

ACM班 & 致远工科荣誉编程综合实践上机考试

PPCA 2021 Round 1

B 卷

题目名称	图像压缩
题目类型	提交 <code>.hpp</code>
内存限制	1024 MB
测试点时限	10.0 秒
测试点数目	10
评分规则	-

注意事项：

- 此卷只有一道题目，你不应该期望在这道题目获得满分。
- 建议考试时间 5 小时，并且请在考试结束前将你的代码提交到我们的评测机上进行评测。
- 建议使用和评测机环境相同的 Linux 编程环境。
- 不得在考试期间使用自己的手机、电脑等电子产品，所有操作必须由机房电脑完成。
- 严禁抄袭，包括但不限于查看/复制自己考试前的提交代码，使用各大代码共享平台的代码。

你可能需要用到：

- 评测机：<https://oj.yukisaki.cn/>
- 模拟器：<http://82.156.197.136/>
- C++参考手册中文版：<https://zh.cppreference.com/>
- C++参考手册英文版：<https://cppreference.com/>
- `#include <bits/stdc++.h> & using namespace std;`

图像压缩

题目描述

K是二次元。打开相册，眼睛里就会倒映出一片花花绿绿。但是很快他就意识到苍白的储存介质已无力满足他的饕餮，而压缩欲望在K仅存不多的自我认知看来实在太过虚幻。于是他决定请你动用技术帮助维护他的粮仓，毕竟扬汤止沸总比锅鼎溢裂好一些。**图像的压缩率**在他看来是首要的，因此这也是评判你的压缩算法的第一标准。当然，谁也不愿意见到本来具象的食粮像自己的魂灵一样变得模糊不清，你的算法不应粗劣到可以破坏K心中神圣的美感，所以**压缩后图像的质量**将是你助人为乐的底线。

图像数据以RGB三通道的形式给出（准确的说是BGR的顺序），每一通道是 $0 \sim 255$ 的整数。假如一张图片的像素是 $m \times n$ ，则数据是 `unsigned char [m][n][3]`。具体可见下发文件。

你需要完成的图像压缩算法分成 `encode` 和 `decode` 两个步骤：

- 在 `encode` 阶段，你需要压缩原始图像数据，以你希望的占用更小的形式储存到磁盘上，其大小和原始数据大小的比率即为**压缩率**；
- 在 `decode` 阶段，你需要储存在磁盘上的压缩数据重新读出来，重新转化成RGB三通道的形式，这就是压缩后的图像数据。

有许多指标可以衡量压缩后图像的质量。其中比较淳朴的是**峰值讯噪比**（Peak signal-to-noise ratio, PSNR），它简单而有效地使用**均方误差**（MSE）来衡量图像的质量。在实现你的压缩算法时，你其实不需要特别清楚PSNR的机理。

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [Original(i, j) - Compressed(i, j)]^2$$
$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right)$$

以下是维基百科上的一些标准：

- PSNR接近 50dB，压缩后的图像仅有些许非常小的误差。
- PSNR大于 30dB，人眼很难察觉压缩后和原始影像的差异。
- PSNR介于 20dB 到 30dB 之间，人眼就可以察觉出图像的差异。
- PSNR介于 10dB 到 20dB 之间，人眼还是可以用肉眼看出这个图像原始的结构，且直观上会判断两张图像不存在很大的差异。
- PSNR低于 10dB，人类很难用肉眼去判断两个图像是否为相同，一个图像是否为另一个图像的压缩结果。

输入格式

注意：这里描述的输入格式是 `judge.cpp` 的输入格式

第一行有两个整数 m, n 表示图片的大小是 m 行 n 列的。

接下来的 $m \times n \times 3$ 行，每行一个 $0 \sim 255$ 的整数，表示图片的通道信息。

每 3 行描述了一个像素点的颜色，每 $3 \times n$ 行描述了一行的颜色信息。

输出格式

注意：这里描述的输入格式是 `judge.cpp` 的输出格式

除了标准输出信息之外，`judge.cpp` 会输出一个名为 `out.txt` 的文件，这里保存的是图片经过你的压缩以及解压之后得到恢复之后的图片通道数据，你有两种方法将其可视化：

- 把它上传到我们的 [可视化工具](#) 中观察解压之后的图片质量。
- 使用 `python rev.py` 可以自动读取 `out.txt` 并将其还原成 `save.png` 的图片。

数据范围

时间限制：10000 ms

空间限制：1024 mb

本题一共有 10 测试点，均来自正常尺寸的图片，大小不超过 1920×1080 。其中的 5 个测试点的数据已经下发。你可以通过修改 `rev.py` 或者把输入文件上传到 [可视化工具](#) 中进行查看。

评测标准

如果你希望用自己的图片来调试，`cast_into_data.py` 可以将图片（png, jpg等格式都支持）生成输入的 `in` 文件。

提交时，请将 `compress.hpp` 内的所有内容粘贴至提交框。

我们根据你的还原之后图片的质量 $PSNR$ 以及你的压缩率来进行给分。对于某一张 `q Bytes` 的图片，你的压缩大小为 `p Bytes`：

- 如果 $PSNR \in [30dB, \infty)$ 那么你的得分就是 $q - p$
- 如果 $PSNR \in [20dB, 30dB)$ 那么你的得分是 $\lfloor 0.8(q - p) \rfloor$
- 如果 $PSNR \in [10dB, 20dB)$ 那么你的得分是 $\lfloor 0.6(q - p) \rfloor$
- 如果 $PSNR \in [0dB, 10dB)$ 那么你的得分只有 $\lfloor 0.2(q - p) \rfloor$

你的总分是所有测试点分数之和。

小提示

如果你想要追求极致的分数，希望能够通过二进制读写的方式来压缩这张图片，你可以访问 [中文手册](#) 或者 [英文手册](#) 来查阅你想要的函数。

你不应该查阅其他任何的网站，包括 **OJ** 内部你曾经写过的代码、百度、CSDN、StackOverflow 等等。

在 `compress.hpp` 中，你不应该定义任何的全局变量。