PPM P6 Image Reader

Given a PPM (Portable Pixmap) file in P6 format (binary), read the image data and return it as a 3-dimensional List. PPM P6 files store RGB color images in a compact binary format.

Your task is to parse the PPM P6 file and return the image data as a List with shape (W, H, C) where H is height, W is width, and C is the number of color channels (3 for RGB).

Input Format

A filename string representing the path to a PPM P6 format image file.

Output Format

A 3d **list** with shape (W, H, 3) where:

- W: Width of the image in pixels (integer)
- H: Height of the image in pixels (integer)
- 3: Number of color channels (Red, Green, Blue)

The list should contain integers in the range [0, 255] representing the RGB color values.

Constraints

- The input file is a valid PPM P6 format file without any comments
- The maximum color value is 255
- The image dimensions ($W \times H$) are positive integers

File Format

The PPM P6 file format consists of:

- **Line 1:** Magic number ”P6” indicating binary PPM format
- **Line 2:** Two integers separated by whitespace: width (W) and height (H)
- **Line 3:** Maximum color value (255 in this problem)
- **Line 4 (Remaining bytes):** RGB pixel values in row-major order as raw binary data. Each pixel consists of three bytes (Red, Green, Blue) with values in the range [0, 255]

The pixel data contains $W \times H$ pixels, each with 3 color values (RGB), for a total of $3 \times W \times H$ bytes.

Sample Input

```
img.ppm
```

Sample Output

```
# "img_full.ppm" saved sucessfully
```

Scoring

- You will receive 40 points if the output image matches the expected output image order regardless of the shape of the output list.
- You will receive an additional 60 points if the output list has the correct shape (W, H, 3).

Implementation

1. download attachment(`[Files]`)
2. unzip the downloaded file
3. Fill in the `get_image` function in `solution.py`.
4. make sure that all of the files are in the same directory.

Warnings

- You are **NOT ALLOWED** to use the `numpy` library (except in the main block.)
- You are **NOT ALLOWED** to use the `struct` library.

Hints

- PPM P6 files are binary files, so it will be opened in binary mode using the `rb` flag like ‘`wav_header`’ problem.
- use `int.from_bytes()` method to convert bytes to integer.
- How many bytes is needed to represent each channel in the pixel (range [0, 255])?
- PPM P6 endianess is big-endian.

Expected Output Image

You should see the full ~~meme~~ summary of this course after running the main block in `solution.py`. The next page shows the expected output image.

2102208 Prog EE 2025:
Part JSS, SSK:
EE108
Linear alg
numpy
IEEE754
Big O
TAs
When you finally understand recursion:
Part SAK:
TAs, SAK
EE108
Part SAK
TAs, SAK
EE108
Part TSL:
EE108
TSL
Part MWS:
?
Solve the problem
to unlock Part MWS