```
Courses Aalto 2023 Spring Nuance Log out Help
```

Aalto 2023

Index	Contest	Submissions	Pre	0	CP	1	2a	2 b	2c	3a	3b	4	5	9a	9c)	IS	4	6a	6b	9a
MF 1	2 9a	SO 4 5 6 F	9a	$\overline{\mathbf{x}}$	0a	0b	9a	9b)											

CP3a: fast solution with doubles ★★

Please note that you can still submit, but as the course is already closed, your submissions will not be graded.

To get started with the development, download the code templates, unzip the file, edit cp.cc, and run ./grading test or ./grading benchmark to try it out — see the instructions for more details!

Upload your solution as a file here
Please upload here the file cp.cc that contains your solution to task CP3a.
Choose File No file chosen
or copy-paste your code here
Submit

Your submissions

Your submissions to CP3a will appear here; you can simply reload this page to see the latest updates.

What you will need to do in this task

Please read the **general instructions for this exercise** first. Here are the additional instructions specific to this task:

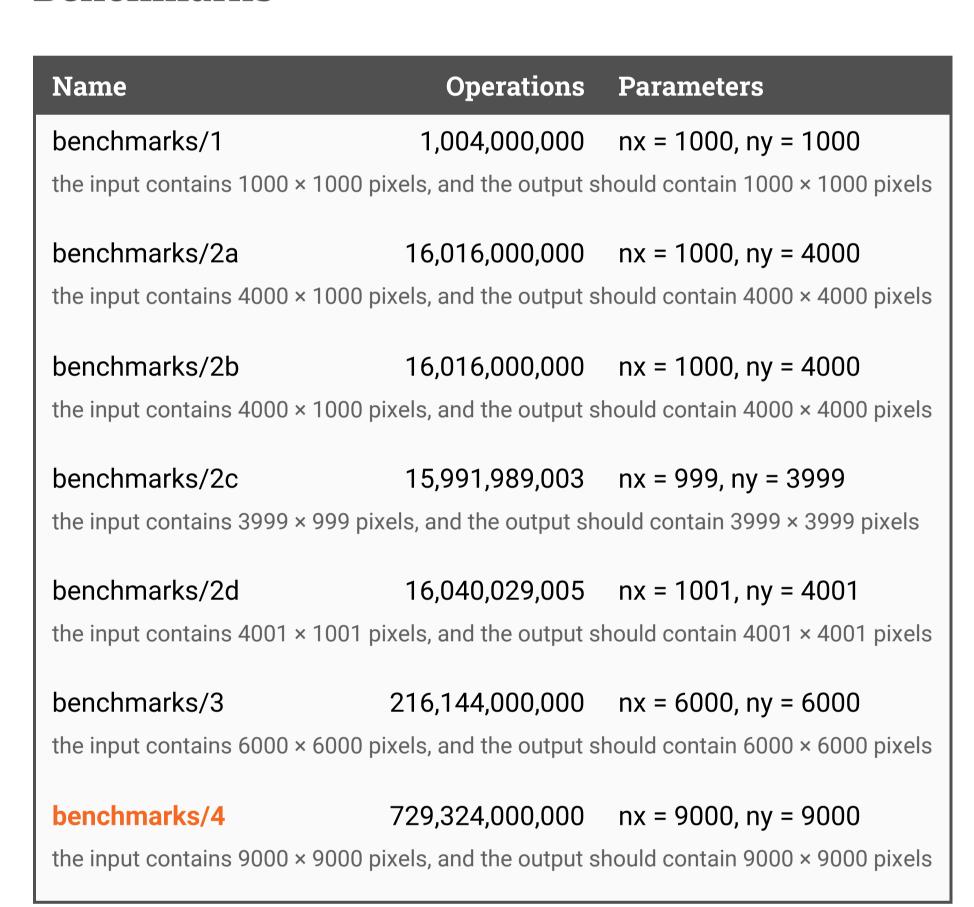
Using all resources that you have in the CPU, solve the task **as fast as possible**. You are encouraged to exploit instruction-level parallelism, multithreading, and vector instructions whenever possible, and also to optimize the memory access pattern. Please do all arithmetic with **double-precision** floating point numbers.

What I will try to do with your code

I will first run all kinds of tests to see that your code works correctly. You can try it out locally by running ./grading test, but please note that your code has to compile and work correctly not only on your own computer but also on our machines.

If all is fine, I will run the benchmarks. You can try it out on your own computer by running ./grading benchmark, but of course the precise running time on your own computer might be different from the performance on our grading hardware.

Benchmarks

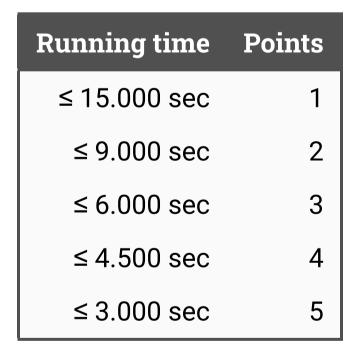


Here "operations" is our rough estimate of how many useful arithmetic operations you will **at least** need to perform in this benchmark, but of course this will depend on exactly what kind of an algorithm you are using.

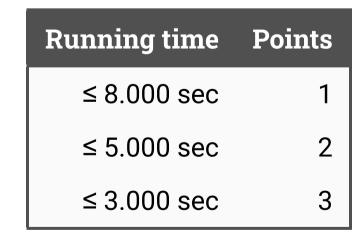
Grading

In this task your submission will be graded using benchmarks/4: the input contains 9000×9000 pixels, and the output should contain 9000×9000 pixels.

The point thresholds are as follows. If you submit your solution no later than on **Sunday**, **14 May 2023**, **at 23:59:59 (Helsinki)**, your score will be:



If you submit your solution after the deadline, but before the course ends on Sunday, 04 June 2023, at 23:59:59 (Helsinki), your score will be:



Contest

Your submissions to this task will also automatically take part in the **contest**, and you can receive **up to 2** additional points if your code is among the fastest solutions this year!

