Alvaro Andres Pinzon Cortes Follow
May 5, 2022 · 2 min read · Listen

Creating simple zero-knowledge verifier contract with ZoKrates (0.7.13) solidity (0.8.0)

Prerequisites

- Install docker

Step by step guide

1. Create a folder named **code** to put the code

2. Create the square.zok file with this content:

```
def main(private field a, field b):
  assert(a * a == b)
  return
```

3. Start docker

4. Open the terminal run this command:

```
docker run -v <PATH_TO_FOLDER>:/home/zokrates/code -ti
zokrates/zokrates:0.7.13 /bin/bash
```

In my case that PATH_TO_FOLDER is were I created the square.zok file which is:

```
docker run -v /Users/andi/Downloads/UdacityGithubNeu-
master/zokrates/code:/home/zokrates/code -ti
```

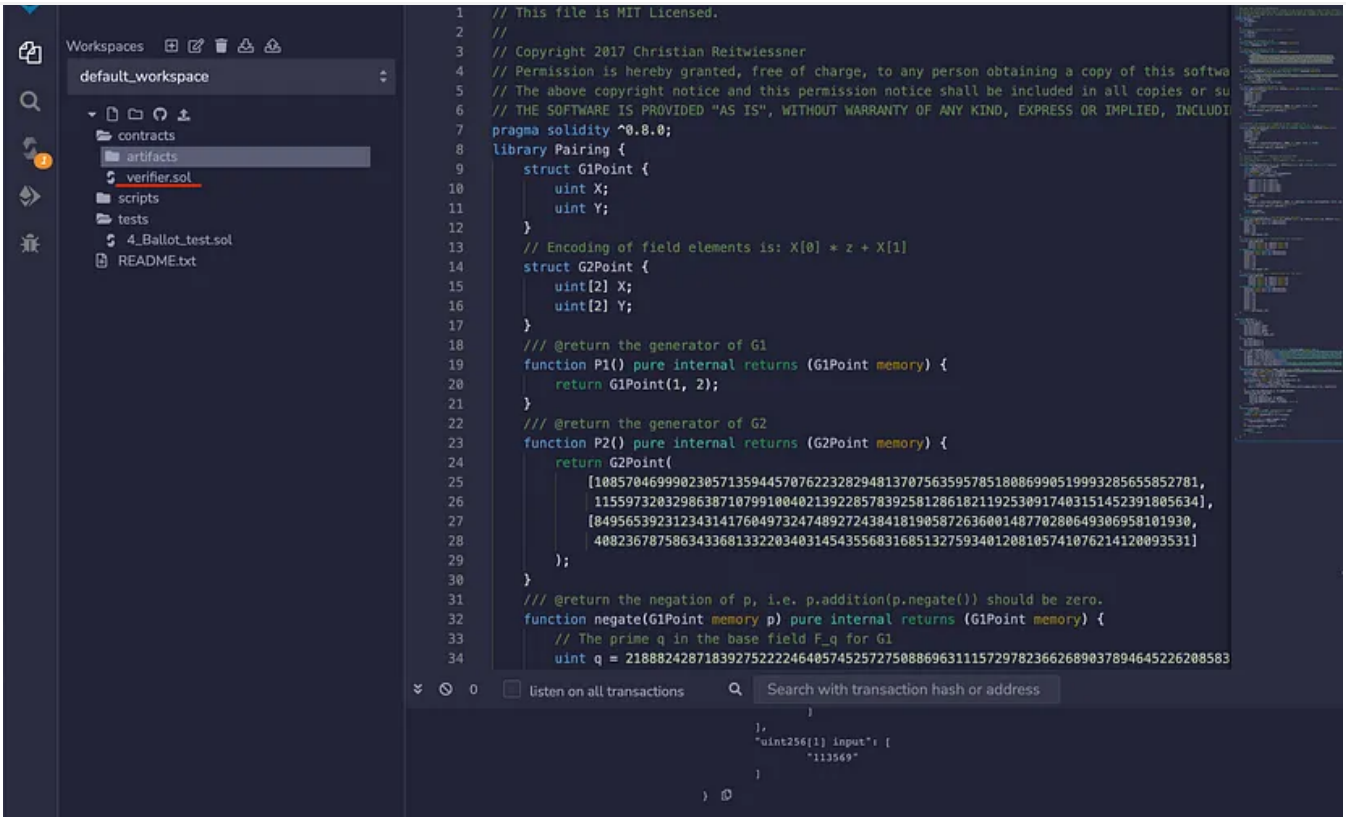
```
zokrates/zokrates:0.7.13 /bin/bash
```


5. Change directory to c

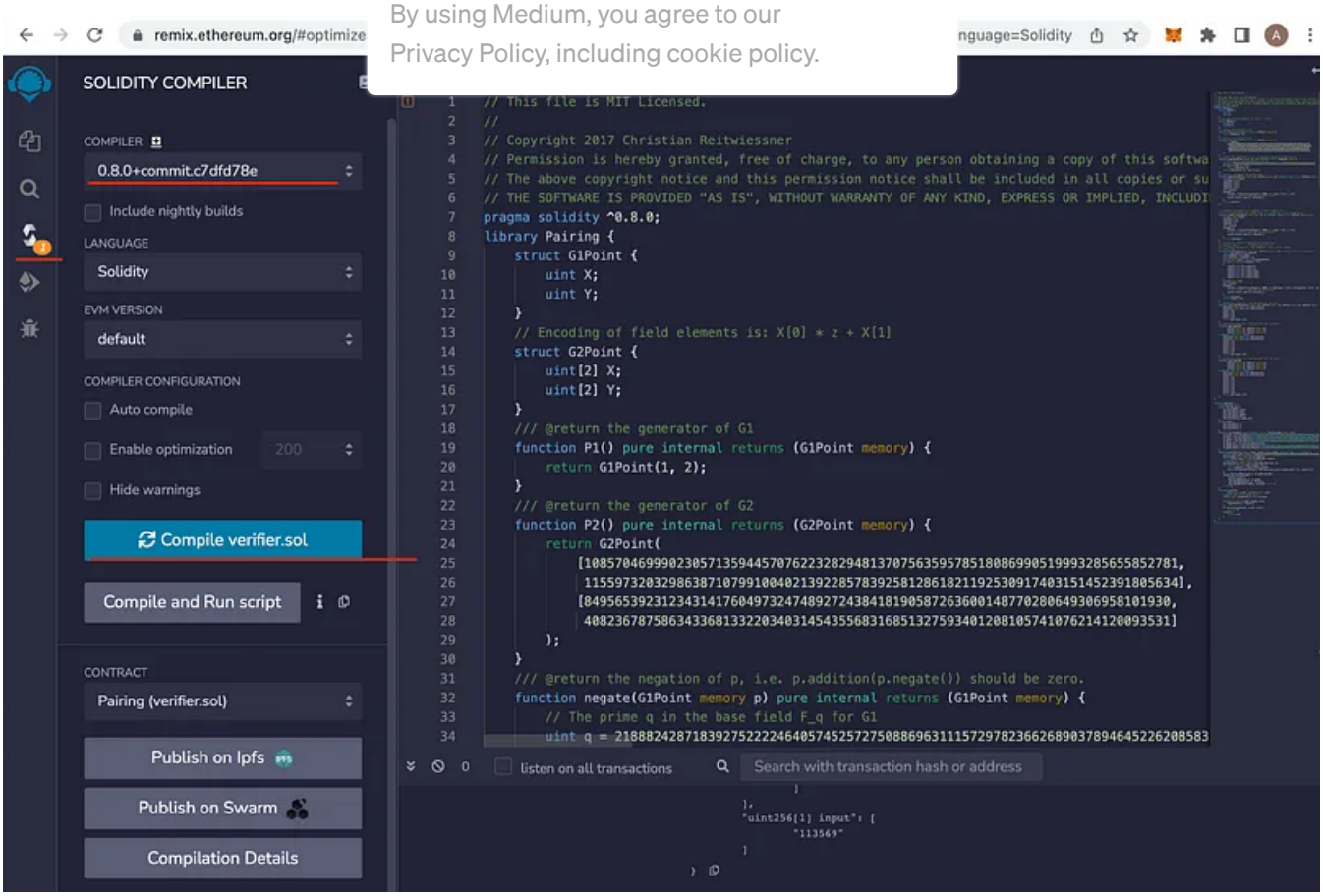
```
cd code
```

6. Run this commands to create the proof and verifier

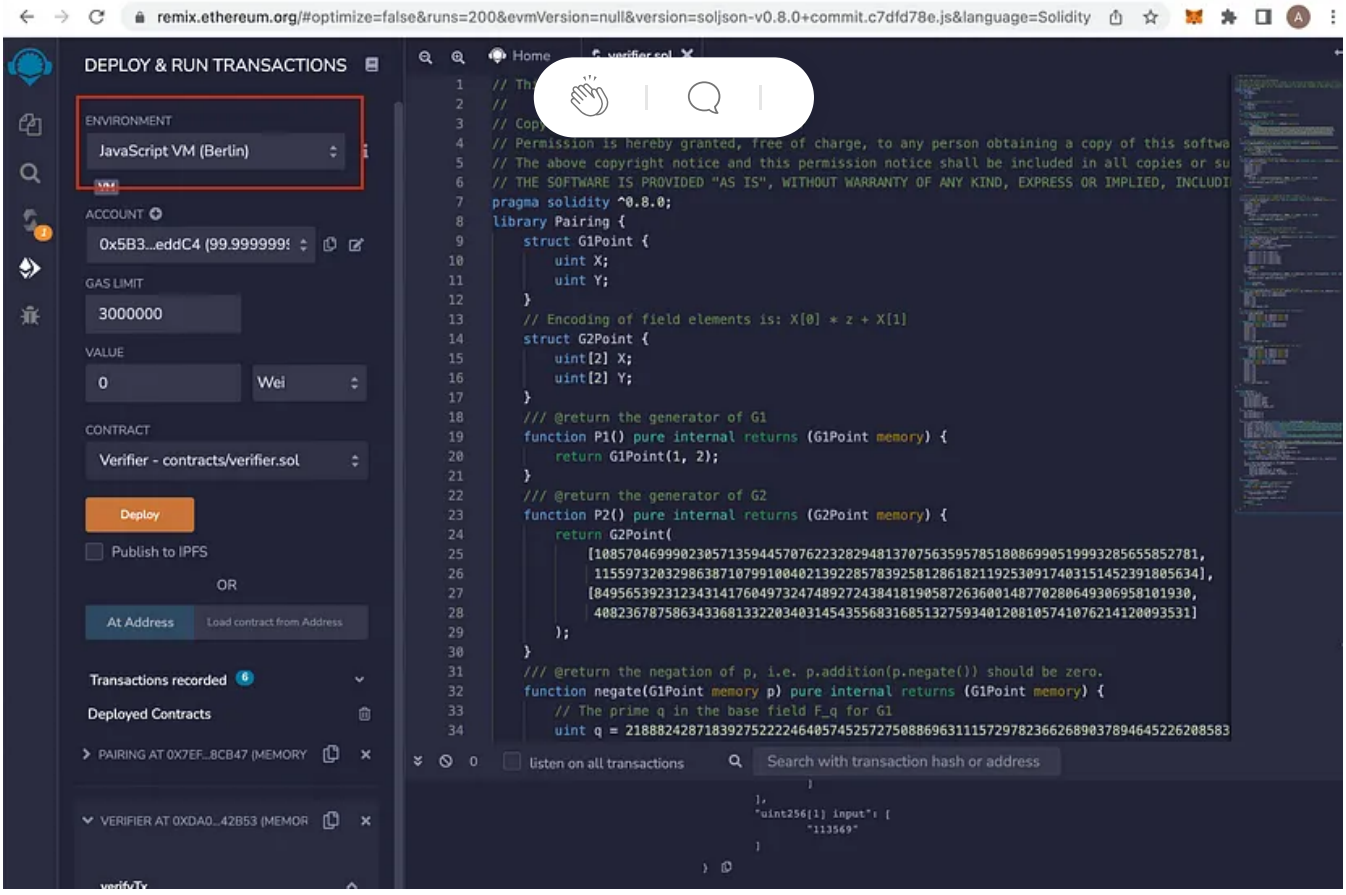
```
# compile
zokrates compile -i square.zok
# perform the setup phase
zokrates setup
# execute the program
zokrates compute-witness -a 337 113569
# generate a proof of computation
zokrates generate-proof
# export a solidity verifier
zokrates export-verifier
```

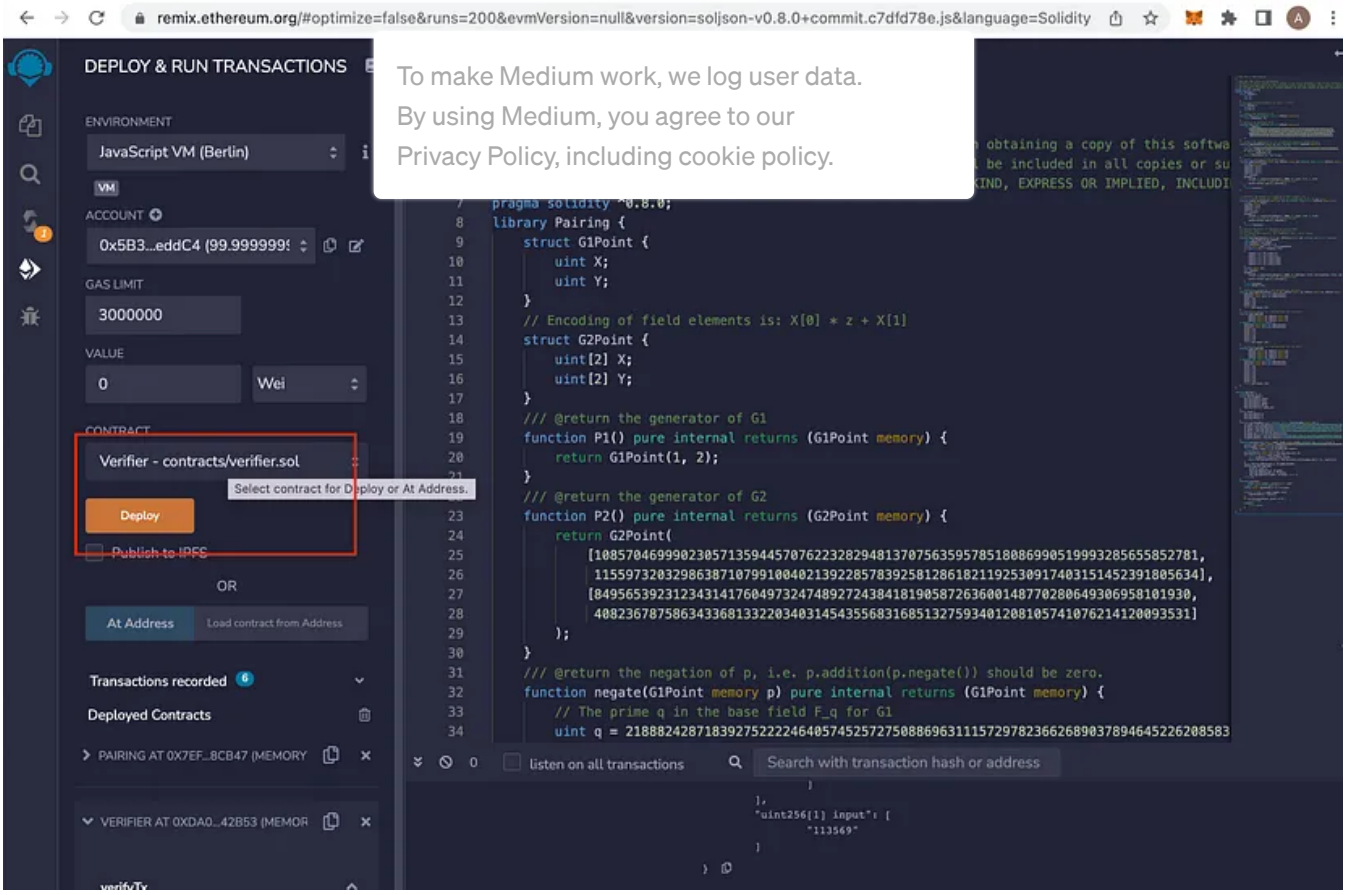



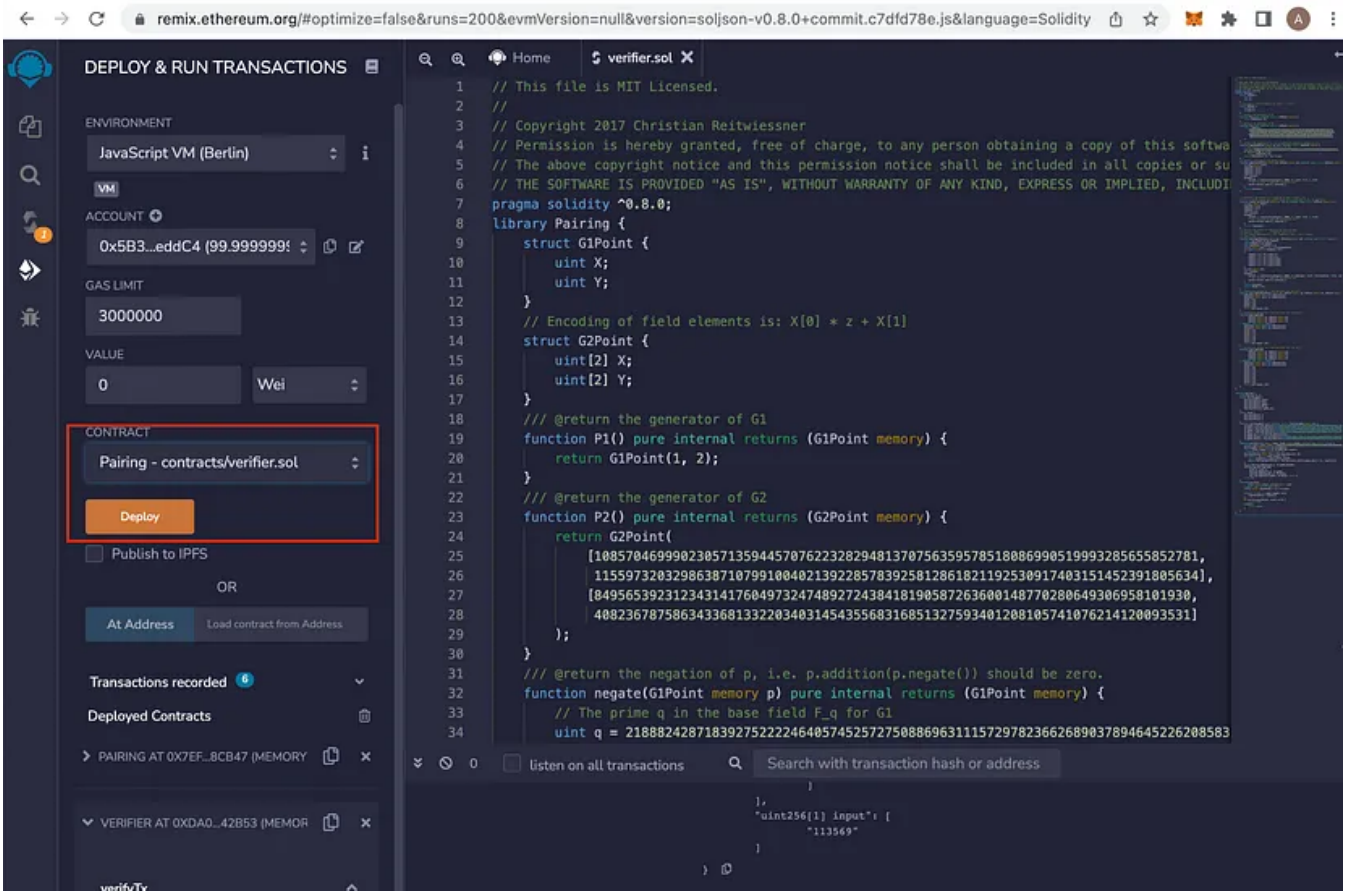
7. Compile



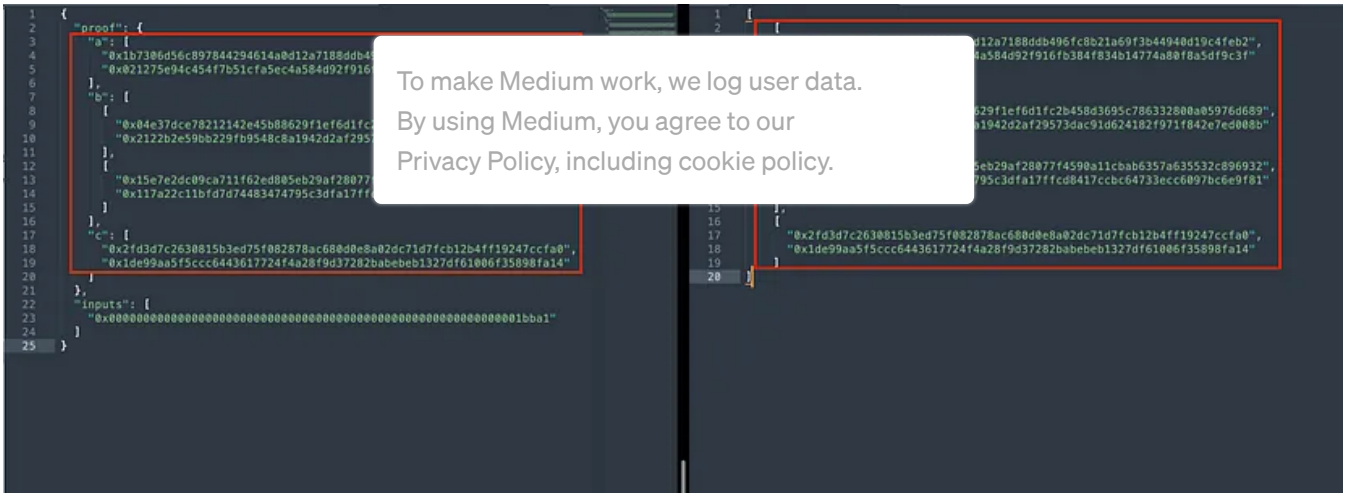
8. Deploy





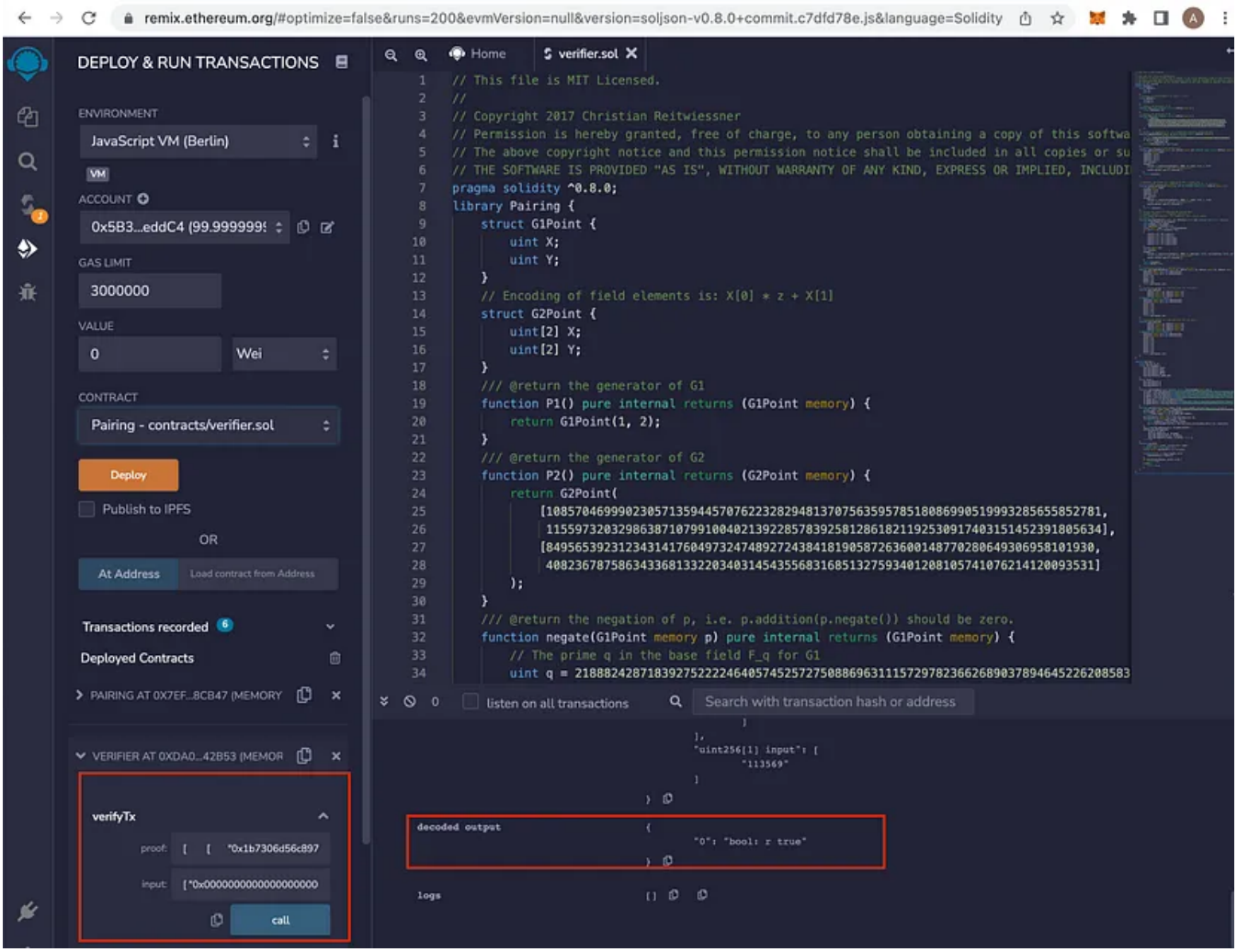


9. The function verifyTx requires an array because structs in solidity map to arrays in the ABI:



Make sure to encapsulate the proof inside [ ] just like in the image.

10. Call the verifyTx function:



Solidity Udacity Zokrates