

第1部分实验室分配：C中的矩阵处理库

该任务的目标是在C语言中实现一个矩阵处理库，提供各种矩阵操作，如换位、矩阵-矩阵乘法、标量乘积等。为此目的，您将得到一个头文件矩阵.h包含库API提供的不同函数的定义（原型），您的任务是在单个C源文件matil.c中实现这些函数。为了部分地验证您的实现，我们为您提供了一个以使用库并执行各种完整性检查的程序形式存在的测试套件。

您可以下载一个存档与库头文件，一个空的矩阵.c源文件，以及测试套件源和支持文件在这里：
<https://olivierpierre.github.io/comp26020/lab/comp26020-lab1.zip>.

矩阵处理库

矩阵.h头定义了一个数据结构，表示一个矩阵和一组函数，允许创建和销毁矩阵对象，对矩阵执行操作，并将矩阵作为文件保存在磁盘上/从磁盘上加载。

矩阵数据结构。数据结构matrix_t表示一个矩阵。它有3个字段：两个整数记录矩阵的行和列的数量，以及一个整数指针的指针。一个二维数组)，持有矩阵的内容。创建一个矩阵对象是一个有两个步骤的过程：必须首先动态分配二维数组，然后填充矩阵的内容。

导出的函数。库的头导出一个由13个函数组成的API，以便在源文件中实现。它们可分为以下类别：

*矩阵分配、初始化和破坏：*矩阵分配用于初始化一个矩阵对象的维数，无矩阵自由分配一个对象。
matrix_init_rand、matrix_init_zeros和matrix_init_identity分别初始化已分配的具有随机整数、零和与单位矩阵对应的内容的矩阵对象的内容¹。

*Subab矩阵运算²：*矩阵_等于检查两个矩阵是否相等。矩阵_sum执行两个矩阵的和。矩阵的乘积将一个矩阵乘以一个标量。矩阵换位转换一个矩阵，矩阵乘积执行两个矩阵的乘积。

将/加载矩阵保存为文件：矩阵_dump_file将一个矩阵对象保存到一个文件中，
matrix_allocate_and_init_file分配一个矩阵对象并从一个文件中加载其内容（这些文件的格式如下讨论）。

您应该检查matial.h中的注释，了解关于每个函数预期行为的更多细节，特别是关于函数应该返回什么值的一些信息，以及它们是否对需要预先分配的矩阵对象进行操作。

您可以自由地在矩阵.c中开发其他函数。例如，在标准输出上显示矩阵的print_matil函数对于调试很有用。

磁盘上的矩阵文件格式。矩阵文件格式简单且基于文本：矩阵的每一行都用文件中的一行表示，构成该行的数字用这一行上的空格分隔。您可以在存档的矩阵样本/目录中找到矩阵文件的示例。

理想情况下，矩阵文件加载函数应该能够处理遵循该格式的非严格定义的文件，e. g. 在一行上的数字之间的空格数量可变，可能被忽略的空行，等等。

¹ https://en.wikipedia.org/wiki/Identity_matrix

² [https://en.wikipedia.org/wiki/Matrix_\(mathematics\)#Basic_operations](https://en.wikipedia.org/wiki/Matrix_(mathematics)#Basic_operations)

测试套件

基本测试套件。为了测试您的代码，您将得到一个在源文件中实现的基本测试套件，即基本测试套件。c。它是一个C程序。它包含一个主要的函数，其中包括矩阵。h。它创建矩阵对象，并通过执行完整性检查来测试库的API的每个函数。这个程序使用了统一的方法³测试框架，并需要编译3个额外的源文件：unity.c、ininty.h和ininty_内部文件.h。该测试程序可以按照关于模块化编译的讲座中所述进行编译。测试套件应该会驱动您的实现：我们鼓励您研究基本测试套件的内容。c来理解要实现的库的函数应该如何表现出来。该测试套件由一系列测试用例组成，每个测试用例都使用RUN_TEST（）宏从main中调用，并在一个自包含的函数中定义。当测试用例所包含的其中一个断言失败时，该测试用例将会失败。例如，当条件的计算结果为false（C中为0）时，断言TEST_ASSERT_TRUE（<条件>）将失败。参见Unity网站⁴以获取对测试套件中使用的其他断言的引用。要完成作业，你不需要理解统一的内容。c，统一。h和unity_ininals.h。这些文件实现了Unity引擎，而且相当复杂。

高级测试套件。基本的测试套件并不全面，尽管通过它包含的所有测试意味着完成了大量的分配，实现看起来有功能，但这并不意味着一切都是完美的。在标记时，除了基本的测试套件外，还将使用一个扩展的测试套件，因此您无法访问测试的详细信息。这个套件将包括其他的功能测试，还将检查实现的稳健性，以防止在使用库API时所犯的错误。要准备这个高级测试套件，请尝试推理：

基本测试套件中可能缺少的功能测试；

如何在C程序中使用库，以及在使用其API时可能出现哪些编程错误，例如尝试分配一个负维数的矩阵，或者尝试从错误格式的文件加载矩阵。

在这里，健壮性意味着您的实现应该努力检测API滥用，并在它发生时采取适当的操作。

可交付成果、提交日期和截止日期

这个分配有一个可交付的内容：已完成的矩阵.c文件。该提交是通过CS部门的Gitlab进行的。您应该有一个名为“26020-lab1-s-matim <您的用户名>”的存储库的分叉。通过将文件推到主分支上并创建一个名为lab1提交的标签来表明提交已准备好标记，提交您的交付成果。**制造肯定的你推向那个清晰的存储库和不另一个一和向加标签于你的投降**正确地说，不这样做很可能导致这个练习的分数为0。

您不需要提交库头文件或任何测试套件的文件。这意味着，在进行分配时，尽管您可以为了调试目的而编辑这些文件，但在提交时，您的代码最终应该可以使用这些文件的未修改版本。

这次任务的截止日期是伦敦时间2022年11月11日下午6点。

标记方案

练习将为10，使用以下标记方案：

该程序功能良好，并通过了基本的测试套件/4

该程序通过了高级测试套件/4

该程序遵循课程中关于动态内存的涵盖的良好C编程实践
C标准库函数的分配和使用/2

³ <https://github.com/ThrowTheSwitch/Unity>

⁴ <https://github.com/ThrowTheSwitch/Unity/blob/master/docs/UnityAssertionsReference.md>