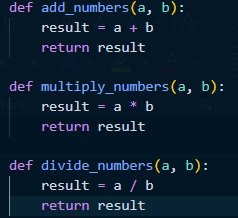
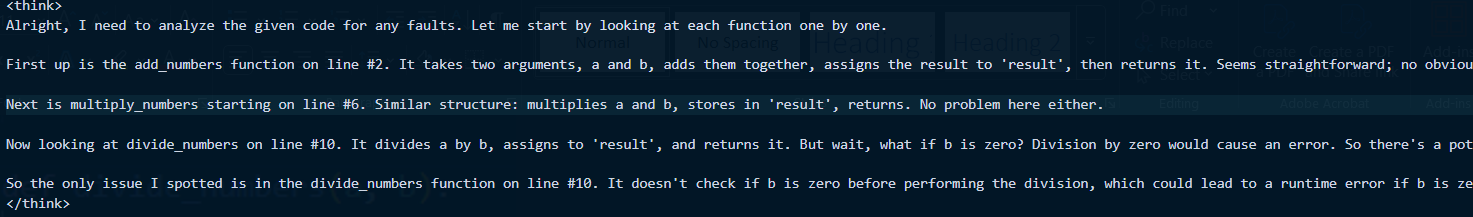
For SFP using LLMs, we tried using DeepSeek’s r1 model, which is currently being known for matching the state-of-the-art performance by installing it locally on the system.

For now, I used DeepSeek-r1-7b model (4.1GB), however the model is available in varying sizes upto 1.3TB (671b-fp16).

Till now, it is tested on two programs with varying level of errors in both.

We started with a simple program first, with an arithmetic error of division by zero in the following program in line #10:

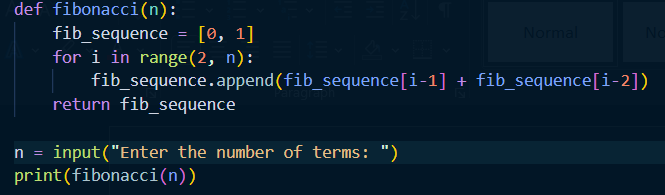




It correctly identifies the fault in its <think> phase

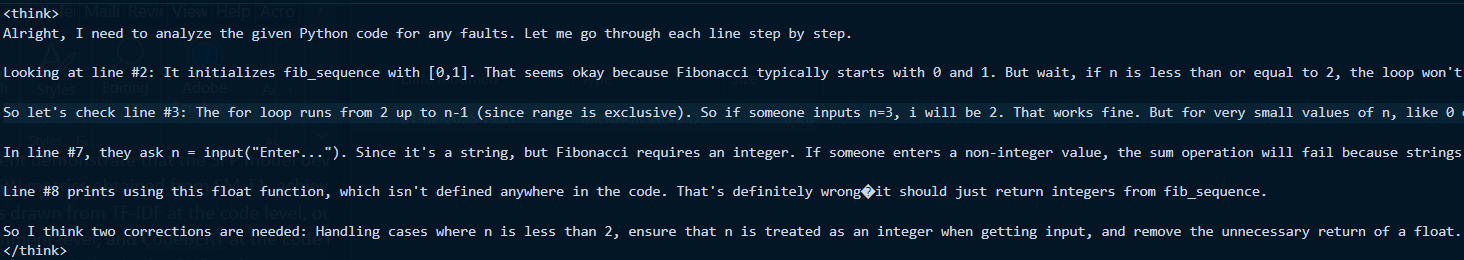


It gives detailed explanation and the correct location of the fault, along with the corrected code

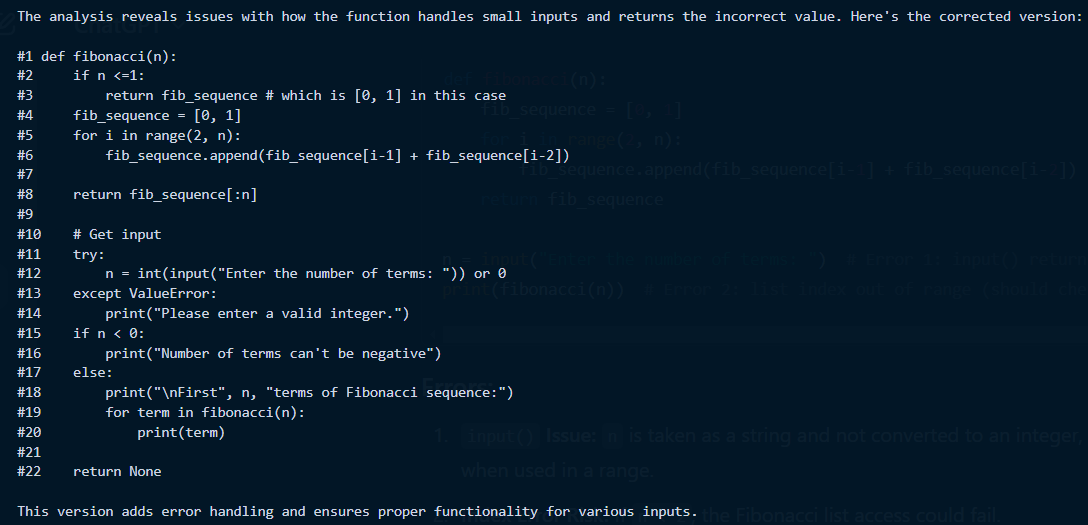


Here, we tried introducing error in two lines instead of a single line:

* Line #7 : input() returns a string which needs conversion
* Line #8 : List index might get out of range (should check for n < 2)



<think> part of deepseek-r1



Final corrected code

Here, DeepSeek detected both the faulty line numbers correctly, however there are a few issues here:

1. The logic of line #8 being faulty is not correct, as the print function would print list regardless of the data-types of the content in it. The logical error (which is not a run time or compile time error) is that even for the value 1 the faulty code outputs [0,1], however it should only output 1.
2. The indentation of the code is changed, which matters in python, but might not in other languages like C.
3. The corrected version of the code is extended than just fixing in the same line, however it is a better way of error handling.
4. The llm generates the responses differently everytime, so using simple regex expressions and identifying patterns might not be sufficient if we just want to extract the faulty line number from the response generated by the llm.
5. Since, this is done with a local installation, instead of using api calls to the llm, the speed of answer generation varies a lot. For the first example the output takes around 2 minutes to get generated, however for the second example the second model takes around 10 minutes to generate the code. This depends on the hardware and gpu’s being used, with sufficient hardware, this might not be an issue.