

Ghid complet pentru concursurile de informatica

(Autor *Mircea Pasoi* <http://www.infoarena.ro/ghid-complet-pentru-concursurile-de-informatica>)

Acest articol se adreseaza pasionatilor de informatica si celor care au de gand sa participe la concursurile si olimpiadele de informatica. Observatiile din cadrul acestui articol sunt, in mare parte, rezultatul experientei autorului.

1. Ce sunt concursurile?

Concursurile de informatica, ca la orice alta disciplina, vor sa fie o metoda de clasificare a participantilor in functie de abilitatile de programare si de cunostintele de algoritmica. In plus, exista unele concursuri in care cei mai buni concurenti sunt rasplatiti. Probabil cel mai important aspect, concursul include aparitia unui factor suplimentar care rastoarna multe din obisnuitele programarii "la domiciliu" sau "la locul de munca" (de obicei programatorii care lucreaza in firme au perioade mult mai lungi de timp pentru a-si duce la final sarcina): **timpul**. Autorul a avut la dispozitie mai mult de sapte ani ca sa descopere pe propria piele importanta foarte mare a acestui factor; si, mai mult decat durata de timp in sine a concursului, care este aceeaasi pentru toti concurentii, conteaza capacitatea fiecaruia de a gestiona bine acest timp. Daca in fata calculatorului de acasa, cu o sticla de suc alaturi si cu muzica mergand, este intr-adevar un lucru laudabil sa justificam matematic fiecare pas al algoritmului, sa nu ne lasam inselati de intuitie si sa scriem programul fara sa ne grabim, alocandu-ne o mare parte din timp numai pentru depanarea lui, in schimb, in timp de concurs lucrurile stau tocmai invers: de demonstratii riguroase din punct de vedere matematic rar are timp cineva, intuitia fiind la mare pret si de nenumarate ori fiind criteriul care aduce victoria; iar timpul la un concurs serios este suficient doar pentru implementarea tuturor programelor, pentru depanare alocandu-se de obicei o perioada destul de mica.

In multe cazuri, cele doua etape principale ale programarii - conceperea si implementare algoritmului - incep sa se bata cap in cap. Exista situatii in care avem la dispozitie un algoritm foarte eficient, dar implementarea acestuia este extrem de dificila, alteori algoritmul ales nu va face fata volumului maxim de date de intrare, iar alteori ne dam seama ca am putea foarte usor sa scriem un program, dar nu suntem in stare sa demonstram ca el merge sigur pe orice set de date de intrare. Foarte des se renunta la implementarea algoritmilor de complexitate optima, care in multe cazuri constituie adevarate focare de "bug"-uri, preferandu-se un algoritm mai lent dar care sa se poate implementa mai rapid si fara dureri de cap. Multi concurenti pierd primii ani de concursuri descoperind aceste lucruri.

Desigur, aceste lucruri variaza foarte mult in functie de viteza fiecaruia de a implementa in limbajul preferat (atentie, viteza mare de tastare a unui text normal nu garanteaza neaparat o viteza mare de tastare a unui program!), cat si de abilitatea de a te concentra la toate detaliile care apar in implementare (acest lucru depinde foarte mult de cat de antrenata este mintea fiecaruia). De fapt, concurentii care ies pe primele locuri de obicei, sunt cei care au reusit sa formeze un echilibru intre cele doua etape ale programarii: pot sa conceapa algoritmi eficienti, care sa se incadreze in limitele de timp si memorie, cat si sa implementeze acesti algoritmi intr-un timp acceptabil, depanand foarte putin.

O intrebare pe care cineva, care citeste aceste randuri, si-o poate pune este urmatoarea: *De ce sa particip la astfel de concursuri?* Principalele avantaje sunt formarea unei gandiri algoritmice, intelegerea metodelor de rezolvare pentru anumite probleme (care pot aparea si in cadrul dezvoltarii de aplicatii), dezvoltarea capacitatii de a reactiona rapid intr-un timp scurt. In timp, experienta acumulata in cadrul pregatirii si a concursurilor va conduce la eficienta mai mare in cadrul proiectelor de dezvoltare de software si, de ce nu, la obtinerea unor salarii mai mari la angajare. Exista multe companii care iau in considerare participarile la concursuri la interviurile pentru angajare, cat si multe facultati din strainatate care le considera un criteriu pentru primirea unei burse (asta nu inseamna ca o astfel de participare garanteaza neaparat o bursa!). Un alt avantaj ce nu este de neglijat este reprezentat de premiile puse in joc, atat cele materiale (calculatoare, diferite componente hardware, excursii in strainatate, carti, etc.), cat si intrarea fara examen la facultate, in cazut obtinerii unor rezultate bune la olimpiadele nationale sau internationale.

2. Care sunt concursurile?

O alta intrebare la care doresc sa aflu raspunsul cei care vor sa participe la concursurile de informatica este: *Care sunt cele mai importante concursuri de programare?* Se pot clasifica doua mari tipuri de concursuri: cele pentru gimnaziu si liceeni si cele pentru studenti.

Concursuri pentru liceu

- **Olimpiada Locala de Informatica** - Se desfasoara in fiecare judet de obicei la inceputul anului (lunile ianuarie-februarie) si, in general, nivelul de dificultate este foarte redus (exceptie facand poate Bucuresti si alte cateva judete), subiectele fiind propuse de profesorii din judetul respectiv.
- **Olimpiada Judeteana de Informatica** - Incepand cu anul 2003 subiectele pentru olimpiada judeteana au fost aceleasi in toata tara; de

obicei se desfasoara cu o luna dupa olimpiada locala si cu cel putin o luna inainte de olimpiada nationala; subiectele sunt de un nivel mediu in general.

- **Olimpiada Nationala de Informatica** - Se organizeaza o data pe an, de obicei in prima vacanta din semestrul II al anului scolar; probele pentru liceu se desfasoara timp de 2 zile, in fiecare zi concurentii avand de rezolvat 3 probleme in 4 ore, iar probele pentru gimnaziu dureaza doar o zi, cate 2 probleme in 3 ore. De asemenea, prima jumătate din clasamentul pentru fiecare clasa de liceu participa la barajele pentru selectia lotului largit de informatica - aici, toti concurentii, indiferent de varsta, au de rezolvat aceleasi probleme. Primii ~20 selectati in urma acestei probe (care se desfasoara tot timp de 2 zile) vor forma lotul national largit; acestia vor participa la mai multe pregatiri, in cadrul carora sunt incluse baraje pentru selectarea echipelor de cate 4 elevi care vor reprezenta Romania la BOI, CEOI si IOI.
- **Balcaniada de Informatica (BOI)** - Prima editie a avut loc la Constanta in anul 1993. Standardele acestui concurs nu erau foarte ridicate, iar concurentii romani se claseaza foarte des pe primele locuri. Cel mai concludent exemplu in acest sens este faptul ca la editia din anul 2001 toate cele patru medalii de aur au fost obtinute de reprezentantii nostri, iar in 2005 ambele medalii de aur acordate au revenit iarasi romanilor. Exista si ani cand nivelul de dificultate al problemelor este mai ridicat, spre exemplu in 2003(Romania) si 2004(Bulgaria).
- **Olimpiada de Informatica a Europei Centrale (CEOI)** - Prima editie a avut loc la Cluj-Napoca in anul 1994. Editia din 2000 a acestui concurs a avut loc tot la Cluj-Napoca si foarte probabil editia din 2009 va fi tot in Romania. Standardele acestui concurs sunt mai ridicate decat cele de la BOI, deoarece elevii din tarile participante sunt intotdeauna foarte bine pregatiti.
- **Olimpiada Internationala de Informatica (IOI)** - Acest concurs reuneste anual elevi din ~80 tari ale lumii. Fiecare tara este reprezentata de cel mult patru concurenti, iar numarul tarilor participante creste in fiecare an. Organizarea concursului este asemanatoare cu cea a celorlalte concursuri internationale mentionate. Concursul este individual si se desfasoara sub forma a doua probe. La fiecare proba concurentii au la dispozitie 5 ore pentru a rezolva 3 probleme cu un grad ridicat de dificultate. Dupa fiecare proba,

programele concurentilor sunt evaluate automat, cu ajutorul unor programe de evaluare. Dupa cele doua zile de concurs se stabileste clasamentul final si se acorda medaliile. Rezultatele sunt secrete pana in momentul decernarii premiilor. La aproape toate editiile acestui concurs, echipa Romaniei a avut rezultate excelente. Astfel, incepand din anul 1993, cu doar doua exceptii, toti cei patru componentii ai echipei tarii noastre au obtinut medalii; conform punctajelor individuale, s-au obtinut doua locuri I (1993 si 1998) cu punctaj maxim si un loc II (2001), iar in clasamentul pe natiuni Romania este o prezenta constanta in primele 10 locuri.

Mai exista si alte concursuri internationale, cum ar fi **Olimpiada Tarilor Baltice**, la care Romania nu participa. Puteti gasi detalii mai multe despre aceste concursuri, cat si problemele propuse, folosind [Google](#). De asemenea problemele si solutiile din ultimii ani pentru concursurile importante pot fi gasite la sectiunea [Downloads](#) a site-ului [infoarena](#), acesta fiind actualizat periodic.

Exista cateva concursuri regionale si nationale similare olimpiadelor (diferenta timpul de concurs, numarul de probe si dificultatea problemelor). Cateva dintre acestea sunt:

- **Marele Premiu al Palatului Copiilor** - concurs organizat la Palatul National al Copiilor din Bucuresti; participa echipe ale Cluburilor Copiilor din mai multe judete ale tarii
- **Concursul "Grigore C. Moisil"** - organizat anual la Lugoj; are o desfasurare similara
- diferite concursuri interjudetene organizate in anumite regiuni ale tarii; mai multe astfel de concursuri poarta numele lui Grigore Moisil; alte concursuri sunt **Info-Oltenia**, **LInfo@SV**, **Urmarii lui Moisil**, etc.
- Stelele Informaticii - concurs cu organizare asemanatoare cu Olimpiada Nationala de Informatica; participarea la acesta se face doar pe baza de invitatie.

De obicei aceste manifestari sunt mai ample; la aceste concursuri exista si alte sectiuni, cum ar fi cele dedicate dezvoltarii de aplicatii software sau de pagini web.

Exista de asemenea numeroase concursuri on-line, la care concurentii participa de acasa si isi trimit solutiile prin intermediul internetului. Desigur, nu se pot acoperi integral toate concursurile disponibile pe internet in acest articol. Este foarte importanta citirea regulamentului inainte de rezolvarea

problemelor, deoarece pot exista restrictii care difera de la caz la caz. Datorita faptului ca majoritatea acestor concursuri sunt internationale, cunoasterea limbii engleze devine o necesitate pentru a putea participa. Pentru o pregatire cat mai eficienta, se recomanda participarea la acest tip de concursuri, atat pentru valoarea premiilor puse in joc, cat si ca modalitate de antrenament.

- **preONI** - Un concurs organizat in scopul pregatirii pentru Olimpiada Nationala de Informatica, organizat de catre studenti si elevi, actuali si fosti olimpici nationali si internationali. In ultimii 3 ani acest concurs a fost organizat de **echipa infoarena**, un grup de olimpici care administreaza primul (si singurul la momentul acesta) site din Romania cu evaluator disponibil 24 din 24. Acest concurs este intr-o continua expansiune, devenind din ce in ce mai complex de la an la an. Pe langa preONI, echipa infoarena organizeaza si alte concursuri; calitatea si valoarea concursurilor organizate precum si a materialelor educationale ce sunt puse la dispozitie este recunoscuta atat de elevi cat si de profesori de renume in informatica.
- **.campion** - Problemele de pe acest site sunt propuse de membri ai Comisiei Olimpiadei Nationale; concursul este structurat in runde de pregatire care dureaza in mod normal 10 zile, si in runde de concurs, care dureaza doar 3 ore; tipul rundei alterneaza. Primii clasati se intalnesc la "marea finala", care se desfasoara sub forma unei probe de concurs, similara cu olimpiadele - castigatorii primesc premii.
- **Bursele Agora** - Printre cele mai vechi concursuri care inca se desfasoara, este organizat de redactia GInfo; formatul acestuia difera de la an la an, in ultimii doi ani fiind structurat in 50 de runde normale + o runda finala. La acest concurs se acorda burse pentru primii doi clasati in valoare de 50\$ si 30\$
- **USACO** - Un concurs prin intermediul caruia se selecteaza lotul national largit al SUA. Concursul consta in mai multe faze, desfasurate pe parcursul anului. Prin bunavointa organizatorilor, concursul a devenit international, iar ultimele editii au adus ca noutate traducerea problemelor in mai multe limbi, printre care, uneori, si limba romana.
- **site-uri ACM** - exista site-uri care va pun la dispozitie si cate o arhiva foarte bogata de probleme si pot fi folosite cu succes in pregatirea pentru olimpiade, si care organizeaza concursuri, mai ales in perioada dinaintea concursurilor regionale ACM: astfel de site-uri sunt: <http://acm.uva.es>, <http://acm.timus.ru>, <http://acm.sgu.ru>,

<http://spoj.sphere.pl>, <http://acm.pku.edu.cn>, <http://acm.zju.edu.cn>. Pentru o lista mai mare de site-uri de pregatire accesati sectiunea [Links](#), o lista permanent actualizata.

- **Internet Problem Solving Contest** - concurs organizat o data pe an. In cadrul acestui concurs se pun la dispozitie enunturile problemelor si fisierele de intrare si se asteapta fisierele de iesire corespunzatoare, fara a se cere si programele care rezolva problemele.

Alte concursuri de programare, dar pe echipe, sunt **Bits'n'Bytes**, **BitWise** si **International Online Programming Contest**; la acestea pot participa si studenti. In anii precedenti au mai fost organizate si alte concursuri la noi in tara asemenea cu cele precizate mai sus precum **Lista lui Francu**, **Cupa Compaq**, **Cupa Fujitsu-Siemens**, **OlimpiadaOnline**, **Algoritmus**.

Concursuri pentru studenti

- **ACM** - un concurs organizat pentru studenti, de natura algoritmica. Diferentele majore fata de concursurile de liceu sunt reprezentate de faptul ca un program trebuie sa rezolve toate seturile de date de intrare prezentate pentru a se acorda puncte (nu se mai acorda punctaje pariale) si viteza cu care se rezolva problemele conteaza, iar evaluarea programelor este imediata. Concursul este pe echipe de cate 3 persoane, toate la un singur calculator. Concursul are loc pe regiuni initial (Centrul Europei, Sudul Europei, etc.) iar echipele calificate se intalnesc la marea finala.
- **TopCoder** - un site care organizeaza concursuri saptamanal, cat si turnee. Formatul este diferit fata de orice concurs intalnit pana acum, si anume fiecare concurent are de rezolvat 3 probleme in 75 de minute: o problema usoara de 250 puncte, una medie de 500 puncte si una grea de 1000 puncte. Punctajul efectiv pentru o problema se da in functie de viteza cu care aceasta este rezolvata. De asemenea, exista si o etapa de "challenge", in care concurentii pot vedea sursele celorlalti si, in caz ca determina un bug intr-o sursa, pot sa ruleze sursa respectiva, iar daca da raspuns gresit concurentul primeste 50 de puncte, altfel este penalizat cu 25 de puncte. Aceste concursuri sunt diferite fata de cele din liceu in care accentul este pe realizarea unui algoritm eficient, deoarece aici se pune accentul mai mult pe o implementare rapida si corecta a unor probleme care folosesc variatii ale unor algoritmi destul de cunoscuti, dar fiind necesara tratarea diferitelor cazuri speciale ce pot aparea. Ca avantaje, premiile la turnee sunt foarte mari (20.000\$

pentru primul clasat), iar intregul sistem de punctare este mult mai complex si mai eficient. Google organizeaza impreuna cu TopCoder concursul [Google Code Jam](#).

3. Inainte de concurs

Din fericire pentru unii si din nefericire pentru altii, majoritatea examenelor iti cer sa dovedesti nu ca esti bine pregatit, ci ca esti mai bine pregatit decat altii. Aceasta inseamna ca si la olimpiada de informatica se aplica legea concurentei. Valoarea absoluta a fiecaruia nu conteaza chiar in totalitate, ceea ce constituie sarea si piperul concursului. Intr-adevar, ce farmec ar avea sa mergi la un concurs la care sa se stie inca dinainte cine este cel mai bun? Este destul de amuzant sa observi cum fiecare spera sa prinda "o zi buna", iar adversarii sai "o zi proasta". Expresia "de la extaz la agonie" se potriveste foarte bine uneori cu ceea ce se poate intampla la concursurile de informatica.

Experienta demonstreaza ca, oricat de mare ar fi bagajul de cunostinte acumulat de un elev, mai e nevoie de ceva pentru a-i asigura succesul la olimpiada de informatica. Aceasta deoarece in timp de concurs lucrurile stau cu totul altfel decat in fata calculatorului de acasa sau de la scoala. Reusita depinde, desigur, in cea mai mare masura de puterea fiecaruia de a pune in practica ceea ce a invatat acasa. Numai ca in acest proces intervin o serie de factori care tin de temperament, de experienta individuala, de numarul de ore dormite in noaptea dinaintea concursului (care in taberele nationale este ingrijorator de mic) si asa mai departe. Trebuie spus ca un concurs de informatica presupune mai mult decat un simplu act de prezenta la locul desfasurarii ostilitatilor. Este chiar trist de remarcat cum spiritul competitiv capata de multe ori tente malitioase.

Primul si cel mai de seama lucru pe care trebuie sa il stiti este ca e important si sa participi, dar e si mai important sa participi onorabil, sa dai dovada de fair-play, iar daca se poate, sa si castigi! :) Nu trebuie sa porniti la drum cu ingamfare; modestia e buna, dar nu trebuie in nici un caz sa duca la neincredere in sine! Fiecare trebuie sa stie clar de ce e in stare si, mai presus de toate, sa se gandeasca ca la urma urmei nu dificultatea concursului conteaza, caci concursul, greu sau usor, este acelasi pentru toti. Mult mai importanta este valoarea individuala si nu in ultimul rand pregatirea psihologica. Fiecare concurs reprezinta de asemenea un prilej de perfectionare: fiecare concurent trebuie sa-si analizeze comportamentul din timpul concursului, si sa determine ce aspecte pot fi imbunatatite pentru a obtine performante mai bune la urmatoarele concursuri. Pregatirea

psihologica este de multe ori neglijata si reprezinta pentru unii exact acel lucru care il impiedica sa obtina performantele dorite. Exista destui concurenti care uita faptul ca aceste concursuri sunt doar niste concursuri, si ajung sa creada ca cel mai important lucru din viata lor este performanta lor la concursuri - fapt care poate fi foarte daunator in cazul unor esecuri. Asadar, recomand tuturor concurentilor sa adopte o atitudine relaxata si optimista la concursuri, sa realizeze ca acestea reprezinta doar o mica parte din viata, si nu in ultimul rand sa se distreze!

O pregatire intensa inaintea unui concurs este foarte importanta deoarece imbunatateste viteza de implementare si reduce riscul aparitiei erorilor, dar nu poate rezolva anumite lacune teoretice. Formula pentru succes poate fi descrisa ca "90% munca, 10% talent". Pentru a obtine performante mari trebuie neaparat o pregatire intensa adecvata. Exista mai multe aspecte ale pregatirii:

- Pregatirea teoretica
- Pregatirea psihica
- Organizarea globala a pregatirii
- Simularea unor probe de concurs
- Discutiile cu alti elevi si profesori referitor la anumite probleme
- Pregatirea de la locul desfasurarii probei

Desigur, un alt aspect care trebuie luat in considerare este supra-antrenamentul! Da, acest concept exista si in concursurile de informatica. Se intampla de obicei ca dupa un antrenament prea greu, mintea sa nu reactioneze la fel de bine ca inainte, fiindca este obosita. De aceea, se si recomanda minim o saptamana de pauza si relaxare totala inainte de orice concurs major.

Pregatirea teoretica

Majoritatea problemelor propuse spre rezolvare la concursuri depasesc cu mult nivelul manualelor de informatica. De exemplu, desi se propun o multime de probleme a caror rezolvare implica detinerea de cunostinte din domeniul teoriei grafurilor, nu toti algoritmi necesari sunt cuprinsi in programa de invatamant. Pentru a rezolva lacunele teoretice, este necesara studierea unor carti care sa acopere un spatiu teoretic cat mai vast.

Unele carti isi propun sa initieze cititorul in tainele diverselor limbaje de programare, altele pun accentul mai cu seama pe tehnicile de programare si structurile de date folosite in rezolvarea problemelor. In general, cele din

prima categorie contin exemple cu caracter didactic si exercitii cu un grad nu foarte ridicat de dificultate, iar celelalte demonstreaza matematic fiecare algoritm prezentat, insa neglijeaza partea de implementare, considerand scrierea codului drept un ultim pas lipsit de orice dificultate. Desigur, fiecare din aceste carti isi are rostul ei in formarea unui elev bine pregatit in domeniul informaticii. Totusi, trebuie considerata observatia ca scrierea unui program impune atat conceperea algoritmului si demonstrarea corectitudinii, cat si implementarea lui, ambele etape fiind complexe si nu lipsite de obstacole.

Cateva dintre cartile care ar trebui parcurse sunt in special cele scrise de fosti olimpici; prin intermediul acestora, autorii va impartasesc o parte din experienta acumulata. Cele mai folositoare carti (unele nu se mai gasesc) pentru pregatire sunt (unele nu sunt disponibile in Romania):

- **Introducere in Algoritmi (Cormen, Leiserson, Rivest)** - "Biblia" algoritmilor; mai este numita si CLR. Traducerea in limba romana a primei editii a acestei carti este disponibila la editura Agora Computer Libris si acum oricine este interesat o poate achizitiona. Contine o descriere amanuntita a tuturor algoritmilor de baza care pot fi folositi in rezolvarea problemelor de concurs.
- **Arta Programarii Calculatoarelor Vol. 1-3 (Donald Knuth)** - Donald E. Knuth este celebru datorita muncii sale de pionierat in domeniul algoritmilor si al tehnicilor de programare; "Daca te crezi un bun programator... citeste Arta Programarii Calculatoarelor de Knuth... Daca poti citi toata cartea, trimite-mi neaparat un C.V." - Bill Gates. Cartile se pot gasi la noi in tara la editura Teora
- **Proiectarea si implementarea algoritmilor (Mihai Oltean)** - aparuta la editura Computer Libris Agora din Cluj-Napoca; este o resursa foarte buna pentru programare dinamica
- **Culegere de probleme si programe PASCAL (Mihai Stroe, Cristian Cadar)** - aparuta la editura Petrion din Bucuresti, contine un capitol introductiv bun despre geometrie, cat si diverse solutii interesante la probleme din concursuri
- **Psihologia concursurilor de informatica (Catalin Francu)** - aparuta la editura L&S din Bucuresti; de asemenea contine solutii interesante la anumite probleme
- **Informatica - culegere de probleme pentru liceu (Emanuela Cerchez)** - aparuta la editura Polirom, contine diverse aplicatii pentru metodele principale de programare

- **Probleme de informatica date la concursurile internationale (Radu Berinde, Dan Ghinea, Horia Andrei Ciocina, Cornel Margine)** - aparuta la editura Fundatiei Pro, contine rezolvari pentru problemele de la principalele concursuri internationale din ultimii ani
- **Fundamentele programarii - Culegere de probleme pentru clasa a IX-a / clasa a X-a (Dana Lica, Mircea Pasoi)** - aparute la editura L&S din Bucuresti, ambele contin un capitol mare de probleme propuse pe la concursuri impreuna cu solutii
- **Arbori (Emanuela Cerchez)** - aparuta la editura Tara Fagilor, trateaza in detaliu arborii
- **Probleme de combinatorica si teoria grafurilor (Ioan Tomescu)** - desi este o carte in principal de matematica multe probleme de acolo au aparut la concursurile romanesti
- **Provocarea algoritmilor (Victor Mitrana)** - aparuta la editura Agni, Bucuresti
- **Computational Geometry: An Introduction (Shamos, Preparata)** - o carte de baza pentru geometria computationala
- **Computational Geometry in C (Joseph O'Rourke)** - alta carte de baza pentru geometria computationala
- **Algorithms (Robert Sedgewick)** - contine informatii despre algoritmi clasici si despre structurile de date cele mai folosite
- **The Algorithm Design Manual (Steven Skiena)** - o carte de referinta, continand implementari ale unui numar mare de algoritmi
- **Programming Challenges: The Programming Contest Training Manual (Steven Skiena, Miguel Revilla)** - o carte folositoare mai ales pentru concursurile ACM

In afara de cartile mentionate, internetul se dovedeste din nou o resursa foarte importanta. O cautare pe internet poate localiza informatii interesante: descrierea unor anumiti algoritmi impreuna cu performantele lor, tratari ale unor probleme clasice prin mai multe metode etc. Desigur pe langa carti este recomandat sa se si lucreze cat mai multe probleme de la editiile anterioare ale concursurilor principale si de pe site-urile cu evaluator disponibil 24 din 24. Astfel de site-uri sunt:

- infoarena
- <http://acm.timus.ru>
- <http://acm.sgu.ru>
- <http://spoj.sphere.pl>

- <http://acm.pku.edu.cn>
- <http://acm.zju.edu.cn>
- <http://ace.delos.com/usacogate>
- <http://www.oi.edu.pl/php/show.php?ac=e100000> (Site cu problemele polonezilor - o resursa foarte buna de pregatire, desi nu detine evaluator online)

Pregatirea psihica

Dupa cum s-a mai zis, este cunoscut faptul ca o atitudine mentala pozitiva este cheia succesului in cele mai multe situatii. Din nefericire, unii concurenti incep proba cu un moral nu tocmai ridicat. Iata cateva din falsele probleme cu care se confrunta anumiti concurenti:

- *Participa si X, care e mai bun ca mine, deci nu am nici o sansa sa castig!*
Fals! Nu s-a demonstrat ca X este mai bun, cel mult a obtinut rezultate mai bune pana acum si poate avea sanse mai mari. Problemele din concursul curent sunt aceleasi pentru toti, conditiile de desfasurare sunt aceleasi si antecedentele nu conteaza. Totul se reia de la zero. In plus, participarea la un concurs puternic poate aduce mai multa experienta pentru viitor. Pentru a ajunge la valoarea necesara castigarii unor concursuri, trebuie sa participati la cat mai multe si sa le tratati cu seriozitate.
- *Am obtinut prea putine puncte in prima zi, nu mai am nici o sansa la premii!*
Problemele de la concursurile cu mai multe probe sunt, in general, destul de dificile. De multe ori, la Olimpiada Nationala sau la concursurile internationale, obtinerea a jumatate din punctele puse in joc inseamna castigarea unui premiu. Daca in prima zi rezultatele obtinute sunt nesatisfacatoare, un rezultat foarte bun in ziua a doua poate aduce premiul dorit. In ultimii ani pentru intrarea in lotul national largit, la barajele de selectie, a fost suficienta obtinerea a mai putin de 200 puncte, dintre cele 600 posibile.
- *Nu sunt suficient de bine pregatit!*
Aceasta apreciere este, uneori, mai realista. Totusi, trebuie sa stiti ca o foarte mare importanta o are inspiratia de moment sau norocul (poate problemele vor fi similare cu unele rezolvate anterior). Evident, optimismul exagerat poate, la randul sau, sa fie daunator. Cel mai bine

ar fi sa adoptati atitudinea cea mai potrivita pentru propria personalitate. Veti observa, in timp, care este aceasta.

- *Problema asta e grea si n-am s-o pot rezolva perfect, asa ca nu ma mai apuc deloc de ea!*
Este usor sa fii printre cei mai buni atunci cand concursul este usor. Mai greu e sa fii cel mai bun atunci cand concursul este dur, pentru ca atunci intervine - inevitabil - dramul de noroc al fiecaruia. Niciodata insa nu se poate invoca greutatea concursului drept o scuza pentru un eventual esec. Concursul este la fel de greu pentru toti. Se poate intampla, mai ales daca probele dureaza mai multe zile (2-3) ca nici unul din concurenti sa nu acumuleze mai mult de 50-60% din punctajul maxim. Totusi, aceasta nu inseamna ca ei nu sunt bine pregatiti; mai mult, unul dintre ei trebuie sa fie primul. Asadar, niciodata nu trebuie adoptata o strategie de genul acesta. Nu trebuie sa va impacientati daca vi se intampla sa nu aveti o idee geniala de rezolvare a unei probleme. Nu va cere nimeni sa faceti perfect o problema, ci numai sa prezentati o solutie care sa acumuleze cat mai multe puncte. Evident, prima varianta este intotdeauna preferabila, dar nu obligatorie. De multe ori se intampla ca un elev sa gaseasca o solutie cat de cat buna la o problema si, macar ca stie ca nu va lua punctajul maxim, ci doar o parte, sa renunte sa caute o solutie mai eficienta, deoarece timpul pierdut astfel ar aduce un castig prea mic si ar putea fi folosit la rezolvarea altor probleme. Desigur, daca nu faci toate problemele perfect, nu mai poti fi sigur de locul I, pentru ca altcineva poate sa te intreaca. Dar pe de alta parte, locul pe care te clasezi conteaza numai la etapa nationala a olimpiadei sau la concursurile internationale. In rest, important e numai sa te califici, adica sa intri in primele cateva locuri.

Pregatirea psihica nu are legatura numai cu concursul propriu-zis. O atitudine mentala pozitiva, de invingator, este utila pe tot timpul pregatirii pentru concursuri.

Organizarea globala a pregatirii

Acest aspect poate contribui la obtinerea unor rezultate excelente. O parte foarte importanta a unei pregatiri sistematice consta in elaborarea unei liste cu metodele si tehnicile cunoscute si necunoscute, punctele slabe, etc. Lista trebuie sa contina algoritmi care apar in mod frecvent in cadrul problemelor de concurs, problemele care apar la implementare, la organizarea timpului

in concurs, etc. Continului listei se modifica in timp; o parte din algoritmi necunoscuti sunt invatati si devin cunoscuti, se descopera noi puncte slabe, se evidentiaza existenta unor algoritmi necunoscuti care trebuie invatati, etc. Aceasta este o modalitate excelenta de a masura progresul si de a gasi noi directii de urmat in pregatire.

Simularea unor probe de concurs

Cea mai potrivita modalitate de pregatire pentru a face fata unei situatii este simularea ei, adica tratarea unei situatii asemanatoare. Acest principiu este valabil in orice tip de competitie, aplicandu-se si in domeniul concursurilor de programare. Daca va pregatiti pentru un concurs, citirea unor carti nu este suficienta! Trebuie sa va analizati comportamentul in situatii similare. Situatiile cea mai asemanatoare unui anumit concurs este un alt concurs de acelasi tip. Participarea la N concursuri creste sansele obtinerii unui rezultat mai bun la al $N+1$ -lea.

Concursurile de pe internet sunt destul de dese si sunt organizate foarte bine. Din nefericire, nivelul de dificultate al acestor concursuri nu este intotdeauna cel dorit de cel care se pregateste. O posibilitate de simulare a unui concurs este rezolvarea problemelor de la o editie precedenta! De exemplu, cineva care se pregateste pentru Olimpiada Nationala ar trebui sa rezolve problemele date la Olimpiada Nationala din anul anterior la clasa respectiva (exista si cazuri in care simulari de acest tip nu sunt foarte concludente; de exemplu, intre 2000 si 2001 programa scolara a suferit cateva variatii, care s-au reflectat in tipul problemelor de la clasa a X-a). Pentru ca simularea sa reflecte cat mai bine realitatea, problemele se rezolva in timpul stabilit, fara pauze, la prima citire (sau, daca au fost citite anterior, nu se recitesc in zilele premergatoare simularii). Experienta obtinuta este apropiata de cea a concursului propriu-zis, dar lipseste stresul care apare, inevitabil, in timpul competitiei. Este important ca, dupa fiecare simulare sau concurs real, sa va analizati comportarea si sa invatati din eventualele greseli de abordare (de exemplu, puteti ajunge la concluzii de genul *Aceasta problema trebuia abordata prima sau Nu am citit integral enuntul si am rezolvat o alta problema*). Concluziile analizei duc la detectarea actiunilor necesare pentru a imbunatati situatia.

In plus, se recomanda notarea celor mai frecvente greseli de implementare si examinarea periodica a listei (inclusiv in timpul depanarii programelor, in cadrul simularii concursurilor); in acest fel, greselile respective vor disparea in timp.

Discutiile cu alti elevi si profesori

Puteti invata foarte mult de la profesori, si in special de la elevi mai experimentati! Exista foarte multe exemple de elevi pentru care a contat foarte mult pregatirea individuala, dar exista unii care au fost ajutati de pregatirea organizata, in grupuri de elevi, sub indrumarea unor profesori cu preocupari de acest gen. In cadrul pregatirilor de acest tip se discuta algoritmi, se propun probleme spre rezolvare, se discuta diversele modalitati de rezolvare, se obtin mai multe informatii despre concursuri, etc. In plus, elevii aduc in discutie diverse probleme cu care s-au intalnit in cadrul pregatirii individuale. Daca doriti sa participati la astfel de pregatiri, trebuie sa luati legatura cu alti elevi interesati de concursuri, sau cu profesori care se ocupa de pregatirea elevilor pentru olimpiade. In prezent se organizeaza pregatiri la nivel de liceu, oras, judet, etc. in anumite zone. Astfel de pregatiri cresc valoarea tuturor participantilor, deci sunt foarte importante. Pregatirile organizate au luat amploare odata cu infiintarea centrelor de excelenta. Pentru a intra in contact cu alti elevi interesati puteti folosi [forumul infoarena](#) si [forumul revistei GInfo](#).

Pregatirea de la locul desfasurarii probei

Desi ar putea parea bizar, acest aspect este foarte important, dar este neglijat de multe ori de catre concurenti. Aceasta pregatire consta in:

- somn odihnitor in noaptea care precede ziua concursului (**foarte important!**)
- obtinerea atitudinii mentale dorite
- prezentarea la timp in sala, cu toate obiectele necesare

Exista cateva obiecte pe care ar trebui sa le aveti la voi. In timpul concursului trebuie tinuta o evidenta drastica a timpului scurs si a celui ramas. E drept ca in general supraveghetorii anunta din cand in cand timpul care a trecut, dar e bine sa nu va bazati pe nimeni si nimic altceva decat pe voi insiva. Unii pot spune *Ei, ce nevoie am de ceas, oricum am ceasul calculatorului la indemana*. Asa e, dar e incomod sa te opresti mereu la jumatatea unei idei si sa verifici cat e ceasul schimband consola sau minimizand fereastra de lucru. In ceea ce priveste hartia de scris, ea este in mod sigur necesara. De fapt, o parte importanta a rezolvarii unei probleme este proiectarea matematica a algoritmului, lucru care nu se poate face decat cu creionul pe hartie. Pe langa aceasta, majoritatea problemelor opereaza cu vectori, matrice, arbori, grafuri, etc., iar exemplele pe care este testat programul realizat trebuie neaparat verificate "de mana". Este recomandat sa aveti mereu si hartie de matematica; este foarte folositoare pentru problemele de geometrie analitica, precum si pentru reprezentarea

matricelor. Nu in ultimul rand, ar fi bine sa aveti o sticla de suc si o ciocolata; din nefericire, concursul incepe deseori cu intarziere si este bine ca foamea sau setea sa nu va preocupe in timpul rezolvarii problemelor.

4. In timpul concursului

Imediat ce primiti problemele, cititi toate enunturile si faceti-va o idee aproximativa despre gradul de dificultate al fiecarei probleme. Neaparat verificati daca se dau limite pentru datele de intrare (numarul maxim de elemente ale unui vector si valoarea maxima a acestora, numarul maxim de noduri dintr-un graf, etc.) si pentru timpii de executie pentru fiecare test. Dimensiunea input-ului poate schimba radical dificultatea problemei. Spre exemplu, pentru un vector cu $N \leq 200$ elemente, un algoritm $O(N^3)$ merge rezonabil, pe cand pentru $N \leq 2000$ acelasi algoritm ar depasi cu mult cele cateva sutimi de secunda care se acorda de obicei. In primele zece minute (sau mai mult) nu se atinge calculatorul. Intotdeauna, cand cititi o problema, este indicat sa intoarceti foaia pentru a vedea daca enuntul continua si pe verso. De obicei, in primele 30 sau 60 de minute ale concursului pot fi adresate intrebari comisiei, pentru a clarifica eventualele ambiguitati din enunturi. Acestea sunt redactate in scris, foile sunt preluate de supraveghetorul din sala si trimise la comisie. Raspunsul s-ar putea sa intarzie, deci este indicat sa nu irositi timpul asteptand raspunsul fara a mai face nimic altceva. Puteti fie sa va ganditi la rezolvarea unei probleme, fie sa incepeti sa implementati (daca exista ceva usor de implementat, cum ar fi o problema simpla sau o rutina pentru citirea datelor de intrare). In majoritatea situatiilor, intrebarile trebuie formulate in asa fel incat raspunsul sa fie *Da* sau *Nu*. Daca intrebarea nu este astfel exprimata sau daca raspunsul se gaseste in textul problemei, veti primi raspunsul *Fara comentarii*, caz in care va trebui mai intai sa studiat corectitudinea intrebării si, daca aceasta este corect formulata, sa recititi enuntul problemei. Concurrentii trebuie sa profite cat mai mult de aceasta perioada, pentru a clarifica eventualele nelamuriri. Un lucru important este ca nu trebuie sa acceptati raspunsuri daca acestea nu sunt insotite de semnatura unui membru al comisiei.

Faceti o impartire a timpului pentru problemele ramase proportional cu dificultatea aparenta a fiecarei probleme. In general problemele au punctaje egale. Incercati sa nu depasiti niciodata limitele de timp pe care le-ati fixat. Daca in schimb reusiti sa economisiti timp fata de cat v-ati propus, cu atat mai bine, veti face o realocare a timpului si veti avea mai mult pentru celelalte probleme. Apucati-va de problema **cea mai simpla**. Mai bine sa duceti la bun sfarsit o problema usoara, decat sa va apucati de o problema grea si sa nu terminati nici una. Daca toate problemele par grele, alegeti-o

pe cea din domeniul care va este cel mai familiar, in care ati lucrat cel mai mult. Daca va este indiferent si acest lucru, alegeti o problema unde simtiti ca aveti o idee simpla de rezolvare.

Incepeti sa va ganditi la algoritmi cat mai buni, estimand in acelasi timp si cat v-ar lua ca sa-i implementati. Faceti, pentru fiecare idee care va vine, calculul complexitatii. Nu trebuie neaparat sa gasiti cel mai eficient algoritm, ci numai unul suficient de bun. In general, trebuie ca, dintre toti algoritmii care se incadreaza in timpul de rulare, sa-l alegeti pe cel care este cel mai usor de implementat. Daca algoritmul gasit este greu de implementat, mai cautati altul o vreme. Trebuie insa ca timpul petrecut pentru gasirea unui nou algoritm plus timpul necesar pentru scrierea programului sa nu depaseasca timpul necesar pentru implementarea primului algoritm, altfel nu castigati nimic. Deci nu exagerati cu cautarile si nu incercati sa reducati dincolo de limita imposibilului complexitatea algoritmului. Mai ales, nu uitati ca programul nu poate avea o complexitate mai mica decat dimensiunea input-ului sau a output-ului. De exemplu, daca programul citeste sau scrie matrice de dimensiune $N \times N$, nu are sens sa va bateti capul ca sa gasiti un algoritm mai bun decat $O(N^2)$. Dintre toate ideile de implementare gasite (care se incadreaza fara probleme in timp), o veti alege pe cea mai scurta ca lungime de cod.

In general, pentru orice problema exista cel putin o solutie, fie si una slaba. Sunt numeroase cazurile cand nu va vine alta idee de rezolvare decat cea slaba. De regula, cand nu aveti in minte decat o rezolvare neeficienta a problemei, care stiti ca nu o sa treaca toate testele (un backtracking, sau un $O(N^5)$, $O(N^6)$, etc.) e bine sa incercati urmatorul lucru:

- Calculati cam cat timp v-ar trebui ca sa implementati rezolvarea slaba. In acest calcul trebuie sa includeti si un timp estimativ de depanare a programului (care variaza de la persoana la persoana) si pe cel de testare. Daca sunteti foarte siguri pe voi, puteti sa neglijiati timpul de testare, dar orice program trebuie testat cel putin pe exemplul de pe foaie.
- Pentru a avea sanse mai mari sa gasiti o alta solutie, este indicat sa incercati sa ignorati complet solutia slaba, sa nu o luati ca punct de plecare. Incercati sa va "goliti" mintea si sa gasiti ceva nou, altfel va veti invarti mereu in cerc.
- Daca va vine vreo idee mai buna, ati scapat de griji si va apucati de implementare (daca aveti timpul necesar).

Din acest moment, pentru varianta aleasa veti scrie programul, fara a va mai gandi la altceva, chiar daca pe parcurs va vin alte idei. Iata unele lucruri pe care e bine sa le stiti despre scrierea unui program:

- Datele de intrare se presupun a fi corecte. Aceasta este o regula nescrisa (uneori) a concursului de informatica.
- Efectul optiunilor de compilare asupra vitezei este semnificativ.
- Daca se poate, evitati lucrul cu pointeri. Programele care ii folosesc sunt mai greu de depanat si se pot bloca mult mai usor.
- Evitati lucrul cu numere reale (comparatii, impartiri, etc.), daca puteti. Operatiile in virgula mobila sunt mult mai lente.
- Alegeti-va numele de variabile in asa fel incat programul sa fie clar. Sunt permise mai mult de doua litere! Numele fiecarei proceduri, functii, variabile trebuie sa-i explice clar utilitatea. E drept, lungimea programului creste, dar codul devine mult mai limpede si timpul de depanare scade foarte mult. Ca o regula generala, claritatea programelor face mult mai usoara intelegerea lor chiar si dupa o perioada mai indelungata de timp (luni, ani). Nu trebuie nici sa cadeti in cealalta extrema. De exemplu, nu depasiti 10 caractere pentru un nume de variabila.
- Salvati programul cat mai des. Daca va obisnuiti, chiar la fiecare 2-3 linii. Dupa ce o sa va intre in reflex n-o sa va mai incomodeze cu nimic acest obicei. Au fost cazuri in care o pana de curent prindea pe picior gresit multi concurenti, iar dupa aceea nu mai este absolut nimic de facut, pentru ca nimeni nu va va crede pe cuvnt ca ati facut programul si ca el mergea.
- Obisnuiti-va sa programati modular. Faceti proceduri separate pentru citirea si initializarea datelor, pentru sortare, pentru afisarea rezultatelor, etc. In general nu se recomanda sa scrieti proceduri in alte proceduri (adica e bine ca toate procedurile sa apartina direct de programul principal). Procedurile, acolo unde e posibil, nu trebuie sa depaseasca un ecran, pentru a putea avea o viziune de ansamblu asupra fiecareaia in parte. Acest lucru ajuta mult la depanare.
- Rulati programul cat mai des, daca timpul va permite. In primul rand dupa ce scrieti procedura de citire a datelor. Daca e nevoie de sortarea datelor de intrare, nu strica sa va convingeti ca programul sorteaza bine, ruland 2-3 teste oarecare. E pacat sa pierdeti puncte dintr-o greseala copilareasca.

- O situatie mai delicata apare cand fisierul de intrare contine mai multe seturi de date (teste). In acest caz, atentia trebuie sporita, deoarece daca la primul sau al doilea test programul vostru da eroare si se opreste din executie, veti pierde automat si toate celelalte teste care urmeaza. Daca in fisierul de intrare exista un singur set de date, atunci pierderea din vedere a unui caz particular al problemei nu putea duce, in cel mai rau caz, decat la picarea unui test. Asa insa, picarea unui test poate atrage dupa sine picarea tuturor celor care il urmeaza. Pe langa corectitudinea strict necesara, programul trebuie sa se incadreze si in timp pentru orice fel de test. Daca la primul sau al doilea test din suita programul depaseste timpul (sau, si mai rau, se blocheaza), e foarte probabil sa fie oprit din executie de catre comisie, deci din nou veti pierde toate testele care au ramas neexecutate.
- Tot in situatia in care exista mai multe seturi de date in fisierul de intrare, daca iesirea se face intr-un fisier, este bine ca dupa afisarea rezultatului pentru fiecare test sa actualizati fisierul de iesire. In felul acesta, chiar daca la unul din teste programul se blocheaza sau da eroare, rezultatele deja scrise raman scrise. Altfel, e posibil ca rezultatele de la testele anterioare sa ramana intr-un buffer in memorie, fara a fi "varsate" pe disc.

Tot la partea de implementare, este bine ca codul sa fie cat mai scurt si cat mai optimizat - dar, despre scrierea unui cod cat mai eficient se poate face un articol cam la fel de mare cat acesta, deci nu se va trata acest subiect aici - metoda cea mai buna in acest sens este sa invatati din sursele altora. Puteti incepe cu articolele [12 ponturi pentru programatorii C/C++](#) si [Multe smenuri de programare in C/C++... si nu numai!](#) si sectiunea [Links](#).

Mai ramane doar partea de depanare. O metoda buna de depanare este urmatoarea:

- Incepeti cu un test nici prea simplu, nici prea complicat (si usor de urmarit cu creionul pe hartie) si executati-l de la cap la coada. Daca merge perfect, treceti la teste mai complexe (se recomanda **minim** 4 teste si maxim 7-8). Daca le trece si pe acestea, puteti zambi. Totusi, daca programul vostru a mers perfect pe 7-8 teste date la intamplare, exista sanse (dar nu extrem de mari!) sa mearga pe majoritatea testelor comisiei, sau chiar pe toate.
- Exemplul dat in enunt nu are in general nici o semnificatie deosebita (de fapt, are mai curand darul de a semana confuzie printre

concurrenti), iar daca programul merge pe acest test particular, nu inseamna ca o sa mearga si pe alte teste.

- Daca la unul din teste programul nu merge corespunzator, rulati din nou testul , dar de data aceasta procedura cu procedura. Dupa fiecare procedura evaluati variabilele si vedeti daca au valorile asteptate. In felul acesta puteti localiza cu precizie procedura, apoi linia unde se afla eroarea. Corectati in aceasta maniera toate erorile, pana cand testul este trecut.
- In acest moment, luati de la capat toate testele pe care programul le-a trecut deja. In urma depanarii, s-ar putea ca alte greseli sa iasa la suprafata si programul sa nu mai mearga pe vechile teste.
- Repetati procedeul de mai sus pana cand toate testele merg. Daca va obisnuiti sa programati modular si ingrijit, depanarea si testarea n-ar trebui sa dureze mai mult de 5-25 minute. Din acest moment, nu mai modificati nici macar o litera in program, sau daca o faceti pastrati-va in prealabil o copie. Nu va bazati pe faptul ca puteti sa tineti minte modificarile facute si sa refaceti oricand forma initiala a programului in caz ca noua versiune nu va fi buna.
- Daca totusi nu-i puteti "da de cap" programului, iar timpul alocat problemei respective expira, aduceti programul la o forma in care sa mearga macar pe o parte din teste si treceti la problema urmatoare.

Feriti-va ca de foc de criza de timp. E mare pacat sa ratezi o problema intreaga pentru ca n-ai avut timp sa scrii procedura de afisare a solutiei, sau lucruri asemanatoare. Rezervati-va intotdeauna timpul pe care il socotiti necesar pentru implementare si depanare. De asemenea, chiar daca concursul este usor, nu e recomandat sa iesiti din sala de concurs inainte de expirarea timpului. Oricat ati fi de convinsi ca ati facut totul perfect, mai verificati-va; veti avea de furca cu remuscarile daca descoperiti dupa aceea ca ceva, totusi, nu a mers bine. Puteti face o multime de lucruri daca mai aveti timp (desi acest lucru se intampla rar). Iata o serie de metode de a exploata timpul:

- Verificati-va programul cu cat mai multe teste de mici dimensiuni. Sa presupunem ca programul vostru lucreaza cu vectori de maxim 10.000 de elemente. E o idee buna sa il rulati pentru vectori de unul sau doua elemente.
- Treceti la polul opus si creati-va un test de dimensiune maxima, dar cu o structura particulara, pentru care este usor de calculat rezultatul si

de mana. De exemplu, vectori de 10.000 de elemente cu toate elementele egale, sau vectori de forma (1, 2, ..., 9999, 10000). Daca nu puteti sa editati un asemenea fisier de mana, copiind si multiplicand blocuri, puteti scrie un program care sa-l genereze.

- Daca inca v-a mai ramas timp, creati-va un program care sa genereze teste aleatoare. Spre exemplu, un program care sa citeasca N si sa creeze un fisier in care sa scrie N numere aleatoare. Intr-o prima faza, puteti folosi aceste teste pentru a verifica daca nu cumva la valori mai mari programul nu da eroare, nu se blocheaza (la alocarea unor zone mari de memorie) sau nu depaseste limita de timp, caz in care mai aveti de lucru.
- Daca tot nu va da nimeni afara din sala, puteti scrie un alt program auxiliar care, primind fisierul de intrare si fisierul de iesire produs de programul vostru, verifica daca iesirea este corecta. Aceasta deoarece, de obicei, este mult mai usor de verificat o solutie decat de produs una. Folosind "generatorul" de teste si "verificatorul", puteti testa programul mult mai bine. De altfel, la multe probleme chiar testele rulate de comisia de corectare sunt create tot aleator.

5. Dupa concurs

Dupa ce ati terminat problemele (se intampla destul de rar) nu iesiti din sala! Este momentul ultimelor teste. La iesirea din sala trebuie sa fiti convinsi ca ati facut tot ce era posibil in conditiile date. Concursul nu s-a terminat inca! Urmeaza corectarea. Va trebui sa verificati punctajul obtinut si sa fiti pregatit sa depuneti o contestatie daca aveti impresia ca ceva nu este in regula. La unele concursuri, corectarea se face in prezenta concurentului; aici aveti ocazia sa solicitati sa vi se arate testele si iesirile furnizate de programul vostru, sa cereti testarea din afara mediului de evaluare, etc. La alte concursuri, comisia ofera, mai tarziu, testele si raspunsurile corecte pentru autoevaluare. Nu ratati ocazia de a va evalua rezolvarile si nu depuneti contestatii decat daca in urma autoevaluarii obtineti un punctaj mai mare. Fiecare concurs este o experienta in plus! Discutati, dupa proba, cu alti concurenti, aflati cum ar fi trebuit rezolvate problemele pe care nu le-ati stiut aborda si ce au gresit ceilalti (este bine sa invatati si din greselile altora).

De multe ori, primul an de participare la olimpiada se soldeaza cu un rezultat cel mult mediu, deoarece, oricat ar spune cineva *Ei, nu-i asa mare lucru sa mergi la un concurs*, experienta acumulata conteaza mult. De aceea, abia de la a doua participare si uneori chiar de mai tarziu incep sa

apara rezultatele. Intentia autorului a fost sa va usureze misiunea si sa va dezvaluie cateva din dificultatile de toate felurile care apar la orice concurs, pentru a nu va da ocazia sa le descoperiti pe propria piele. Speram ca aceste ponturi va vor fi de folos!

Bibliografie

1. **Psihologia concursurilor de informatica**, Catalin Francu, Editura L&S Bucuresti
2. **Despre Concursuri**, Mihai Stroe, Gazeta de Informatica, numarul 13/4, anul 2003
3. [infoarena](#)
4. [Google](#)