vincitori nazionali e selezionando in totale un gruppo di 15 studenti chiamati "Probabili Olimpici".

□ **Olimpica:** vengono scelti i 4 componenti della squadra italiana per partecipare alle IOI, più 2 riserve, tra i 15 Probabili Olimpici dopo un periodo di allenamento e formazione residenziale e telematica con i Selezionatori Nazionali.

Nel seguito descriviamo le tappe di un tale cammino per dare un'idea di come le selezioni avvengano in concreto. Immaginiamo che uno studente che abbia ricevuto in regalo un calcolatore...

## 3. PRIMA DELLE OLIMPIADI: DAL GIOCO ALLA PROGRAMMAZIONE

Il nostro amico studente inizia a utilizzare il calcolatore per attività ludiche e di svago. Successivamente, spinto dalla curiosità e dalla volontà di dominare la macchina (è lo stesso impulso che spingeva le vecchie generazioni a smontare i giocattoli di allora) ne cambia la configurazione, sviluppa esperimenti... Il passo per giungere alla decisione



di programmarlo è vicino; infatti, anche se non tutti gli appassionati di videogiochi vogliono programmare, molti desiderano crearne di nuovi e nasce così la prima spinta verso la programmazione... Lo studente si rivolge ad amici più esperti, naviga in rete o va in biblioteca alla ricerca di materiale adatto alle proprie conoscenze. L'impatto è duro: l'eletrodomestico-calcolatore si trasforma in una macchina infernale difficile da controllare e le conoscenze richieste sembrano immense. Molti studenti motivati riescono a superare questa fase e aspirano a confrontarsi con altre persone che condividono gli stessi interessi. Riuscire a mettere gli studenti di questa comunità invisibile in contatto tra loro è già un successo che ripaga gli organizzatori delle Olimpiadi Italiane.

#### **4. IL PRIMO CONTATTO CON LE OLIMPIADI: LA FASE SCOLASTICA**

C'è attività frenetica in molte scuole durante il mese di novembre per l'organizzazione della selezione scolastica. Il numero delle scuole iscritte alle Olimpiadi è in costante crescita e oggi supera le 500 con oltre 10.000 studenti degli ultimi 3 anni delle superiori. La partecipazione della scuola è spesso sollecitata dagli insegnanti specifici di informatica, e in qualche caso anche dagli studenti. La gara viene gestita localmente dagli insegnanti che rivestono il ruolo di Referenti Scolastici. Il nostro studente deve affrontare una serie di semplici esercizi senza l'ausilio del calcolatore: quesiti logico-matematici e di programmazione (in Pascal e C/C++), a risposta libera o multipla. E qui occorre segnalare l'iniziale delusione del nostro studente autodidatta e sedicente esperto che si trova a competere su un campo che a lui non sembra quello dell'informatica da lui stesso inseguita (che c'entra la matematica? e i quiz di logica e poi...linguaggi obsoleti come il Pascal!). Ma il nostro studente è preparato e comunque supera brillantemente la prova, piazzandosi al primo posto nella scuola (adottando un criterio misto tra il rappresentativo e il meritocratico, sono ammessi alla successiva selezione regionale i primi 2 di ogni scuola e i rimanenti migliori in assoluto fino a coprire la capienza delle sedi di gara). Ha anche compre-

so (e lo comprenderà meglio durante lo sviluppo delle Olimpiadi) che la matematica e la logica sono componenti fondamentali dell'informatica, che Pascal e C++, oltre a essere i linguaggi ufficiali delle IOI, ben si prestano a esprimere problemi di natura algoritmica e che, soprattutto, l'informatica come disciplina scientifica non è lo smantellamento al calcolatore ma una disciplina ricca di teorie e metodologie per la risoluzione di problemi. Inoltre (e queste sono le motivazioni del Comitato Olimpico), l'uso dei quesiti logico-matematici permette di individuare potenziali candidati che hanno buone possibilità di migliorare durante le fasi successive, anche se non già esperti in informatica (si veda il riquadro a p. 13).

Ovviamente, occorre sottolineare che, nonostante gli obiettivi di un'ampia partecipazione, la manifestazione è pur sempre una gara alla ricerca di talenti e che dunque lo studio è di fondamentale importanza, ma l'impegno a trovare motivazioni interiori per crescere scientificamente è altrettanto importante, ovviamente oltre al talento che diversi possiedono. È anche importante coinvolgere i Referenti Scolastici attraverso un forum di discussione che incoraggi a loro volta una crescita professionale. È al miglioramento di questo livello di interazione che il Comitato Olimpico punta e, per questo, ha supportato la creazione di un portale nel [www](http://www.olympiadi-informatica.it) per tutti coloro che sono interessati a questo tipo di attività ([www.olympiadi-informatica.it](http://www.olympiadi-informatica.it)).

#### **5. L'APPROFONDIMENTO: LA FASE REGIONALE**

Gennaio è un mese critico per il nostro studente perché ha luogo la selezione regionale. Durante i mesi di novembre e dicembre ha dovuto impegnarsi in modo significativo per acquisire abilità programmatiche nettamente superiori a quelle necessarie per la precedente selezione. Adesso i problemi da affrontare hanno una natura più algoritmica e richiedono una certa dose di creatività per risolverli durante la gara, in modo da far emergere gli studenti che sono dotati di capacità algoritmiche elementari (vedi esempi a p. 14). È previsto l'uso del calcolatore (finalmente! pensa il nostro amico studente) per cui bisogna trovare il metodo di soluzione, programmarlo in

Pascal o C/C++, compilare, fare una prova di esecuzione... La competizione è diventata più impegnativa: il nostro studente probabilmente sente la necessità di un approfondimento culturale, scientifico e tecnologico. Vuole conoscere meglio l'architettura di un computer, i sistemi operativi (ha sentito parlare di questo mitico Linux), i costrutti dei linguaggi di programmazione e i loro compilatori, gli algoritmi efficienti di risoluzione dei problemi computazionali. Fortunatamente il World Wide Web permette l'accesso a diverse collezioni di esercizi e di altro materiale a disposizione in lingua inglese: è sufficiente immettere i termini *Olympiad Informatics* oppure *IOI tasks* in un motore di ricerca per ottenere vari risultati. Sul fronte interno, il citato portale fornisce alcuni forum di discussione, diversi riferimenti utili e ulteriore materiale didattico in italiano. Il Comitato Olimpico sta anche studiando i mezzi per sollecitare iniziative di formazione regionali, con l'ausilio delle direzioni scolastiche e delle università locali, già messe in atto da alcune regioni (Campania, Emilia-Romagna, Liguria, Lombardia, Trentino Alto-Adige). Rispetto alla fase scolastica, anche la sede di

gara è cambiata: quest'anno vi sono 23 sedi regionali dislocate sul territorio (per la numerosità delle scuole partecipanti, alcune regioni hanno due sedi di gara) e la gestione della gara viene affidata a insegnanti che rivestono il ruolo di Referenti Regionali. I testi dei problemi da risolvere arrivano presso il server locale delle sedi regionali attraverso un server centrale che li comunica in forma criptata, per cui neanche i Referenti ne conoscono il contenuto. Quando inizia la gara, viene resa pubblica la chiave crittografica e, al momento della consegna elettronica, il si-

stema di gara prevede che gli elaborati dei gareggianti siano nuovamente criptati con un'altra chiave segreta. In questo modo, la distribuzione sul territorio dei problemi di gara e la successiva raccolta degli elaborati possono avvenire in modo asincrono tra le varie sedi senza avvantaggiare chi ha potenti mezzi di comunicazione, nel rispetto delle ragionevoli regole di correttezza sportiva e di equità. Al contrario delle scolastiche, la correzione degli elaborati è centralizzata e automatica, con un sistema simile a quello descritto in seguito per la selezione nazionale.

Circa un migliaio di studenti partecipa a questa fase e soltanto 80 di essi passano alla fase successiva: in analogia alle scolastiche, i primi 2 di ogni sede e i rimanenti migliori in assoluto fino a coprire la capienza della sede di gara nazionale.

## 6. IL SALTO QUALITATIVO: LA FASE NAZIONALE

La fase nazionale del processo di selezione è anche, dal 2002, la gara per le "Olimpiadi italiane di Informatica". Useremo una metafora atletica per dare un'idea dell'impegno richiesto per lo studio e la risoluzione di problemi di difficoltà crescente in vista della preparazione per la fase nazionale ai primi di marzo, data in cui si svolgono le gare per le Olimpiadi Italiane di Informatica. Volendo correre i 100 metri, siamo tutti in grado di farlo in tempi più o meno ragionevoli. Pochi sanno farlo sotto i 15" (gli studenti selezionati alle regionali), ma alle gare nazionali occorre scendere sotto i 12" e alle internazionali anche sotto i 10". Alle gare nazionali di informatica siamo di fronte a una parete a picco e molti non sanno scalarla, neanche impiegando un mese: alcuni dei problemi sono irrisolvibili se non abbiamo l'intuizione giusta oppure un profondo allenamento non disgiunto da quel pizzico di intuizione e talento. Le strategie standard di risoluzione dei problemi aiutano, ma dobbiamo scovarle per il caso specifico e dobbiamo associare a esse l'inventiva e la velocità per realizzare in codice quanto escogitato. Il nostro studente deve impiegare la massima disciplina e concentrazione per superare la prova e non perdere tempo.

### Prerequisiti scientifici per la partecipazione alle selezioni nazionali

- Ottima conoscenza di un linguaggio di programmazione tra Pascal, C e C++
- Tecniche di programmazione:
  - memoria dinamica (allocazione/deallocazione) e puntatori
  - array e aritmetica dei puntatori
  - gestione dei file di input/output (testo e binari)
  - strutture e tipi di dati
  - ricorsione (funzioni che invocano se stesse)
- Strutture di dati:
  - pile e code
  - liste concatenate
  - alberi binari (di ricerca o meno)
  - alberi generali
  - grafi e loro rappresentazione
- Algoritmi di base:
  - ordinamento (quicksort o mergesort)
  - ricerca binaria o dicotomica
  - visite di alberi (anticipata, simmetrica, posticipata)
  - visite di grafi (BFS, DFS).
- Cenni a tecniche algoritmiche:
  - divide et impera
  - greedy
  - programmazione dinamica

La gara si svolge sulla falsariga delle gare internazionali delle IOI ma con problemi relativamente più semplici (ma comunque difficili!); vedi esempi a p. 14. Partecipano circa 80 studenti in un'unica sede di gara (quest'anno, al Politecnico di Milano), impiegando una rete di calcolatori con caratteristiche identiche ma con la possibilità di usare uno qualunque dei due sistemi operativi più diffusi (Linux e Windows). La novità dell'evento risiede nel fatto che gli studenti entrano a contatto diretto per la prima volta con i loro Selezionatori Nazionali, i quali gestiscono completamente il sistema di gara ed effettuano la valutazione degli elaborati ai fini della classifica finale. Il sistema di correzione utilizza Linux e i compilatori g++ e gcc (*GNU C/C++ Compiler*) e fpc (*Free Pascal Compiler*). Oltre al tempo limite di cinque ore per risolvere i tre problemi (come in una giornata di gara delle IOI), esiste un ulteriore limite sul tempo massimo di esecuzione, in genere di pochi secondi. Infatti, non solo le soluzioni proposte devono poter essere compilate senza errori nel sistema di correzione, ma devono terminare e restituire il risultato corretto *entro* il tempo limite. I concorrenti hanno la possibilità di abituarsi a questo sistema il pomeriggio precedente al giorno di gara.

I vincitori delle Olimpiadi Italiane ricevono le medaglie a fasce con una regola simile a quella per le IOI: 5 medaglie d'oro, 10 d'argento e 20 di bronzo. Inoltre, i primi 15 classificati (oro e argento) sono ammessi alla fase olimpica come Probabili Olimpici. La premiazione avviene in forma ufficiale alla presenza delle autorità, della stampa e della radio-televisione.

## 7. LA SQUADRA ITALIANA: LA FASE OLIMPICA

Lo *status* di Probabile Olimpico per il nostro studente comporta un impegno significativo, considerato che va integrato con i normali impegni scolastici. La fase olimpica dura circa tre mesi, fino a maggio, e prevede due corsi residenziali tenuti da docenti universitari e tutor (solitamente ex-olimpici) del Gruppo di Selezionatori Nazionali (ultimamente gli studenti e gli allenatori sono

ospitati dall'Università di Pisa e dalla Scuola Normale Superiore). È un'esperienza di alto valore educativo, non solo dal lato scientifico, ma anche umano e sociale: in tale contesto, spesso si formano delle amicizie che proseguono ben oltre la normale attività. Il clima è molto collaborativo e i Selezionatori hanno la possibilità di conoscere meglio le attitudini dei Probabili Olimpici durante il periodo di formazione. Vengono discussi argomenti e tecniche di risoluzione più evolute, e sono affrontati insieme svariati problemi delle IOI così da entrare nel clima di gara. L'allenamento prosegue anche al di fuori dei due corsi residenziali, che solitamente durano circa una settimana l'uno e sono tenuti a ridosso delle vacanze di Pasqua e del "ponte" tra il 25 Aprile e il Primo Maggio. I Probabili Olimpici ricevono dei libri di testo specializzati su algoritmi e programmazione, e vengono incoraggiati a risolvere per proprio conto i problemi delle IOI precedenti in ambiente Linux usando i linguaggi C/C++ con il compilatore gcc/g++ (gli strumenti ufficiali per la valutazione degli elaborati alle IOI). Alla fine del secondo corso è prevista una gara con le stesse modalità delle nazionali ma con problemi più

### Studenti che hanno vinto la medaglia d'oro alle Olimpiadi nazionali (OII)

#### OII 2003:

1. Stefano Maggiolo – ITI Severi di Padova
2. Stefano Soffia – Liceo Scientifico Aldo Moro di Reggio Emilia
3. Andrea Bergia – ITI G. Vallauri di Fossano (CN)
4. Samuel Kerschbaumer – ITI Max Valier di Bolzano
5. Andrea Matiatic – Liceo Scientifico France Preseren di Trieste
6. Nicola Pasta – ITI Majorana di Seriate (BG)
7. Dieter Steiner – ITI Max Valier di Bolzano
8. Matteo Brigo – ITI Carlo Zuccante di Venezia

#### OII 2004:

1. Alessandro Piva - Liceo Scientifico Majorana di Latina
2. Alessio Caiazza - ITI Antonio Meucci di Firenze
3. Luca Barbieri - Liceo Scientifico Donatelli Pascal di Milano
4. Alessio Guerrieri - Liceo Scientifico Leonardo da Vinci di Trento
5. Stefano De Rossi - Liceo Scientifico Galileo Galilei di Dolo (VE)
6. Paolo Giarrusso - Liceo Scientifico Galileo Galilei di Catania
7. Dario Zappalà - Liceo Scientifico Galileo Galilei di Catania
8. Marco Signorini - ITI Galileo Galilei di Livorno
9. Matthias Silbernagl - ITI Max Valzer di Bolzano

#### OII 2005:

1. Luca Barbieri - Liceo Scientifico Carducci di Milano
2. Giorgio Audrito - Liceo Scientifico M. Curie di Pinerolo (TO)
3. Andrea Canciani - Liceo Scientifico G. D. Cassini di Genova
4. Francesco Bonazzi - Liceo Scientifico Leonardo da Vinci di Milano
5. Marco Bonizzato - Liceo Scientifico Galileo Galilei di Verona

difficili: è sorprendente vedere come il livello scientifico di molti Probabili Olimpici evolva in pochi mesi!

La selezione della squadra dei quattro studenti, e delle due riserve, si basa quindi sui risultati della gara nazionale, della gara suddetta e di un colloquio che i candidati hanno con i membri del Comitato Olimpico e del Gruppo di Selezionatori Nazionali. I quattro della squadra ricevono in regalo un portatile per continuare ad allenarsi telematicamente con alcuni dei Selezionatori fino al momento della partenza per le IOI. I loro nomi vengono ufficialmente comunicati dal Comitato agli organizzatori delle IOI per l'inserimento nell'albo dei partecipanti.

### **8. LA PARTENZA: LA FASE INTERNAZIONALE**

Per i componenti della squadra, l'attività di allenamento non si ferma a maggio ma prosegue telematicamente fino alla settimana di gara delle IOI, utilizzando il server italiano di gara del portale ufficiale delle Olimpiadi Italiane e quello russo del *St. Petersburg Institute of Fine Mechanics and Optics* e partecipando a varie gare in linea che si tengono in altri Paesi (nel passato, si è anche partecipato ufficialmente alle "Olimpiadi europee" e si sono anche organizzati raduni di preparazione prima della partenza). Infine, l'appuntamento è all'aeroporto di Milano o di Roma per partire insieme alla volta della nazione che ospita la gara delle IOI, ove si ritrovano oltre 500 persone tra studenti, accompagnatori e guide locali. Gli accompagnatori sono coinvolti in una serie di incontri tecnici che portano alla definizione dei problemi di gara e in una serie di assemblee per deliberare su questioni gestionali del comitato internazionale. Gli studenti, oltre alle due giornate di gara, partecipano a gite, attività culturali e sociali (talvolta insieme agli accompagnatori). Il primo giorno c'è la cerimonia di apertura e poi una sessione simulata per fare pratica con il sistema di gara. Il secondo e il quarto giorno sono dedicati alla gara, il terzo e il quinto allo svago e alle gite, il sesto e ultimo alla cerimonia solenne di premiazione. Indipendentemente dal fatto che si riesca a ottene-

re una medaglia o meno, è appagante aver superato una serie di selezioni per essere annoverati tra i migliori giovani talenti informatici a livello mondiale.

A p. 15 alcuni esempi di esercizi dati alle IOI.

### **9. CONCLUSIONI: OLIMPIADI, GARE DI INFORMATICA E SCUOLE**

Certamente, chi ama ascoltare musica, leggere libri o guardare film non è detto che voglia diventare musicista, scrittore o regista. Nel caso dell'informatica, la conclusione non è però così immediata: siamo sempre più sommersi dall'informazione e imparare a gestirla in modo appropriato è un dovere formativo degno quanto imparare a leggere e scrivere. Gli analfabeti del futuro saranno coloro che non sapranno organizzare e gestire selettivamente la marea di informazioni a cui siamo sottoposti. La conoscenza dell'informatica (e non solo) nelle scuole superiori è uno dei prerequisiti perché ciò avvenga, ricordandosi di non confondere lo strumento (il calcolatore e il *software*) con la scienza (l'informatica) come diceva Edsger Dijkstra, *premio Turing 1972* (il premio Nobel per l'informatica): "*Computer science is no more about computers than astronomy is about telescopes*".

Qual è il rapporto fra questa premessa e le Olimpiadi di Informatica? Queste ultime, insieme alle altre Olimpiadi per le scuole superiori (di Astronomia, Biologia, Chimica, Fisica e Matematica), non costituiscono semplicemente un motivo di orgoglio della comunità scientifica internazionale ma rappresentano un'importante cartina di tornasole per valutare la diffusione della cultura scientifica nelle scuole superiori e quindi delle generazioni future. Lo stato dell'insegnamento dell'informatica nelle scuole superiori italiane non è certamente fra i migliori: a parte gli Istituti tecnici specialistici e alcuni licei sperimentali, l'insegnamento è assente né la recente riforma ha realizzato la svolta che si sperava: gli Istituti tecnici di Informatica e gli istituti sperimentali confluiranno nelle otto tipologie di liceo (dal classico al tecnologico)... ove l'informatica non c'è!

Ma se Sparta piange...anche negli Stati Uniti ci si lamenta della insufficiente preparazione nelle scuole. Un'intensa discussione ha avuto luogo infatti in diversi forum e riviste (cfr. *slashdot.org*, *CNET News.com* e *Communication of the ACM*), a partire dalla primavera del 2005, in merito all'andamento della squadra statunitense nelle gare ACM-ICPC (*ACM International Collegiate Programming Contest*) organizzate dall'ACM. Poiché queste gare, rivolte agli studenti universitari, sono per il resto analoghe a quelle delle IOI, molte delle osservazioni che stiamo per riportare si applicano anche a quest'ultime e ai problemi della formazione scolastica.

Le pessime prestazioni della squadra statunitense nelle ultime gare ACM-ICPC (le peggiori in 29 anni di gare) vengono commentate non per semplici motivi campanilistici, ma in quanto campanello d'allarme di una sempre minore competitività statunitense nel panorama mondiale dell'ICT. Tale percezione non è propria solo degli USA ma anche di molti altri paesi industrializzati come l'Italia che temono l'invasione scientifica e tecnologica dei paesi dell'Est europeo e asiatico. Il segnale d'allarme viene lanciato da James Foley, professore al *Georgia Institute of Technology* e da Craig Barret, presidente dell'*Intel Corporation*: oltre al collasso delle aziende basate sulla *Internet economy* e all'affidamento dello sviluppo di software a esperti fuori dal paese (*outsourcing*), disincentivando di fatto molti giovani a investire il loro futuro nell'ICT, ha pesantemente contribuito l'educazione scolastica, poco sensibile a formare capillarmente una cultura informatica a livello di scuola superiore. Un anonimo dirigente di un'importante azienda americana aggiunge che la preparazione scientifica degli studenti statunitensi è agli ultimi posti tra i Paesi industrializzati, come riportato dall'*Organization for Economic Cooperation and Development* (per gli addetti ai lavori, il famoso PISA - *Programme for International Student Assessment*). David Patterson, professore della *University of California* a Berkeley, fornisce ulteriori considerazioni in merito. Innanzi tutto, c'è una maggiore

diffusione dell'ICT in alcuni Paesi in via di sviluppo in quanto non richiede enormi investimenti; inoltre, primeggiare in tali gare è una forma di orgoglio nazionale per cui si investe molto su pochi a scapito di molti (si pensi alle nuotatrici olimpiche della Germania dell'Est). A livello nazionale, diversi corsi di studio in informatica sono obsoleti (Patterson cita il corso di compilatori a Berkeley) e la ricerca scientifica non è finanziata sufficientemente: i continui tagli al finanziamento pubblico hanno tarpato le ali a molti laboratori di ricerca, fornendo implicitamente un messaggio negativo ai giovani. Patterson conclude quindi che un ridimensionamento è accettabile in quanto sono scomparsi diversi speculatori apparsi alla soglia dell'informatica durante il boom di Internet (chiaramente poco interessati alla disciplina in sé stessa): tuttavia, il timore di *outsourcing* ha eccessivamente spaventato le future generazioni di programmatori nazionali.

Nonostante qualche voce fuori del coro (per esempio, Norman Matloff, professore della *University of California* a Davis), ce n'è a sufficienza per riflettere sul panorama della preparazione degli studenti nelle scuole superiori: sul piano politico, partiamo in ritardo rispetto agli Stati Uniti e quindi potremmo non fare gli stessi errori...ma il non far niente non può che acuire il distacco se non verso gli Stati Uniti, almeno verso i Paesi dell'Est! Sul piano delle gare IOI, dobbiamo dichiararci molto soddisfatti per i risultati finora conseguiti, anche in assenza di una specifica preparazione scolastica, ma tutta basata sull'impegno e sul talento dei nostri studenti e sull'abnegazione della squadra degli allenatori.

### Ringraziamenti

Si ringrazia la dott.ssa Margherita Ortolani per il supporto organizzativo fornito durante la scrittura di questo articolo, il prof. Romeo Rizzi per aver collaborato alla traduzione dei testi riportati nella scheda di p. 15 e i vari colleghi che hanno contribuito alla stesura dei testi riportati nelle schede di p. 13 e 14. Inoltre, siamo riconoscenti al prof. Paolo Boldi, all'Ing. Rosario Cristaldi, al dott. Luca Foschini, al prof. Sebastiano Vigna e a tutti gli ex-olimpici,

per i loro significativi contributi nella creazione del sito Web per gli allenamenti.

### Bibliografia

- [1] Cooper C.: *Can the U.S. still compete?* CNET News.com, Apr. 2005 ([http://news.com.com/Can+the+U.S.+still+compete/2010-1071\\_3-5672106.html](http://news.com.com/Can+the+U.S.+still+compete/2010-1071_3-5672106.html))
- [2] Frauenheim E.: *U.S. slips lower in coding contest.* CNET News.com, Apr. 2005 ([http://news.com.com/U.S.+slips+lower+in+coding+contest/2100-1022\\_3-5659116.html](http://news.com.com/U.S.+slips+lower+in+coding+contest/2100-1022_3-5659116.html))
- [3] Frauenheim E.: *Can Johnny still program?* CNET News.com, Apr. 2005 ([http://news.com.com/Can+Johnny+still+program/2008-1036\\_3-5675770.html](http://news.com.com/Can+Johnny+still+program/2008-1036_3-5675770.html))
- [4] Matloff N.: *Globalization and the American IT worker.* *Commun. ACM*, Vol. 47, n. 11 (Nov. 2004), p. 27-29 (<http://doi.acm.org/10.1145/1029496.1029516>).
- [5] Matloff N.: *Johnny can so program.* CNET News.com, May 2005 ([http://news.com.com/Johnny+can+so+program/2010-1007\\_3-5700858.html](http://news.com.com/Johnny+can+so+program/2010-1007_3-5700858.html))
- [6] Patterson D.A.: *Reflections on a programming Olympiad.* *Commun. ACM*, Vol. 48, n. 7, (Jul. 2005), p. 15-16. (<http://doi.acm.org/10.1145/1070838.1070852>)
- [7] Patterson D.A.: *Restoring the popularity of computer science.* *Commun. ACM*, Vol. 48, n. 9 (Sep. 2005), p. 25-28 (<http://doi.acm.org/10.1145/1081992.1082011>).
- [8] Skiena S.S., Revilla M.: *Programming Challenges.* Springer, NY 2003 (ISBN: 0387001638).

BRUNO FADINI è professore ordinario di Calcolatori elettronici presso l'Università Federico II di Napoli dal 1973 e presidente del Comitato Olimpico italiano per le Olimpiadi di Informatica. Già presidente dell'AICA, ha diretto o partecipato a molte attività di ricerca nazionali ed internazionali. I suoi interessi scientifici e didattici sono oggi rivolti all'architettura dei sistemi di elaborazione ed ai processi di e-learning. È autore di oltre 100 pubblicazioni scientifiche e di 10 libri.

E-mail: fadini@unina.it

ROBERTO GROSSI è professore associato di Informatica presso l'Università di Pisa dal 1998 e allenatore della nazionale italiana per le Olimpiadi Internazionali di Informatica. I suoi interessi scientifici e didattici sono rivolti al progetto, all'analisi e alla sperimentazione degli algoritmi e delle strutture di dati. Ha visitato diversi centri di ricerca internazionali (dagli Stati Uniti al Giappone), pubblicando oltre 60 lavori scientifici e un brevetto.

E-mail: grossi@di.unipi.it

## Olimpici che hanno vinto una medaglia IOI



**Gilberto Abram**, del Liceo Scientifico di Cles – TN (argento, IOI 2003). Ho deciso di compiere i miei studi universitari all'estero e frequento i corsi di Electrical Engineering and Computer Science presso il Massachusetts Institute of Technology, Cambridge, MA, USA. Inoltre lavoro saltuariamente nel gruppo di ricerca di ottica ed elettronica quantistica del prof. Franz Kaertner, a partire dall'estate 2005. Ho iniziato a programmare quando frequentavo le scuole elementari, usando il linguaggio Basic per il Commodore 64. Il primo programma impegnativo che

ho scritto è stato per realizzare un motore grafico tri-dimensionale nel sistema operativo DOS che non ho portato a termine, cosa che ho fatto per un interprete del linguaggio Scheme nell'ambito del corso di programmazione. Visto che i miei voti alle superiori non erano bellissimi, a parte quelli in alcune materie, volevo dimostrare di essere in gamba almeno in qualcosa come programmare e le olimpiadi erano l'opportunità che cercavo. Poi a partire dalla fase regionale ho iniziato ad apprezzare gli algoritmi.



**Alessandro Arzilli**, dell'ITI Sarocchi di Siena (bronzo, IOI 2000). Sono appassionato di informatica e, per continuare a occuparmene, fingo di frequentare il Corso di Laurea in Informatica all'Università di Pisa. Ho iniziato a programmare a circa 14 anni usando il Pascal come linguaggio e, francamente, non ricordo il mio primo program-

ma impegnativo. Traggo soddisfazione dalla mia passione per l'informatica: decisi di partecipare alle olimpiadi perché la mia insegnante di informatica, notando tale passione, insistette che mi iscrivessi.



**Giorgio Audrito**, del Liceo Scientifico Curie di Pinerolo – TO (bronzo, IOI 2004 e 2005). Studio Matematica all'Università di Torino e, per il momento, ho potuto vivere di rendita negli esami di informatica per la preparazione ricevuta durante gli allenamenti di Pisa. Oltre all'università mi dedico alla musica, in particolare al pianoforte e alla composizione: l'anno passato ho scritto la colonna sonora per due film muti del museo del cinema di Torino. Quest'anno sto seguendo anche corsi di organo e di violino e, durante la scorsa dura estate, ho dovuto anche dare l'esame dell'8° anno di pianoforte subito dopo le olimpiadi! Ho iniziato a programmare quando frequentavo le scuole elementari, scrivendo in Pascal qualche semplice giochino con interfaccia testuale, ma ho smesso per poi iniziare nuovamente durante gli anni del liceo. Per le olimpiadi ho imparato a usare il linguaggio C, scrivendo codice di qualche "utilità" per me; in Pascal ho sviluppato giochini e programmi di grafica (per quanto si possa fare con 16 colori). Forse il primo (e unico?) programma impegnativo è una unit in Pascal contenente una serie di oggetti astratti per strutturare un programma a eventi. Ho partecipato alle olimpiadi perché mi sono sempre pia-

ciati i problemi logici in informatica e matematica: quando frequentavo il secondo anno del liceo e non potevo ancora gareggiare, mi sono fatto dare i testi della fase scolastica per provare a svolgerli!

**Luca Barbieri**, del Liceo Scientifico Carducci di Milano (oro, IOI 2004, e argento, IOI 2005). Frequento il primo anno del corso di Matematica presso la Scuola Normale Superiore di Pisa. Ho iniziato a programmare quando frequentavo la scuola elementare, usando il linguaggio Basic (ehm). Non ricordo esattamente quale fosse, ma uno dei primi programmi che avessero una qualche utilità svolgeva automaticamente le espressioni date per il compito di matematica, includendo tutti i passaggi intermedi (ma non dirò mai i nomi dei miei compagni di scuola che ne beneficiarono). Sono venuto a conoscenza delle olimpiadi internazionali scientifiche attraverso le Olimpiadi della Matematica, a cui la mia scuola aderiva. Incoraggiato dai successi in tale ambito e dalla mia passione preesistente per l'informatica, ho deciso di provare a sostenere anche le Olimpiadi di Informatica [ndr: Luca ha vinto delle medaglie anche in altre olimpiadi].



**Matteo Bruni**, dell'ITI Montani di Fermo - AP (bronzo, IOI 2003).

Frequento l'Università di Bologna, corso di Ingegneria Informatica, e sono al terzo anno. Ho iniziato a programmare verso i 10 anni, poi un po' più seriamente a 14 anni, adoperando il linguaggio GWBASIC e passando successivamente al Visual Basic e quindi al linguaggio C. Probabilmente il primo programma utile che ho scritto era quello per trovare il modo di riempire ottimamente il contenuto di un CD: avendo un insieme di file la cui dimensione complessiva superava la capacità del CD, dovevo prenderne un sotto-insieme per massimizzare lo spazio occupato nel CD. Quando mi sono messo a farlo mi sembrava una sciocchezza, però mi sono smentito subito. All'epoca non sapevo della complessità del problema, una variante del problema dello zaino (problema notoriamente NP-completo)!



**Dario Cazzaro**, dell'ITI Zuccante di Venezia (bronzo, IOI 2004). Sono iscritto al secondo anno di Ingegneria Informatica all'Università di Pisa e sono Allievo Ordinario presso la Scuola Superiore S. Anna di Pisa. Ho iniziato a programmare con il Commodore 64 verso i 10-11 anni con il Basic (mentre aspettavo il lunghissimo tempo di caricamento dei giochi) e poi ho continuato quest'attività iscrivendomi a un istituto tecnico. Grazie anche alla scuola, mi sono appassionato all'informatica scrivendo un po' di programmi (giochini, chat e così via) per



trascorrere le lunghe ore nei laboratori con i miei compagni. Il primo programma importante è stato, probabilmente, quello per la mia tesina di esame di stato: un gioco (a più giocatori e via rete) uguale a Risiko che permetteva di far giocare autonomamente anche i calcolatori. Partecipare alle olimpiadi era quasi una tradizione nella mia scuola e, chiaramente, partecipare alle nazionali una meta ambita. Che dire? Il gusto per la sfida era molto alto e, abbinato a quel tipo di informatica per nulla "meccanica" ma che ti costringe a ragionare, mi ha invogliato a impegnarmi seriamente per raggiungere dei buoni risultati a livello internazionale.

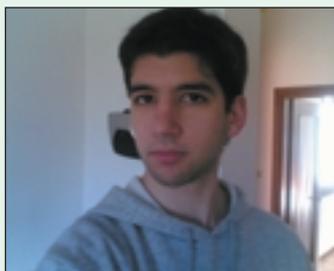


**Alessio Guerrieri**, del Liceo Scientifico da Vinci di Trento (argento, IOI 2005). Studio Informatica presso l'Università di Trento. I miei primi linguaggi di programmazione sono stati ToolBook (se vale, è un sistema di e-learning), il Pascal (usato per una settimana) e Visual Basic (adoperato tanto): il primo programma impegnativo è stato il videogioco di Dragonball, scritto appunto in Visual Basic. Ho deciso di partecipare alle olimpiadi inizialmente per saltare scuola (ehm), poi per interesse (giuro!).



**Alessandro Maconi**, dell'ITC Greppi di Monticello Brianza – LC (bronzo, IOI 2001). Studio Ingegneria Elettronica presso il Politecnico di Milano e attualmente frequento il primo anno della laurea specialistica di tale corso. Ho iniziato a programmare a dodici anni, nel tentativo di imitare mio cugino più grande che programava in Basic, che è quindi diventato il mio primo linguaggio. Ho sviluppato il primo program-

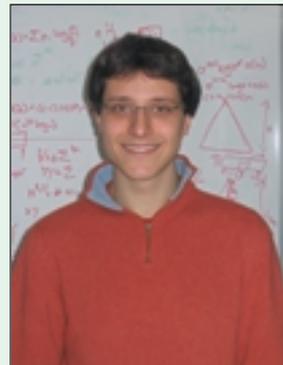
ma impegnativo a quattordici anni: un tentativo di realizzare una specie di derive molto rozzo per disegnare una funzione. Potevo scegliere fra tre tipi di funzione: retta, parabola, iperbole equilatera. Effettuata tale scelta, dovevo indicare il valore dei parametri, l'intervallo di valutazione della funzione e il colore con cui volevo disegnarla: il programma disegnava gli assi cartesiani e tracciava la funzione scelta. Ho partecipato alle olimpiadi perché avevo visto un manifesto, chiedendo così al mio professore di informatica di far partecipare la scuola. Il mio sogno era di riuscire ad arrivare alla fase mondiale in Finlandia (nel 2001 ero in quarta) o in Corea (nel 2002 sarei stato in quinta).



**Stefano Maggiolo**, dell'ITI Severi di Padova (bronzo, IOI 2002). Studio Matematica all'Università di Ferrara e ho iniziato a programmare a 9-10 anni, credo, usando il linguaggio qbasic. Il primo programma impegnativo (o almeno, il primo di cui non vergognarsi) è stato una specie di assembler. Perché ho partecipato alle olimpiadi... se permettete, che domanda è? E perché

no? Per accrescere il mio ego – anzi, perché ci obbligò il professore di informatica (fino alla fase regionale non sapevo che esistessero anche le gare internazionali): va bene, lo confesso, voglia di agonismo (bla bla bla), possibilità di andare in giro per l'Italia e nel mondo (bla bla bla). Insomma, fatelo anche voi!

**Giuseppe Ottaviano**, del Liceo Scientifico Fermi di Ragusa (argento, IOI 2003). Frequento il terzo anno del corso di laurea in Matematica dell'Università di Pisa e collaboro con un'azienda straniera specializzata nei motori di ricerca. Ho iniziato a programmare a 13-14 anni utilizzando il linguaggio Visual Basic (ehm), ma presto mi sono spostato sui linguaggi C e C++. Non ho mai scritto da zero un programma veramente impegnativo (a parte le soluzioni per gli esercizi delle olimpiadi), al massimo ho scritto qualche patch per programmi di altri. Il programma che mi ha richiesto più tempo è stato, recentemente, un programma giocattolo per simulare un sistema di molle attraverso la risoluzione di equazioni differenziali con delle tecniche viste a lezione. Ho deciso di partecipare alle olimpiadi perché mi ha sempre affascinato l'informatica teorica e la realizzazione in codice degli algoritmi. Ho cominciato per gioco e non mi aspettavo di arrivare neanche alle nazionali: volevo soltanto misurarmi con gli altri ragazzi che condividevano i miei stessi interessi per vedere se tutto il tempo che avevo perso a programmare era servito a qualcosa!



**Alessandro Piva**, del Liceo Scientifico Majorana di Latina (bronzo, IOI 2004). Studio Informatica all'Università "La Sapienza" di Roma e, una volta presa la laurea triennale, spero di poter andare a studiare a Pisa. Ho iniziato a programmare quando frequentavo le scuole elementari, utilizzando un Commodore 64 e il Basic: era molto interessante, anche se ci metteva 10 minuti a caricare un programma da una cassetta! Poi è venuta l'era del linguaggio Pascal, e così via: il mio primo programma impegnativo è stato probabilmente un gioco che ho realizzato (o almeno, ho iniziato) quando frequentavo le scuole medie. Raffigurava una pallina con occhi e bocca (molto facile da disegnare) che si muoveva in un piccolo bosco. La visuale era isometrica ed era scritto completamente in Pascal. Molto più ambizioso, sempre del medesimo periodo, un programma per visualizzare forme in tre dimensioni... sempre in Pascal! Le immagini che produceva erano a dir poco grottesche, con cubi che sembravano piramidi, ma erano pur sempre tridimensionali! Ho deciso di partecipare alle olimpiadi per le incredibili opportunità che mi offrivano: conoscere ragazzi appassionati come me, fare dei corsi intensivi con gli allenatori e, soprattutto, fare tanti bei viaggi! La cosa più importante che le olimpiadi mi hanno lasciato, comunque, è ben altra: è cambiato completamente il mio atteggiamento verso i progetti e gli impegni che mi assumo. Ho visto che posso ottenere buoni risultati se mi impegno a fondo, quindi adesso affronto le cose con più autostima e con meno timore. Senza contare, poi, che sono venuto a contatto con persone estremamente capaci e simpatiche, con cui continuo a tenermi in contatto!



## Esempi di esercizi dati alle selezioni scolastiche

*Logico-matematico:* Il direttore di un ristorante con capienza massima di 150 posti non ricorda quante erano le persone da lui servite in occasione dello scorso cenone di fine anno. Ricorda però che volendo sistemare tutte le persone servite in tavoli da 3 ne restava fuori esattamente una; inoltre, la stessa cosa succedeva sistemando tutte le persone in tavoli da 5 o tutte in tavoli da 7. Quante erano le persone servite in occasione dello scorso cenone di fine anno?

*Schema di risoluzione:* Erano  $n = 106$  persone poiché  $n \leq 150$  e  $n - 1$  deve essere divisibile per 3, 5 e 7; essendo quest'ultimi dei numeri primi, l'unico multiplo che soddisfa le condizioni è  $n - 1 = 3 \cdot 5 \cdot 7 = 105$ .

*Programmatico (Pascal):* Data la seguente funzione ricorsiva

```
function mistero(m : integer; n : integer) : integer;
```

```
begin
```

```
if ( m = 0 ) then
```

```
  mistero := n
```

```
else if ( n=0 ) then
```

```
  mistero := mistero( m-1, 1 )
```

```
else
```

```
  mistero := mistero( mistero( m-1, n-1 ), n-1 )
```

```
end;
```

calcolare quale tra le seguenti risposte corrisponde ai valori restituiti invocando

```
writeln(mistero(0,3), ' ', mistero(1,3), ' ', mistero(2,3), ' ', mistero(3,3));
```

*Programmatico (C/C++):* Data la seguente funzione ricorsiva

```
int mistero(int m, int n)
```

```
{if ( m == 0 )
```

```
  return n;
```

```
else if ( n==0 )
```

```
  return mistero( m-1, 1 );
```

```
else
```

```
  return mistero( mistero( m-1, n-1 ), n-1 );}
```

calcolare quale tra le seguenti risposte corrisponde ai valori restituiti invocando

```
printf("%d %d %d %d\n", mistero(0,3), mistero(1,3), mistero(2,3), mistero(3,3));
```

a. 3 1 2 0

b. 3 2 1 0

c. 0 1 2 3

d. Nessuna delle risposte precedenti

*Schema di risoluzione:* Mantenendo traccia delle chiamate ricorsive attraverso un albero delle chiamate ricorsive e dei corrispondenti parametri attuali, risulta che la risposta corretta è la "b".

## Esempi di esercizi dati alle selezioni regionali

**Descrizione (livello di difficoltà  $D=2$ ):** Il nobile chimico Alfredo produce nel suo laboratorio due sostanze liquide potenzialmente inquinanti: l'Aminozalina e il Brinofulo. A fine giornata le deve smaltire in appositi contenitori, dislocati lungo il tragitto che parte dal laboratorio e arriva alla sua abitazione. Per limitare le possibilità d'inquinamento, Alfredo deve distribuire l'Aminozalina nel maggior numero possibile di contenitori mentre deve dividere il Brinofulo nel minor numero possibile di contenitori. Tuttavia Aminozalina e Brinofulo non possono essere assolutamente mescolati nel medesimo contenitore, altrimenti la loro miscela esplose. Ogni volta che raggiunge un contenitore per lo smaltimento dei liquidi, Alfredo deve eseguire una sola delle tre seguenti azioni: (i) versare Aminozalina fino al riempimento del contenitore; (ii) versare Brinofulo fino al riempimento del contenitore; (iii) non versare nulla nel contenitore. Data la quantità  $A$  di litri di Aminozalina e la quantità  $B$  di litri di Brinofulo da smaltire, e conoscendo l'elenco degli  $N$  contenitori (con rispettiva capacità) nell'ordine secondo cui sono incontrati lungo il tragitto dal laboratorio alla sua abitazione, aiutiamo Alfredo a decidere se e quale sostanza versare in ciascun contenitore.

**Dati di input:** Il file `input.txt` contiene nella prima riga gli interi  $A$  e  $B$  (rispettivamente i litri di Aminozalina e di Brinofulo da smaltire) e il numero  $N$  di contenitori disponibili. Tali valori sono separati da uno spazio. Nelle successive  $N$  righe (usando una riga per ogni contenitore) è contenuto un numero per riga: tali numeri rappresentano le capacità dei singoli contenitori elencati nell'ordine in cui vengono incontrati da Alfredo.

**Dati di output:** Il file `output.txt` deve contenere  $N$  righe, una per ogni contenitore. Ogni riga contiene due numeri separati da uno spazio, rispettivamente il numero di litri di Aminozalina e di Brinofulo smaltiti nel corrispondente contenitore. Notiamo che ogni riga deve contenere uno zero nei casi (i) e (ii) descritti sopra, e due zeri nel caso (iii).

**Schema di risoluzione:** Ordiniamo prima tutti i contenitori in base alla loro capacità, in ordine crescente (prima i contenitori meno capaci e poi quelli più capaci). Esaminando la sequenza in tal modo ordinata di contenitori, decidiamo di riempire i primi (quelli meno capaci) con l'Aminozalina fino a smaltirne  $A$  litri, così da usare il maggior numero possibile di contenitori. Per il Brinofulo, decidiamo di riempire partendo dagli ultimi contenitori fino a smaltirne  $B$  litri, così da usare il minor numero possibile di contenitori. Tale approccio goloso (*greedy*) funziona per le assunzioni del problema non riportate per brevità.

## Esempi di esercizi dati alle selezioni nazionali

**Descrizione (livello di difficoltà  $D=3$  e tempo limite di 10 secondi):** A Camelot, nel periodo di maggior splendore, ogni amicizia è corrisposta. Ogni cavaliere è amico di oltre la metà dei suoi compagni d'arme, ossia, se  $N$  è il numero dei cavalieri, ciascuno di essi può contare su almeno  $N/2$  validi amici. La tavola rotonda è disposta in un enorme salone, circondata da  $N$  sedie, con una sedia per cavaliere. Com'è facilmente immaginabile, ogni cavaliere pretende di avere amici ai suoi due lati, altrimenti rifiuta di sedersi al tavolo. Tutti contano su Mago Merlino per disporre i cavalieri intorno al tavolo nella soddisfazione generale di tutti. Ma il poverino ha perso la formula magica per ottenere ciò in men che non si dica! Aiutiamo Merlino, anche con soluzioni parziali, a disporre i cavalieri intorno al tavolo in modo che ogni cavaliere abbia degli amici ai suoi lati.

**Dati di input:** Il file `input.txt` contiene nella prima riga l'intero  $N$ , il numero dei cavalieri della tavola rotonda. I cavalieri sono numerati da 1 a  $N$ . Ognuna delle successive  $N$  righe contiene una sequenza di  $N$  valori o oppure 1, separati da uno spazio. La sequenza contenuta nell' $i$ -esima di tali righe rappresenta le relazioni di amicizia del cavaliere numero  $i$ . In particolare, il  $j$ -esimo valore in tale riga indica se i cavalieri  $i$  e  $j$  sono amici (valore = 1) o meno (valore = 0). Poiché l'amicizia è sempre corrisposta, se il cavaliere  $i$  è amico del cavaliere  $j$  allora il cavaliere  $j$  è amico del cavaliere  $i$ .

**Dati di output:** Il file `output.txt` deve contenere una sola riga contenente una sequenza di  $N$  numeri separati da uno spazio, per rappresentare la disposizione dei cavalieri attorno alla tavola rotonda. Il primo numero è sempre 1, a rappresentare il cavaliere numero 1. Il resto della sequenza è una permutazione dei numeri 2, 3, ...,  $N$  (cioè una disposizione in un qualche ordine, senza ripetizioni). Per ogni coppia  $i$  e  $j$  di numeri adiacenti nell'intera sequenza, i cavalieri  $i$  e  $j$  devono essere amici. Inoltre, il cavaliere alla fine della sequenza deve essere amico del cavaliere numero 1, in quanto vicini di posto.

**Schema di risoluzione:** Rappresentiamo il problema come un grafo  $G$  i cui nodi sono i cavalieri e i cui archi collegano due nodi se i corrispondenti cavalieri sono amici. Il problema di disporre i cavalieri attorno alla tavola rotonda equivale a trovare un ciclo hamiltoniano nel grafo  $G$ , ovvero un cammino che attraversa tutti i nodi una e una sola volta e che si ricongiunge al nodo di partenza. In generale, non conosciamo un algoritmo efficiente per risolvere tale problema (tecnicamente, la sua versione decisionale è *NP-completa*): fortunatamente possiamo ottenerlo nel nostro caso sfruttando l'importante ipotesi del problema, quella per cui ciascun nodo in  $G$  ha almeno  $N/2$  nodi adiacenti (tecnicamente, il grado è almeno  $N/2$ ). A tale scopo, troviamo un qualunque cammino massimale  $P$  che non passa due volte attraverso uno stesso nodo e indichiamo i suoi due nodi estremi con  $x$  e  $y$  (tale cammino attraversa un numero di archi che è pari almeno al grado dei suoi nodi, ovvero almeno  $N/2$ ). Per l'ipotesi sul grado dei nodi, possiamo sempre trovare due archi che collegano  $x$  e  $y$  a due nodi adiacenti (siano essi  $w$  e  $z$ ) all'interno del cammino  $P$ : tali archi  $(x, z)$  e  $(w, y)$ , incrociandosi, creano un ciclo denominato  $C$  ottenuto eliminando l'arco  $(w, z)$  da  $P$  e aggiungendo  $(x, z)$  e  $(w, y)$ . Notiamo che deve esistere un nodo  $u$  in  $C$  che è collegato a un nodo non appartenente a  $C$  (quest'ultimo è infatti più lungo di  $N/2$  e tutti i nodi hanno grado almeno  $N/2$ ). Rompiamo il ciclo  $C$  in corrispondenza di  $u$  per ottenere un cammino che sicuramente estendiamo a partire da  $u$ : ripetiamo più volte i passi precedenti di estensione del cammino/ciclo corrente fino a toccare tutti i vertici una e una sola volta (cioè il ciclo diventa hamiltoniano). Da notare che tale soluzione termina velocemente perché aggiunge almeno un nodo al cammino corrente per ogni iterazione, ma non si applica al caso generale perché non è detto che valga l'ipotesi sul grado dei nodi.

## Esempi di esercizi dati alle gare internazionali

**Descrizione del primo problema (tempo limite di 0,5 secondi):** Byteman ha il più bel giardino in Bytetown. Nel giardino ha piantato  $n$  rose. È arrivata l'estate e i fiori sono cresciuti grandi e belli. Byteman si è reso conto di non poter curare tutte le rose da solo e ha deciso di assumere due giardinieri per aiutarlo. Byteman vuole selezionare due aree rettangolari in modo che ciascun giardiniere possa curare le rose all'interno di ciascun rettangolo. I rettangoli devono essere disgiunti e contenere esattamente  $k$  rose ciascuno. Byteman intende recintare il perimetro dei due rettangoli ma, essendo a corto di denaro, vuole impiegare meno recinto possibile. Aiutate Byteman a individuare i due rettangoli: il giardino di Byteman, un rettangolo lungo  $l$  metri e largo  $w$  metri, è diviso in  $l \cdot w$  quadratini di lato un metro. Fissate un sistema di coordinate con gli assi paralleli ai lati del giardino. Ogni quadratino è individuato da una coppia di interi  $(x, y)$  con  $1 \leq x \leq l$  e  $1 \leq y \leq w$ . I quadratini possono contenere anche più di una rosa. Selezionate i due rettangoli con lati paralleli a quelli del giardino e con vertici a coordinate intere, notando che un rettangolo con angoli nei quadratini  $(l_1; w_1)$ ,  $(l_1; w_2)$ ,  $(l_2; w_1)$  e  $(l_2; w_2)$  contiene tutti i quadratini  $(x, y)$  con  $l_1 \leq x \leq l_2$  e  $w_1 \leq y \leq w_2$  e ha perimetro di lunghezza  $2 \cdot (l_2 - l_1 + 1) + 2 \cdot (w_2 - w_1 + 1)$ . I due rettangoli debbono essere disgiunti, ossia non possono avere quadratini in comune. Qualora abbiano parti di perimetro in comune, essi dovranno avere comunque recinzioni separate. Scrivete un programma che legge da standard input le dimensioni del giardino, il numero delle rose nel giardino, il numero delle rose da riporre in ciascuno dei due rettangoli, e le posizioni delle rose; trova gli angoli di due rettangoli la cui somma dei perimetri è minima, sotto le condizioni di cui sopra; scrive su standard output il minimo valore per la somma dei perimetri di due rettangoli disgiunti contenenti ciascuno precisamente  $k$  rose, (oppure la singola parola NO, ove una tale coppia di rettangoli non esista).

**Dati di input:** La prima riga del file input.txt contiene due interi separati da un singolo spazio, ossia la lunghezza  $l$  e la larghezza  $w$  del giardino ( $1 \leq l, w \leq 250$ ). La seconda riga contiene due interi separati da un singolo spazio: il numero  $n$  di rose nel giardino e il numero  $k$  di rose che dovrebbe essere racchiuso in ciascun rettangolo ( $2 \leq n \leq 5000$ ,  $1 \leq k \leq n/2$ ). Le  $n$  righe seguenti contengono le coordinate delle rose, una rosa per ogni riga: la riga  $(i + 2)$  riporta due interi  $l_i$  e  $w_i$  ( $1 \leq l_i \leq l$ ,  $1 \leq w_i \leq w$ ), separati da un singolo spazio, ovvero le coordinate del quadratino contenente la rosa  $i$ -esima. Due o più rose possono riproporsi nello stesso quadratino.

**Dati di output:** Il file output.txt deve contenere un'unica riga, contenente solamente un intero che rappresenta il valore minimo per la somma dei perimetri di due rettangoli disgiunti contenenti ciascuno precisamente  $k$  rose, oppure una singola parola NO, nel caso una tale coppia di rettangoli non esista.

**Schema di risoluzione:** Ipotizziamo che sia  $w = l$  per semplicità, ottenendo così un giardino quadrato di lato  $w$ . Indichiamo con il termine  $k$ -rettangolo un rettangolo interno al giardino che contiene  $k$  rose. Essendoci all'incirca  $w^4$  possibili rettangoli da considerare nel giardino, la soluzione che seleziona i  $k$ -rettangoli tra questi ultimi e poi li esamina a coppie per trovarne una con perimetro minimo, riesce a risolvere solo un piccolo numero di casi nel tempo limite assegnato al problema (in quanto il numero di passi computazionali è molto elevato, proporzionale a circa  $w^8$  operazioni di calcolo). Bisogna quindi procedere in altro modo, evitando di calcolare esplicitamente tali valori. Forniamo due idee che aiutano a risolvere diversi casi (laddove occorre raffinarle, per risolverli tutti, con ulteriori tecniche algoritmiche, pervenendo così a una soluzione che richiede all'incirca  $w^3$  operazioni di calcolo). La prima idea permette di calcolare il numero di rose di un qualunque rettangolo in un numero costante di operazioni, indipendentemente dalla dimensione del rettangolo, dopo un'opportuna pre-elaborazione che richiede circa  $w^2$  operazioni. Quest'ultima considera in modo incrementale soltanto i rettangoli "prefissi"  $P(x, y)$  tali che hanno un angolo nel quadratino  $(1, 1)$  e quello opposto in  $(x, y)$ : dato un qualunque rettangolo con angoli nei quadratini  $(l_1; w_1)$ ,  $(l_1; w_2)$ ,  $(l_2; w_1)$  e  $(l_2; w_2)$ , il numero di rose in esso contenuto può essere sempre espresso come la somma del numero di rose in  $P(l_1 - 1, w_1 - 1)$  e  $P(l_2, w_2)$  a cui va sottratto il numero di rose in  $P(l_1 - 1, w_2)$  e  $P(l_2, w_1 - 1)$ . La seconda idea è che non è necessario considerare tutte le coppie di  $k$ -rettangoli, ma solo quelle che non ne contengono ulteriori al loro interno: presa una coppia di tali  $k$ -rettangoli disgiunti, quest'ultimi devono essere separabili da una linea orizzontale o verticale (altrimenti non sarebbero disgiunti). Possiamo quindi considerare  $w$  tagli orizzontali (e altrettanti tagli verticali) nel giardino, concentrandoci sul sotto-problema di trovare il  $k$ -rettangolo di perimetro minimo in ciascuna parte del taglio corrente. Possiamo affrontare quest'ultimo sotto-problema utilizzando uno schema basato su una "finestra scorrevole" di altezza prefissata che permette di identificare i  $k$ -rettangoli di tale altezza molto velocemente. Maggiori dettagli sono riportati nel sito [www.ioi2005.pl/downloads/tasks.php](http://www.ioi2005.pl/downloads/tasks.php).

**Descrizione del secondo problema (tempo limite di 5 secondi):** Consideriamo una sequenza non-decrescente di interi  $s_1, \dots, s_{n+1}$  (dove  $s_i \leq s_{i+1}$  per  $1 \leq i \leq n$ ). La sequenza  $m_1, \dots, m_n$  definita da  $m_i = \frac{1}{2}(s_i + s_{i+1})$  per  $1 \leq i \leq n$ , è chiamata la *sequenza delle medie* di  $s_1, \dots, s_{n+1}$ . Per esempio, la sequenza di medie per 1, 2, 2, 4 è pari a 1,5, 2, 3. Notiamo che gli elementi di una sequenza di medie possono essere frazionari. Tuttavia, il problema proposto tratta esclusivamente sequenze di medie i cui elementi sono tutti interi. Data una sequenza non-decrescente di  $n$  interi  $m_1, \dots, m_n$  il problema richiede di calcolare il numero di sequenze non-decrescenti costituite da  $n + 1$  interi  $s_1, \dots, s_{n+1}$  aventi  $m_1, \dots, m_n$  come sequenza delle medie. Scrivete un programma che legge da standard input una sequenza non-decrescente di interi e calcola il numero di sequenze non-decrescenti la cui sequenza delle medie corrisponda alla sequenza data.

**Dati di input:** La prima riga del file input.txt contiene un intero  $n$  ( $2 \leq n \leq 500000$ ). Le seguenti  $n$  righe contengono la sequenza  $m_1, \dots, m_n$ . Il valore di  $m_i$  ( $0 \leq m_i \leq 100000000$ ) è fornito nella riga numero  $i + 1$ .

**Dati di output:** Il file output.txt deve contenere esattamente un intero che rappresenta il numero di sequenze non-decrescenti di interi che hanno la sequenza di input come sequenza di medie.

*Descrizione del terzo problema (tempo limite di 3 secondi):* Il Mountain Amusement Park ha aperto un nuovissimo simulatore di montagne russe. Il percorso consiste di  $n$  spezzoni collegati agli estremi, a partire dal primo spezzone fissato a quota 0. L'operatore Byteman può riconfigurare il percorso a piacere aggiustando le inclinazioni di spezzoni consecutivi. L'inclinazione dei rimanenti spezzoni non viene alterata. Ogni volta che egli agisce su uno spezzone, lo spezzone successivo viene sollevato o abbassato al fine di mantenere la continuità della rotaia, fermo restando che il primo spezzone resta fisso a quota 0. Ciascuna corsa inizia con il lancio del carrello a cui è impressa una certa energia, sufficiente al raggiungimento di una data quota  $h$ . Il carrello continua la sua corsa fino a quando non supera quota  $h$  oppure raggiunge la fine del percorso. Data la sequenza di tutte le corse e di tutte le operazioni di riconfigurazione avvenute nella giornata, calcolate, per ciascuna corsa, il numero di spezzoni attraversati dal carrello nella sua corsa. Internamente, il simulatore rappresenta le inclinazioni lungo il percorso come una sequenza di  $n$  dislivelli, uno per ogni spezzone. L' $i$ -esimo numero di rappresenta, in centimetri, il dislivello in corrispondenza dello spezzone  $i$ -esimo. Ipotizzando che il carrello sia a quota  $h$  dopo aver attraversato i primi  $i - 1$  spezzoni, esso raggiungerà quota  $h + d_i$  dopo aver attraversato anche lo spezzone  $i$ -esimo. Inizialmente gli spezzoni sono tutti orizzontali, ovvero  $d_i$  è pari a 0 per tutti gli  $i$ . Le corse e le operazioni di riconfigurazione si alternano arbitrariamente durante la giornata. Ciascuna riconfigurazione è specificata da tre numeri:  $a$ ,  $b$  e  $D$ . Gli spezzoni coinvolti sono quelli del segmento da  $a$  a  $b$  (inclusi gli estremi). Il dislivello per ciascun spezzone del segmento viene posto uguale a  $D$ . In altre parole,  $d_i = D$ ,  $a \leq i \leq b$ . Ciascuna corsa è specificata da un numero  $h$  che rappresenta la massima quota raggiungibile dal carrello. Scrivete un programma che legge da standard input una sequenza di operazioni di riconfigurazione e di corse e, per ciascuna corsa, calcola il numero di spezzoni attraversati dal carrello.

*Dati di input:* La prima riga del file `input.txt` contiene un intero positivo  $n$  che rappresenta il numero di spezzoni,  $1 \leq n \leq 1\,000\,000\,000$ . Le righe che seguono contengono le operazioni di riconfigurazione e le corse. Segue un terminatore. Ciascuna riga successiva alla prima è una delle seguenti:

- Riconfigurazione — una sola lettera 'I', e gli interi  $a$ ,  $b$  e  $D$ , separati da spazi singoli (dove  $1 \leq a \leq b \leq n$  e  $-10\,000\,000\,000 \leq D \leq 10\,000\,000\,000$ ).
- Corsa — una sola lettera 'Q', e un intero  $h$  ( $0 \leq h \leq 10\,000\,000\,000$ ) separati da uno spazio;
- Una sola lettera 'E' — il terminatore, a indicare la fine dell'input.

Possiamo ipotizzare che, in un qualsiasi momento, la quota raggiunta da un qualsiasi punto del percorso sia nell'intervallo  $[0, 10\,000\,000\,000]$  in centimetri. L'input non contiene più di 100000 righe.

*Dati di output:* Il file `output.txt` deve contenere, nella riga  $i$ -esima, un solo intero che rappresenta il numero di spezzoni attraversati dal carrello durante l' $i$ -esima corsa della giornata.