Sfaturi de bună practică

pentru concurenţii OJI / ONI 2013

Elevii care vor participa la Olimpiada Judeţeană/Naţională de Informatică trebuie să aibă în vedere următoarele:

A) Pentru evitarea erorilor la compilare, codul surselor trebuie să **respecte Standarul C++**, respectiv S**tandardul Free Pascal** (cu alte cuvinte, să fie corect din punct de vedere al limbajului)

B) Faptul că vitezele de citire/scriere a datelor pot să difere de la o distribuţie GNU C la alta.

C) Faptul că există reguli de formatare a fişierelor de test la concursurile de algoritmică.

A) Respectarea standardului limbajului.

Codul scris în conformitate cu standardul limbajului este un **cod portabil** şi va fi compilat fără erori cu oricare compilator GNU C.
Ne vom referi în special la limbajele C/C++, deoarece în cazul acestora apar cele mai multe situaţii de practică defectuoasă.

1. **Nu salvaţi soluţiile cu extensia .c, decât dacă sunteţi siguri că NU doriţi să utilizaţi biblioteca C++.**
   Fişierele cu extensia **.cpp** vor fi compilate conform standardului **C++** (ISO C++ 98), iar cele cu extensia **.c** vor fi compilate conform standardului **C** (ISO C 99). În exemplu, `sursa.c` nu va compila, în timp ce `sursa.cpp` compilează:

sursa.c	sursa.cpp

`// Se foloseste biblioteca C`
`#include <stdio.h>  // OK`
 	
`// NU puteti folosi Biblioteca C++ !!`
`#include <fstream.h> // Eroare`
 	
`struct A { };`
`A x; // Eroare (standardul C)`
 	
`int main()`
`{`
`  // variabila locala i in for`
`  for (int i = 0; i < 10; ++i) //Eroare`
`  printf("%d", i);`
 	
`  return 0;`
`}`
	`// Se foloseste biblioteca C`
	`#include <stdio.h>  // OK`
	
	`// Se foloseste Biblioteca C++`
	`#include <fstream.h>  // OK`
	
	`struct A { };`
	`A x; // OK (standardul C++)`
	
	`int main()`
	`{`
	`  // variabila locala i in for`
	`  for (int i = 0; i < 10; ++i) // OK`
	`  printf("%d", i);`
	
	`  return 0;`
	`}`

2. Header-e

Standardul limbajul C++ defineşte 33 de headere, conform tabelului:

<algorithm>	<iomanip>	<list>	<queue>	<streambuf>
<bitset>	<ios>	<locale>	<set>	<string>
<complex>	<iosfwd>	<map>	<sstream>	<typeinfo>
<deque>	<iostream>	<memory>	<stack>	<utility>
<exception>	<istream>	<new>	<stdexcept>	<valarray>
<fstream>	<iterator>	<numeric>	<sstream>	<vector>
<functional>	<limits>	<ostream>		

Facilităţile **Bibliotecii C Standard** sunt furnizate de 18 headere adiţionale:

<cassert>	<ciso646>	<csetjmp>	**<cstdio>**	<ctime>
<cctype>	<climits>	<csignal>	<cstdlib>	<cwchar>
<cerrno>	<clocale>	<cstdarg>	**<cstring>**	<cwctype>
<cfloat>	**<cmath>**	<cstddef>		

IMPORTANT!
Heder-ele cu extensia .h sunt *deprecated* pentru limbajul C++. Compilatoarele utilizate la OJI/ONI 2013 nu suportă stilul vechi de declarare

Exemple:

Corect	Deprecated (pentru headere C++)

`#include <fstream> // header-e C++`
`#include <iostream>`
`#include <iomanip>`
 	
`#include <cmath>  // header-e C`
`#include <cstdio>`
`#include <cstring>`
`using namespace std;`
 	
// directiva **using** precizează spaţiul de nume **std**
// în care este definită Biblioteca Standard C++
	`// Incorect (deprecated)`
	`#include <fstream.h>`
	`#include <iostream.h>`
	`#include <iomanip.h>`
	
	`// Corect (nondeprecated)`
	`#include <math.h>`
	`#include <stdio.h>`
	`#include <string.h>`

Observaţie: Pentru header-ele C, standardul acceptă (deocamdată) declarări cu extensia .h : <stdio.h>, <math.h>, <string.h>, etc.

B). Vitezele operaţiilor de intrare-ieşire în limbajele C/C++

Pentru o distribuţie GNU C dată, operaţiile de citire/scriere C şi C++ diferă uneori în ce privește viteza de executare.
Funcţiile **`scanf()`**, **`printf()`** de pildă, sunt pentru anumite distribuţii, mai rapide decât operaţiile de inserţie (<<) sau extracţie (>>) din stream-uri, în timp ce pentru alte distribuţii lucrurile stau exact invers !

Concurenţii sunt sfătuiţi să studieze mediile de lucru pentru OJI/ONI şi să aleagă acele metode de citire/scriere pe care le consideră optime.

1. **OJI 2013.**
   Compilatorul mediului **Code::Blocks 10.05** are particularitatea că produce executabile pentru care **vitezele de citire-scriere cu stream-uri sunt mai lente decât operaţiile similare cu funcţii.**

2. **ONI 2013**

Pentru mediile instalate la ONI 2013, (Ubuntu 12.04: **gcc 4.6.3**, Windows 7: **gcc 4.4.1**), s-au facut teste de viteză, aşa cum se poate vedea în tabelul de mai jos:

Ubuntu 12.04
gcc 4.6.3
Procesor: Intel(R) Core TM i-5 2320 CPU 3.00 GHz/ Core

Iteratii	<cstdio>		<fstream>	
	fscanf()	fprintf()	>>	<<
1000000	0.080 sec	0.085 sec	0.070 sec	0.070 sec
10000000	0.820 sec	0.770 sec	0.630 sec	0.600 sec

Windows W7
gcc 4.4.1 (pachetul OJI)
Procesor: Intel(R) Core TM i-5 2320 CPU 3.00 GHz/ Core

Iteraţii	<cstdio>		<fstream>	
	fscanf()	fprintf()	>>	<<
1000000	0.190 sec	1.330 sec	0.360 sec	1.330 sec
10000000	1.630 sec	2.600 sec	3.480 sec	2.230 sec

Concluzie:

- **În Linux e mai rapidă citirea şi scrierea cu ajutorul stream-urilor**
- **În Windows scrierea are loc cu viteze aproximativ egale, însă citirea este mai rapidă cu ajutorul funcțiilor C.**

ATENTIE! **NU folosiţi** `endl` **. Utilizaţi** `'\n'`**.**

La scrierea în streamuri, datele de ieşire se acumulează într-un buffer (stream) care se goleşte periodic, nefiind nevoie de accesarea discului la fiecare operaţie de scriere.
Manipulatorul de format `endl`, face *flush* stream-ului de ieşire, ceea ce forţează scrierea pe disc. Dacă aceasta se întamplă într-un ciclu, atunci viteza scade catastrofal:

Ubuntu 12.04
gcc 4.6.3
Procesor: Intel(R) Core TM i-5 2320 CPU 3.00 GHz/ Core

`#include <fstream>`
`using namespace std;`
 	
`int main()`
`{`
`  ofstream fout("numere.out");`
`  int n = 10000000;`
`  for (int i = 0; i < n; ++ i)`
`  fout << 7 << '\n';`
`  fout.close();`
`}`
	`#include <fstream>`
	`using namespace std;`
	
	`int main()`
	`{`
	`  ofstream fout("numere.out");`
	`  int n = 10000000;`
	`  for (int i = 0; i < n; ++ i)`
	`  fout << 7 << endl;`
	`  fout.close();`
	`}`

Timp de executare: **0,98** secunde	Timp de executare: **18.16** secunde !!!!

C) Formatul fişierelor

1. La OJI şi ONI, ca de altfel la toate competiţiile de algoritmică naţionale sau internaţionale, este o practică curentă aceea ca **ultima linie din fişierele de test de intrare, cât şi din cele de ieşire, să se termine cu caracterul newline.** Concurenţii trebuie să ţină seama de acest lucru.

2. **Salvați sursele doar cu numele și extensiile enumerate în enunțul problemelor**.

3. **Respectați formatul fișierului de de ieșire!**
   De exemplu, fie o problemă care are două cerințe: a) și b). Să presupunem că se cere ca răspunsul pentru cerința a) trebuie se găsească pe linia 1, iar răspunsul pentru cerința b) să se găsească pe linia 2. În situația în care nu ați reușit să rezolvați cerința a) dar aveți un răspuns pentru b), veți scrie răspunsul pentru cerința b) pe linia 2 și nu pe prima linie !