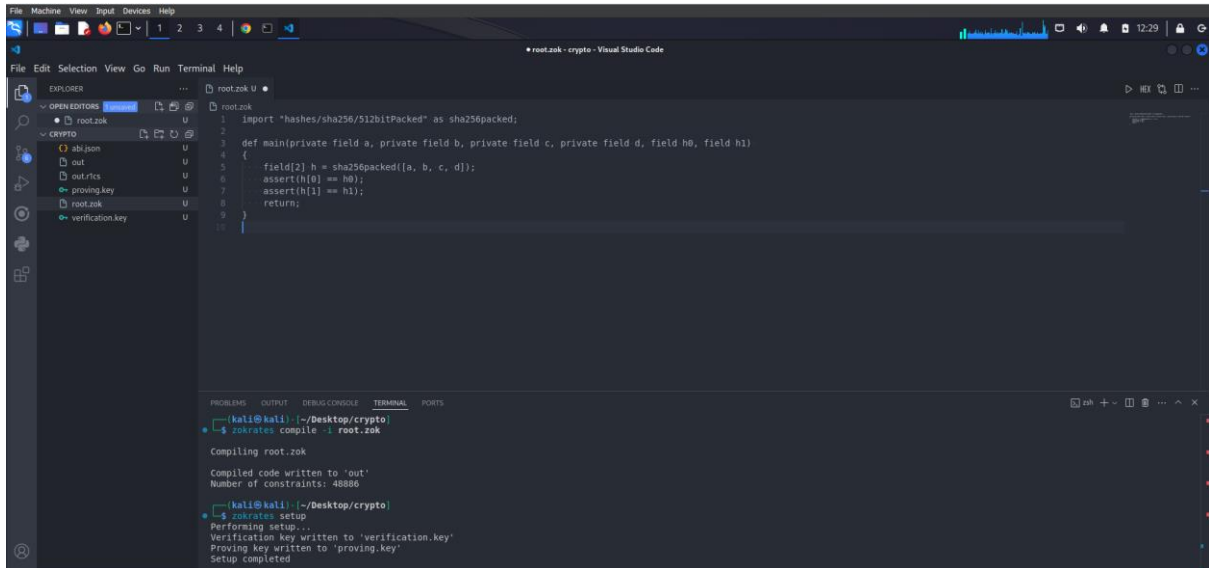


Cryptography

section – 1

1. Compiling root.zok



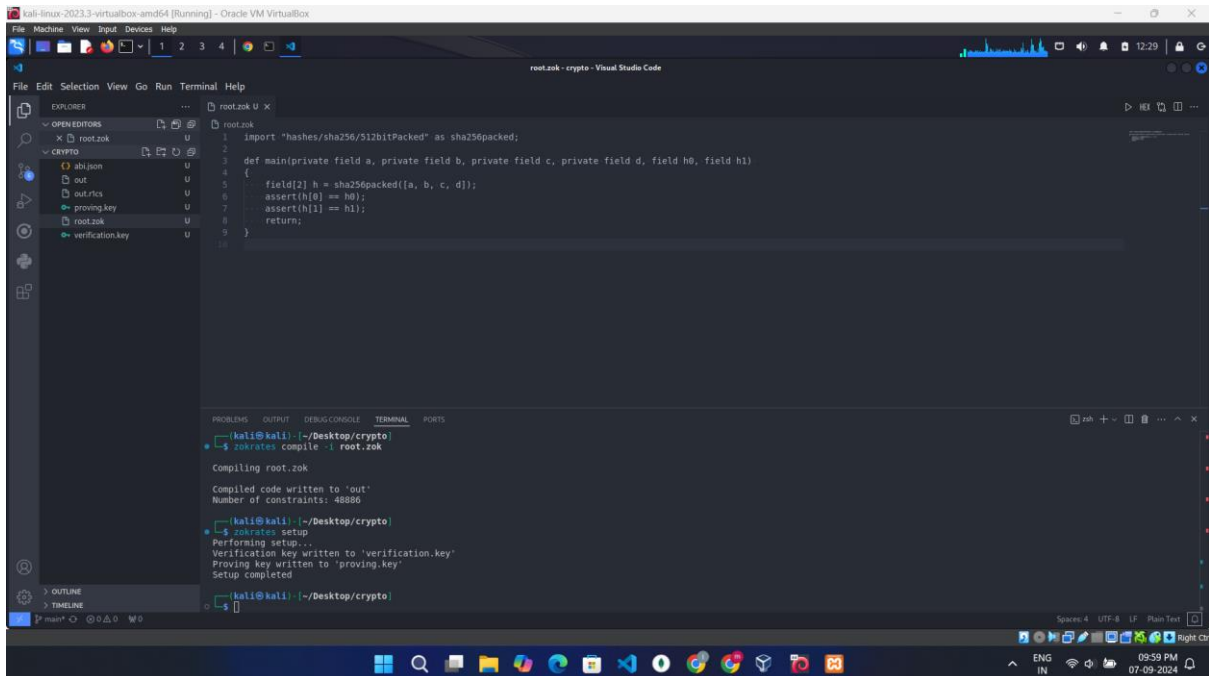
```
File Machine View Input Devices Help
root.zok - crypto - Visual Studio Code

EXPLORER
root.zok
  abi.json
  out
  out.rcts
  proving.key
  root.zok
  verification.key

root.zok
1 import "hashes/sha256/512bitPacked" as sha256packed;
2
3 def main(private field a, private field b, private field c, private field d, field h0, field h1)
4 {
5   field[2] h = sha256packed([a, b, c, d]);
6   assert(h[0] == h0);
7   assert(h[1] == h1);
8   return;
9 }
10

TERMINAL
(kali@kali) ~/Desktop/crypto
$ zokrates compile -i root.zok
Compiling root.zok
Compiled code written to 'out'
Number of constraints: 48886
(kali@kali) ~/Desktop/crypto
$ zokrates setup
Performing setup...
Verification key written to 'verification.key'
Proving key written to 'proving.key'
Setup completed
```

2. setup root.zok



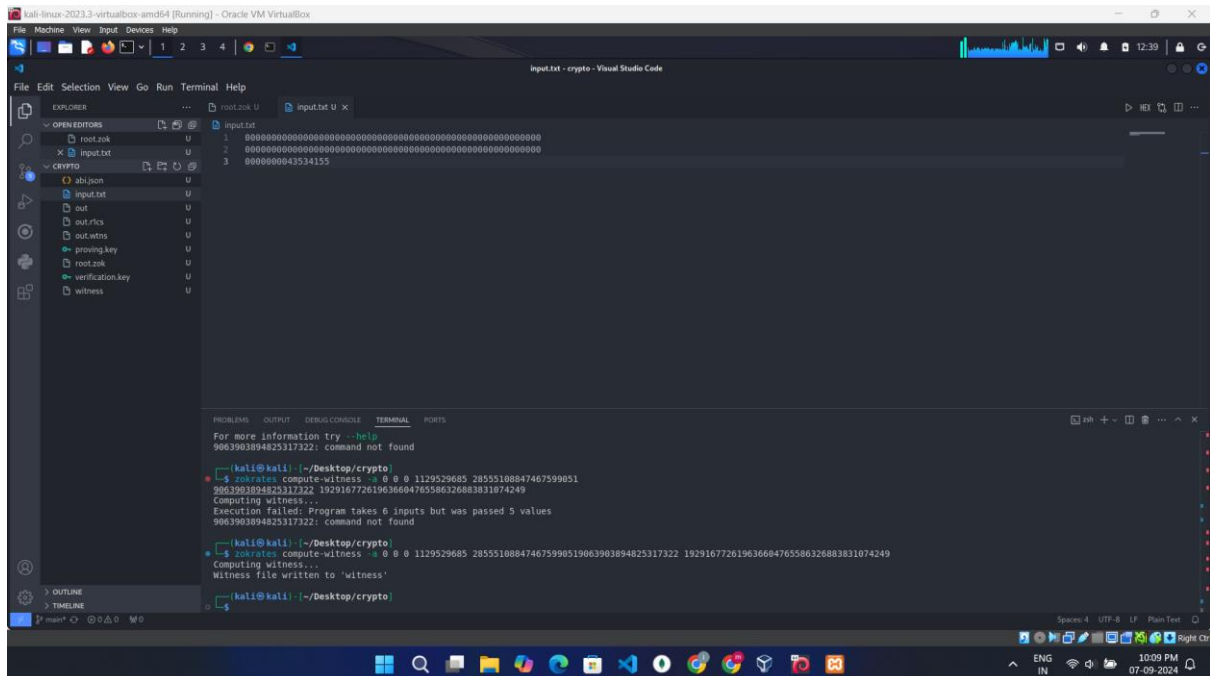
```
File Machine View Input Devices Help
root.zok - crypto - Visual Studio Code

EXPLORER
root.zok
  abi.json
  out
  out.rcts
  proving.key
  root.zok
  verification.key

root.zok
1 import "hashes/sha256/512bitPacked" as sha256packed;
2
3 def main(private field a, private field b, private field c, private field d, field h0, field h1)
4 {
5   field[2] h = sha256packed([a, b, c, d]);
6   assert(h[0] == h0);
7   assert(h[1] == h1);
8   return;
9 }
10

TERMINAL
(kali@kali) ~/Desktop/crypto
$ zokrates compile -i root.zok
Compiling root.zok
Compiled code written to 'out'
Number of constraints: 48886
(kali@kali) ~/Desktop/crypto
$ zokrates setup
Performing setup...
Verification key written to 'verification.key'
Proving key written to 'proving.key'
Setup completed
$
```

3. Giving input to root.zot



4. Generating proof and verifying in root.zot

