TESTS

General features

The input is saved in a `.inp` file. The expected output is saved in a `.txt` file. The real output is written in a `.out` file. Both outputs files are compared using the function diff of bash.

There is no error handling in this programs, so if the input is not as wanted, everything will fail unexplicably.

Format for test_point

The following functions can be called from the `.txt` file:

> `add_point` `remove` `assign` `+=` `+` `-=` `-` `*=` `^` `*` `==` `clockwise` `X` `Y` `quad` `slope` `quadrant` `distance`

After the name of the command, the necessary arguments should be listed in order.

If a command returns a point it is printed following this sintax: `x y` (x space y). The result is printed with a precision of 2 decimals after the comma.

If the command returns a boolean, either yes or no is printed.

If it returns a number, the number is printed.

Every result is finished with an endline. As an example the file `test_point.inp` should produce the output `test_point.txt`.

Format of test_conv

The functions provided are:

> `polygon` `remove` `assign` `+` `*` `bbox` `isInside` `num_vert` `num_edges` `area` `isRegular` `centroid` `Hull`

Each one is associated to one of the methods of the class. As in point, you first write the function and then the arguments. For more detail look at `test_conv.inp` and its expected output `test_conv.txt`.

Format of test

This format is the one provided by Jordi Petit in the page of the project. However, the example there is incomplete, when an error occurs, this are the possible messages:

0. error: command with wrong number or type of arguments
1. error: undefined identifier
2. error: not enough parameters
3. error: unrecognized command
4. error: wrong format

And for the warnings:

0. warning: not enough parameters
1. warning: nothing to do or show

Format of test_time

This one isn't compared to anything. It just shows in each line the process it has done, the number of vertices it had, and the time it needed to do so.