

DelftBlue Birthday Promotion

On **June 14th**, we celebrate the second birthday of our beloved supercomputer **DelftBlue**. With the arrival of the Phase 2 hardware in January 2024, it has reached new levels of capability and performance.

We want to celebrate this with a little challenge to all users of DelftBlue: Hand in the fastest **Sudoku solver** program running on DelftBlue **by June 21st, 23:59**, and you win an amazing prize: An exclusive project share on DelftBlue for one year, giving you high priority independent of what your colleagues in the research or education shares are doing!

The Sudoku Challenge

Write the fastest program to solve the following Sudoku puzzle on DelftBlue:

		1						2
			3		4	5		
6			7			1		
	4		5					
	2						8	
					6		9	
		5			9			4
		8	2		1			
3						7		

Figure 1: The 2024 Sudoku Challenge: input board.

What is a Sudoku puzzle?

A Sudoku puzzle consists of 9 blocks arranged in a 3×3 matrix. Each block consists of 9 fields, again arranged in a 3×3 matrix, see the figure above.

Starting from a configuration like this (with some numbers “filled in”, the sudoku is considered **solved** if and only if:

- Fields that contain a number in the starting configuration are unchanged.
- Every field contains an integer number between 1 and 9, inclusive.
- Every block of 9 elements contains every integer from 1 to 9 exactly once.
- Every row of 9 elements contains every integer from 1 to 9 exactly once.
- Every column of 9 elements contains every integer from 1 to 9 exactly once.

Your task

Write a program that solves the Sudoku shown in Figure 1

- as fast as possible

- using any programming language you want
- using any hardware available in DelftBlue

You are allowed to use modules from the software stack. If your solution relies on software developed by others, make sure to respect copyright and license conditions.

It must be possible to change the input board in the program and still get a correct solution, i.e., “hard-coding” the solved sudoku board is not considered a valid program.

The program should print (at least) in ASCII to the terminal (1) the initial sudoku-board shown in Figure 1, and (2) the solved sudoku board. Empty fields may be displayed as ‘0’, e.g.:

```
+-----+-----+-----+
| 0 0 1 | 0 0 0 | 0 0 2 |
| 0 0 0 | 3 0 4 | 5 0 0 |
| 6 0 0 | 7 0 0 | 1 0 0 |
+-----+-----+-----+
| 0 4 0 | 5 0 0 | 0 0 0 |
| 0 2 0 | 0 0 0 | 0 8 0 |
| 0 0 0 | 0 0 6 | 0 9 0 |
+-----+-----+-----+
| 0 0 5 | 0 0 9 | 0 0 4 |
| 0 0 8 | 2 0 1 | 0 0 0 |
| 3 0 0 | 0 0 0 | 7 0 0 |
+-----+-----+-----+
```

Submission

Send an e-mail ****before Friday, June 21st, 23:59**** with the title “DelftBlue birthday promotion” to **info-DHPC@tudelft.nl**. In the e-mail, report the “wallclock time” (aka “elapsed time”) required by your program as measured by a suitable timer function in your programming language (e.g., “time.perf_counter()” in Python, “omp_get_wtime()” from OpenMP, “std::chrono::high_resolution_clock::now()” in C++) or “tic()/toc()” in Matlab. You may start the timer after printing the initial board and before the solver algorithm begins, and stop it before you print the result.

As an attachment to your e-mail, add a **.zip**, **.tar**, or **.tar.gz** archive containing anything needed to reproduce your result on DelftBlue, but at least:

- A README.txt file explaining briefly the idea/algorithm/implementation as well as instructions on how to compile and run the program
- A job script containing all steps to build and run the example. If submitted on DelftBlue it should convincingly demonstrate that your code is correct and as fast as you claim. . .

By submitting a program, you agree to its publication on DHPC’s communication channels. Whether you are disclosed as the author is your choice.

The winner will be determined unanimously by a Jury consisting of the DHPC contact group based on (a) the time-to-solution achieved, and (b) the originality and elegance of the solution.