

Write a `mean_fits` function that takes a list of FITS files as an argument, reads them in, and returns the mean image data of the FITS files. All the images have the same dimensions and your calculated mean array should match those dimensions.

Your function should be able to process an arbitrary number of files. If you're unsure how to do that, take a look at the sample solutions to the previous problems.

The mean stack of these files will be a large 200 x 200 array, so we will only look at the central value of your returned array, which is where we expect the pulsar to be.

For the files `image0.fits`, `image1.fits` and `image2.fits`, your program should work like this:

```
>>> mean_fits(['image0.fits', 'image1.fits', 'image2.fits'])[100, 100]
0.017356586332122486
```

We can access the mean at `x=100, y=100` using the index operator right after the function call, as we expect its return value to be an array. The central mean value for the files `image0.fits`, `image1.fits`, and `image3.fits` should be:

```
>>> mean_fits(['image0.fits', 'image1.fits', 'image3.fits'])[100, 100]
0.01006323037048181
```

Using all the FITS files provided (images 0 to 4), your program should work like this:

```
>>> mean_fits(['image0.fits', 'image1.fits', 'image2.fits', 'image3.fits', 'image4.fits'])[100, 100]
0.014150320738554
```