Informatics for Astronomers - WS2020

Roland Ottensamer, Marina Dütsch, Miguel Verdugo, Gerald Mösenlechner

Exercise sheet 6 - Python errors, functions and C

The following will be also part of the assessment:

(1) Try to present exercises in a way that everyone can understand (even those who didn't do the exercises), so please explain the vital parts of your solution in a clear way.

(2) Try to also include some background information where applicable, and/or explain the possible context/motivation for the given exercise.

---

1. Please find out a list of built-in exceptions (errors) in python and their meaning.

   For a few, show (with examples) when and how they occur.

2. Write a python function that takes a one-dimensional integer array of arbitrary length and a float that represents a percentage (it should have a default value of 10%).

   The function should remove all duplicates from the array and return the number of removed entries and the resulting array. If the percentage of removed numbers is bellow the given percentage, an error shall be raised.

   The input array can be generated with e.g `np.random.randint(value, size=N)`, where `value` is the upper limit of the possible values and `N` the number of random values to be generated.

3. Take the .txt file `Photometry_example.txt` from moodle. The file contains data of an 200×200 pixels image of a star. Load the image using `numpy.genfromtxt()`.

   1. Use `matplotlib.pyplot.imshow()` to properly plot the image. Save it as jpg image.
   2. Write a function that takes the array and find the position of the star returning the (x, y) coordinates (the coordinates of the maximum) and the peak flux.

4. Write a simple C function that takes three integer numbers and returns the largest number.

5. Write Example 3.2 as a C function. The function shall read the file using the `fopen()` and `fscanf()` function and write the content to an array (you can use either a 1D or 2D array). You can allocate the memory of the array directly (e.g. `int array[200][200]` or `int array[200*200]`) or with a pointer and the malloc function. Note that you only need to print the result (if you want you can use a struct to return all asked values).