

# Introduction to Tom's Solutions for IOI'94

I have analysed and solved all six competition problems of IOI'94 (I did not do the 'backup' problems). My analyses and solutions are neither claimed to be the only ones nor to be the best. Nevertheless, I hope they will be useful to future competitors, coaches, and organizers.

I have written all my solutions in Turbo Pascal. There are three reasons for this.

1. Turbo Pascal was used by 75% of the competitors at IOI'94 (20% used Turbo C++, 4% used QuickBasic, and 1% used LCN Logo). In fact, all competitors finishing in the top 15 used Turbo Pascal.
2. Pascal is easier to read and understand than C++, Basic, and Logo.
3. I personally prefer Pascal over C++, Basic, and Logo for writing programs.

I have avoided special (Turbo) Pascal features as much as possible. This is not possible when dealing with files other than standard input and standard output (see below). Furthermore, I have sometimes used the (less known) standard Pascal functions `succ` and `pred` (which return the successor and predecessor respectively) and the Turbo Pascal procedures `inc` and `dec` (which increment and decrement respectively). For integer variable `v` and integer expression `e`, their meaning is captured by

```
succ(e)    is the same as    e+1
pred(e)    is the same as    e-1
inc(v, e)  is the same as    v := v + e
dec(v, e)  is the same as    v := v - e
inc(v)     is the same as    inc(v, 1)
dec(v)     is the same as    dec(v, 1)
```

All the programming problems at IOI'94 have the same overall i/o structure. Input is offered in a text file named `INPUT.TXT`, output is to be produced in a text file named `OUTPUT.TXT`. All my programs start as follows:

```
program ioi94dayXprbYverZ(input, output, inp, out);
{ Tom Verhoeff, Eindhoven University of Technology }
{ "One sentence describing idea behind this solution" }
```

The standard files `input` and `output` are used for development testing only. The files `inp` and `out` are used for reading the input and writing the output. (In Turbo Pascal you can omit the file parameters in the heading.)

The program heading is followed by a general section that defines procedures to open and close the files `inp` and `out`:

```
{ General Section }

const
  Test = true ; { set to false before compiling final version }

var
  inp, out: text ;

procedure Init ;
begin
  if Test then writeln('IOI''94 - Day 1 - Problem 3: The Primes') ;
  assign(inp, 'input.txt') ;
  reset(inp) ;
```

```
assign(out, 'output.txt') ;
rewrite(out) ;
if Test then writeln('Initialized')
end { Init } ;

procedure Fini ;
begin
close(inp) ;
close(out)
end { Fini } ;
```

The constant `Test` is used to control the input and output of data for development testing (not for grading after the competition). I have kept it at `true` because then some helpful output is produced. When the programs are to be tested officially, it should be set to `false`.

The last part of the program is the problem-specific section. It usually looks like this:

```
{ Problem-Specific Section }

var
  "Global variables for input data and answer"

procedure ReadInput ;
  "Read input from file inp into global variables"

procedure ComputeAnswer ;
  "Compute global answer variables from global input variables"

procedure WriteOutput ;
  "Write output from global answer variables to file out"

begin
Init ;
ReadInput ;
ComputeAnswer ;
WriteOutput ;
Fini
end.
```

For some problems it is easier to output the answer during the computation, instead of first storing it into global variables.

Procedure and function headings are sometimes documented in the following way:

```
procedure P(a, b: integer; var q, r: integer);
{ pre: b > 0 }
{ post: q = a div b and r = a mod b }
```

The condition after `pre:` should hold before procedure `P` is called (this is the caller's obligation). The condition after `post:` will hold after `P` has been called (this is the procedure's obligation).

---

[Tom Verhoeff](http://www.win.tue.nl/~tverhoeff/)  
[Eindhoven University of Technology](http://www.win.tue.nl/~tverhoeff/)