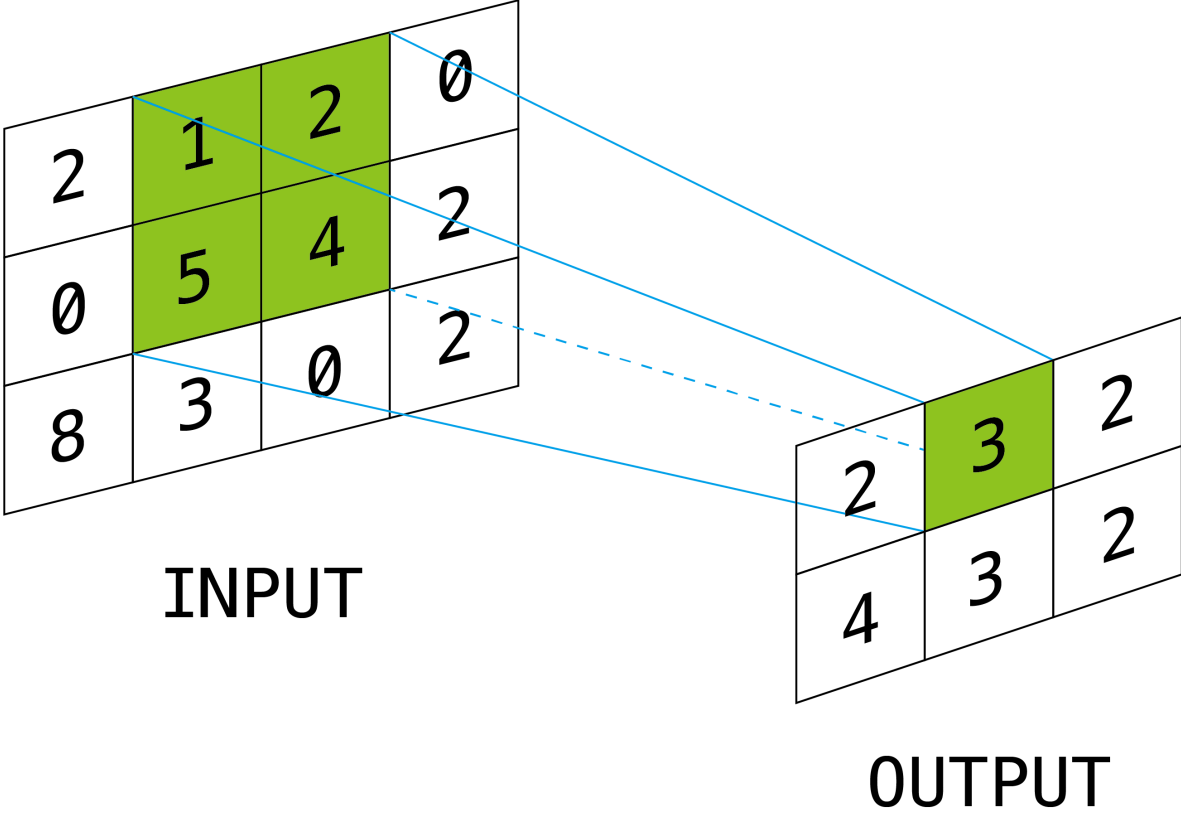


题目描述

卷积 (convolution) , 是一种被广泛应用于信号处理、图像处理、深度学习等领域的一种基本运算。在本题中, 我们要计算一个特殊的 $N \times N$ 的二维卷积。其过程是: 给定一个大小为 $(H + N - 1) \times (W + N - 1)$ 的矩阵, 经过计算后得到一个 $H \times W$ 的矩阵, 该矩阵的每一个元素都是原矩阵以该点为左上角的 $N \times N$ 的正方形中的元素的平均值。



$$(1+2+5+4) \div 4 = 3$$

输入/输出格式

输入为大小为 $(H + N - 1) \times (W + N - 1)$ 的内存中的单精度浮点矩阵, 矩阵中的每个元素都在 $[0, 1]$ 之间, 均为独立等概率随机生成。

该矩阵以一个 $(H + N - 1) \times (W + N - 1)$ 的数组的形式给出。保证 $5000 \leq W \leq 10000$, $5000 \leq H \leq 10000$, $2 \leq N \leq 10$ 。

为了方便运算, W 和 H 都保证是 8 的整数倍。

请你将结果的 $H \times W$ 的矩阵输出到一个长度为 $H \times W$ 的一维数组中。你的答案和标准答案之间的差的绝对值在 10^{-5} 以内被认为正确。

请不要在程序中向标准输出流输出任何信息, 否则程序会被判为0分。

评分方式

本题共有 10 个测试点, 每个测试点 10 分。

设 t_i 为选手程序运行第 i 个测试点的时间； c_i 为选手程序计算第 i 个测试点的正确性，1 为正确，0 为不正确； T_i 为本次比赛所有选手的程序中，能够正确计算第 i 个测试点的最短用时。选手在第 i 个测试点的分数 s_i ，由如下公式给出：

$$s_i = c_i \times \left(1 + 9 \times \frac{T_i}{t_i} \right)$$

选手本题的得分是各测试点得分之和，满分共计 100 分。

对于10%的数据， $N = 3$ ；对于20%的数据， $N = 5$ ；对于30%的数据， $N = 7$ ；对于40%的数据， $N = 10$ 。

解题框架

本题我们提供解题框架。

C/C++ 解题框架

你需要在 `solve.c` 中实现 `solve` 函数。

具体地，在所给的框架中，包含以下文件和文件夹：

- `Makefile`：用于编译，生成可执行文件，并执行测试。运行 `make` 来编译并执行测试。
- `solve.c` | `solve.cpp`：你所需要填写的代码文件，提交时仅能提交该文件。
 - 你需要实现 `solve` 函数。`output` 是一个已经分配好的长度为 $H \times W$ 的数组。请将计算出来的矩阵放在 `output` 中。
 - 你可以在该文件中添加你可能要用到的变量、函数和其他代码。
- `simd.h`：CPU的 SIMD 指令的封装。
- `solve.h`：`solve` 函数声明。
- `main.c` | `main.cpp`：该文件负责评测。该文件中包含了 `solve_naive` 函数。该函数提供了一个可用于参考的 `solve` 函数的实现。

提示

- 选手可以使用多线程来加速程序的性能。
- 选手可以使用 CPU 的向量指令来加速程序性能，`simd.h` 文件里包含了选手可能使用的一些向量指令的封装。