**CS572-Project -1**

**Submitted to**

**Prof. David Brennan**

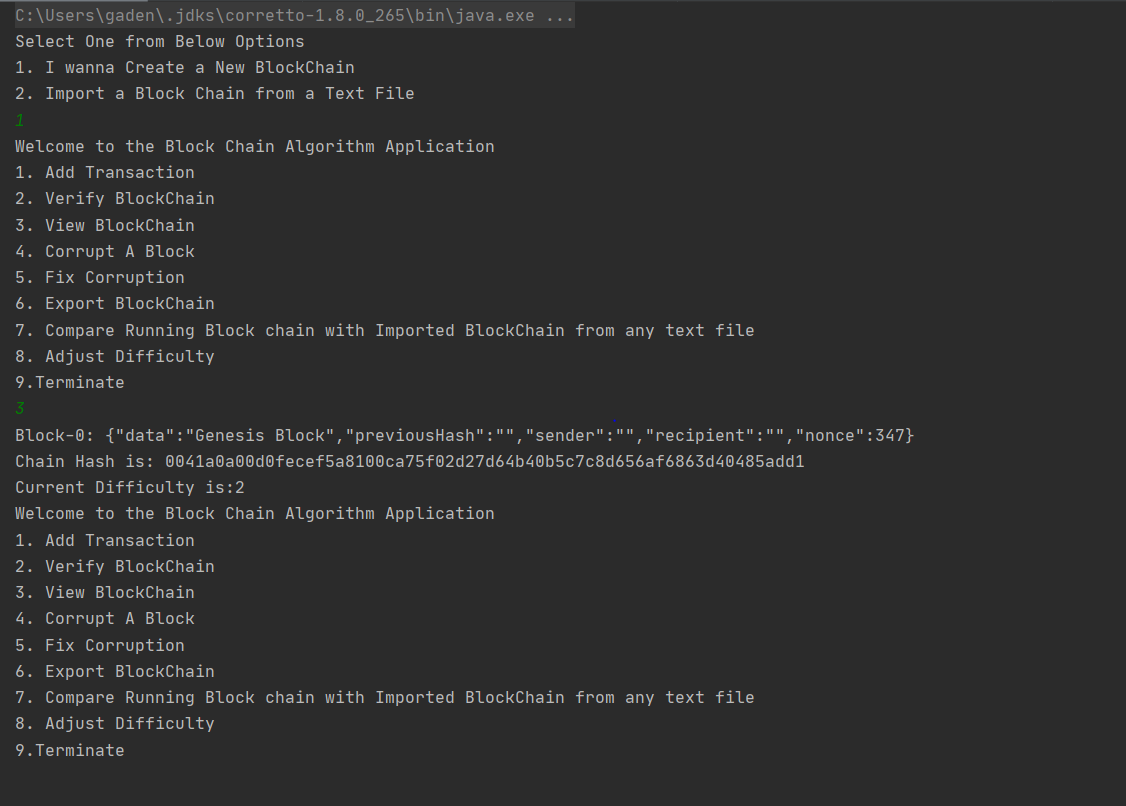
**Submitted by**

**Nihal Reddy Gade**

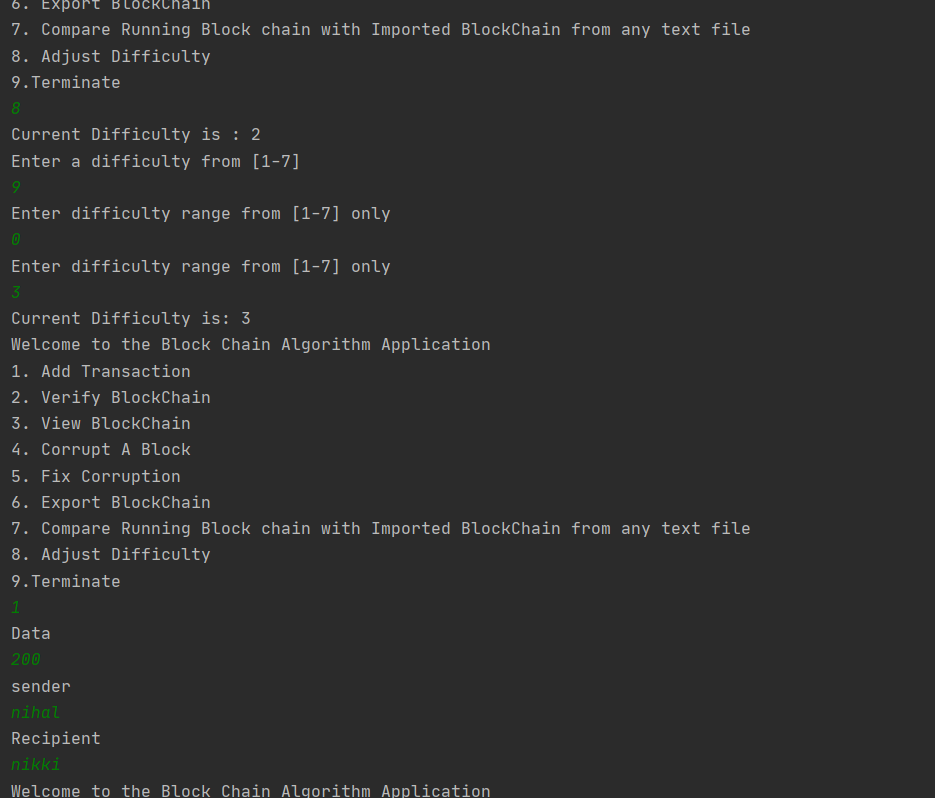
**#310363**

**Attempting operations on all 9 options:**

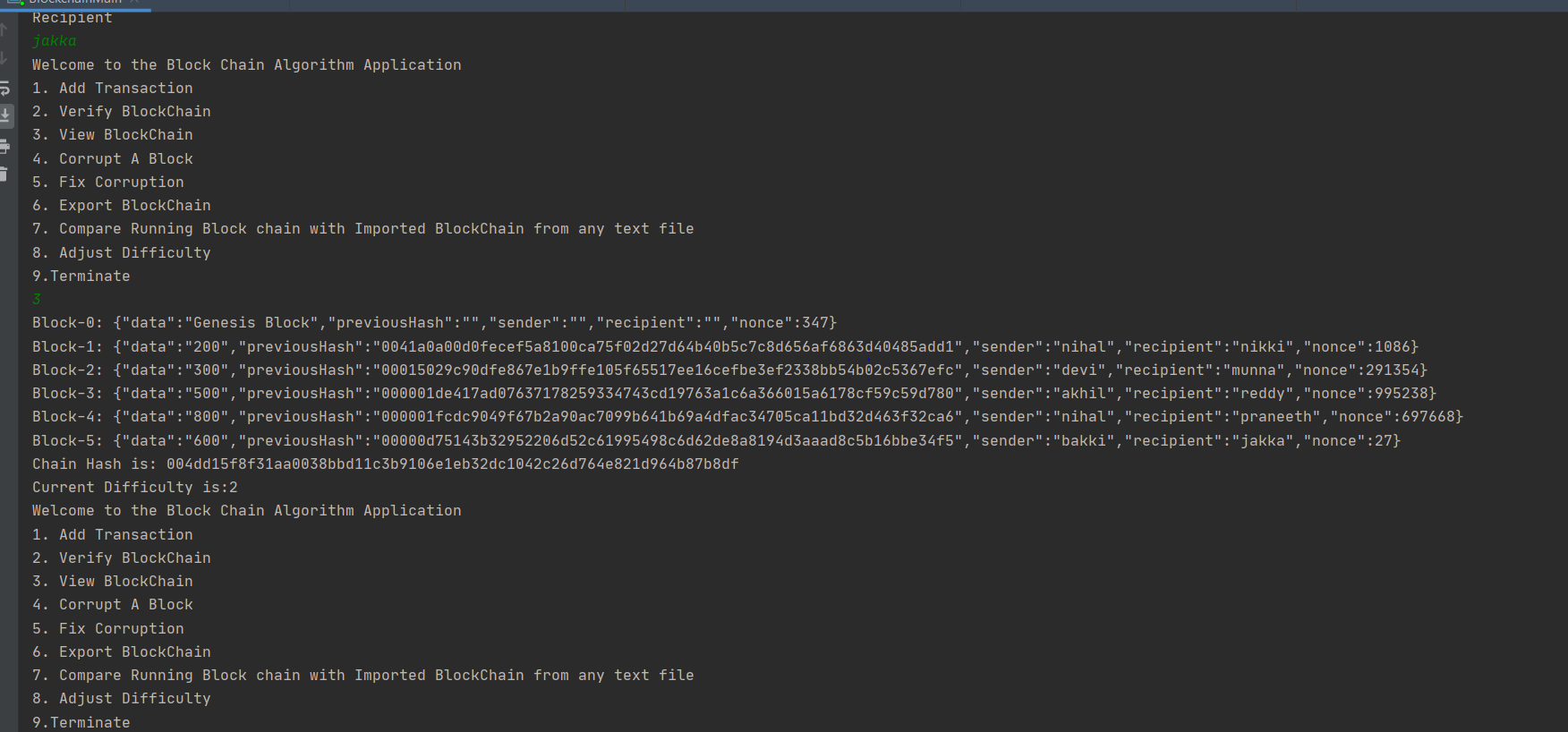
* First, I will create a new block chain and all the UI is shown in below screenshots in logical and understandable format.
* When I click on option1, Genesis block is called first and we can view it by clicking on option 3. View Blockchain.



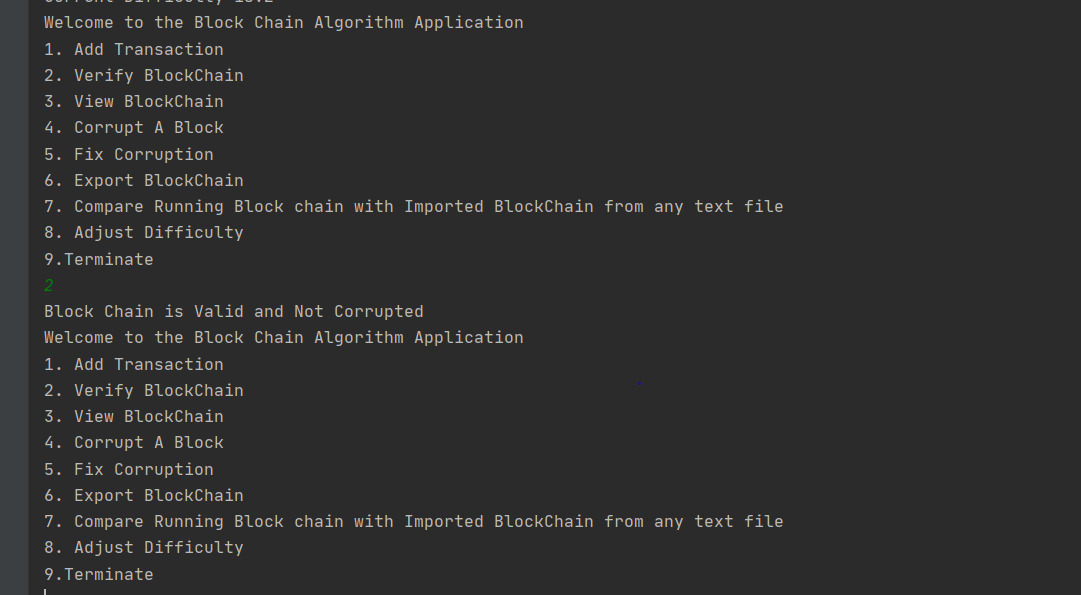
* Now Let us add few more blocks. We can change the current difficulty [1-7] by using option 8. Adjust Difficulty. If user enters difficulty out of range from [1-7] he will be asked again and again.
* I will add three more blocks and check the view option to see everything went well.



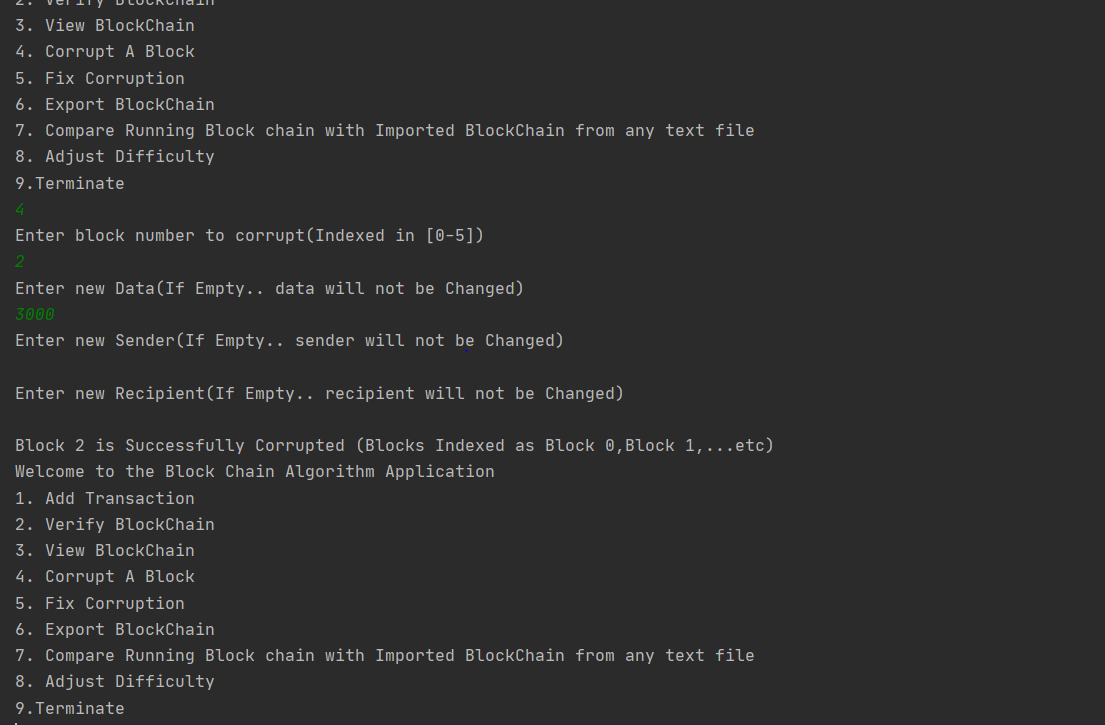
* After adding few blocks and by Viewing we have Blocks[0-5]



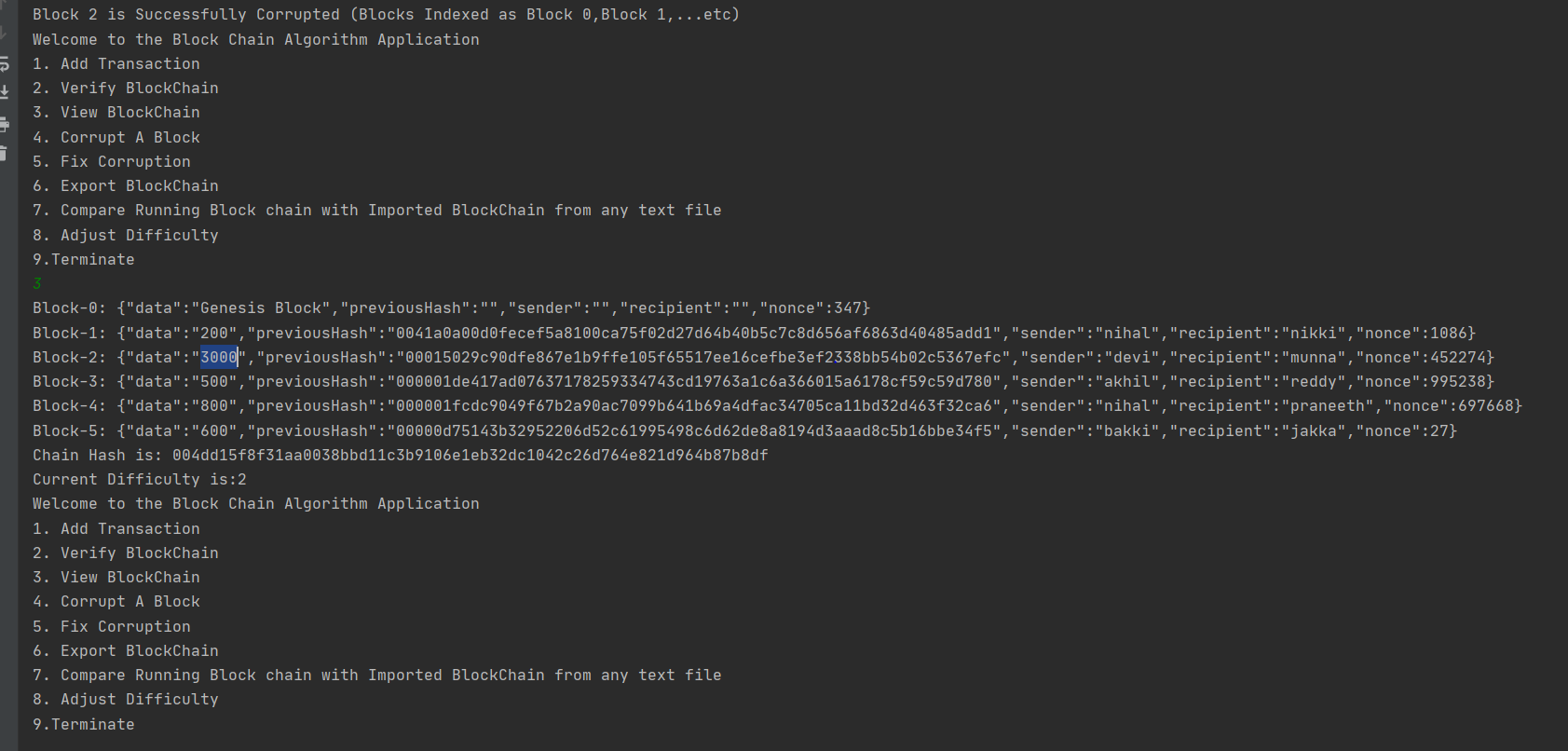
* Now if we verify the block by option 2. Verify Blockchain. We get its valid and not corrupted.



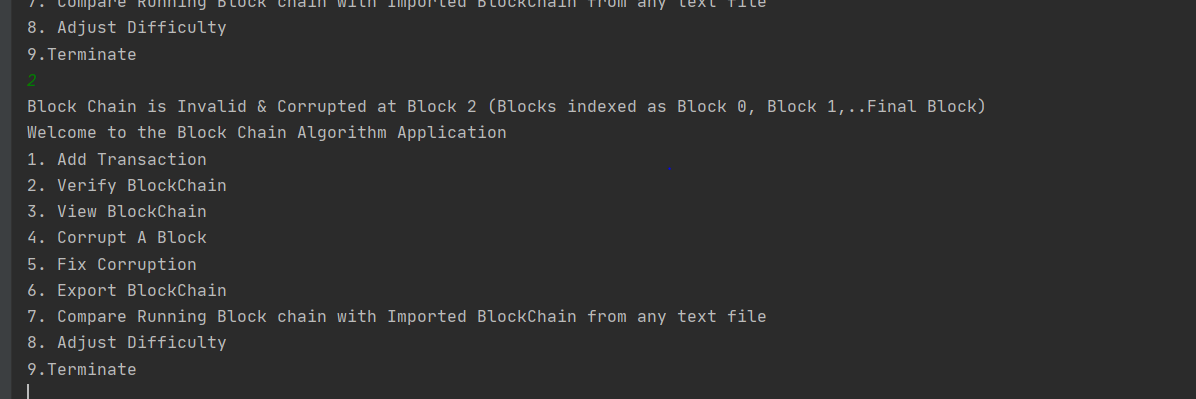
* Now Let us Corrupt it by option 4. Corrupt A Block. I will corrupt Block 2 indexed from 0.
* We can change data, sender and recipient values if any field is not changed it will be fixed as previous value.
* I am only changing data value for now.



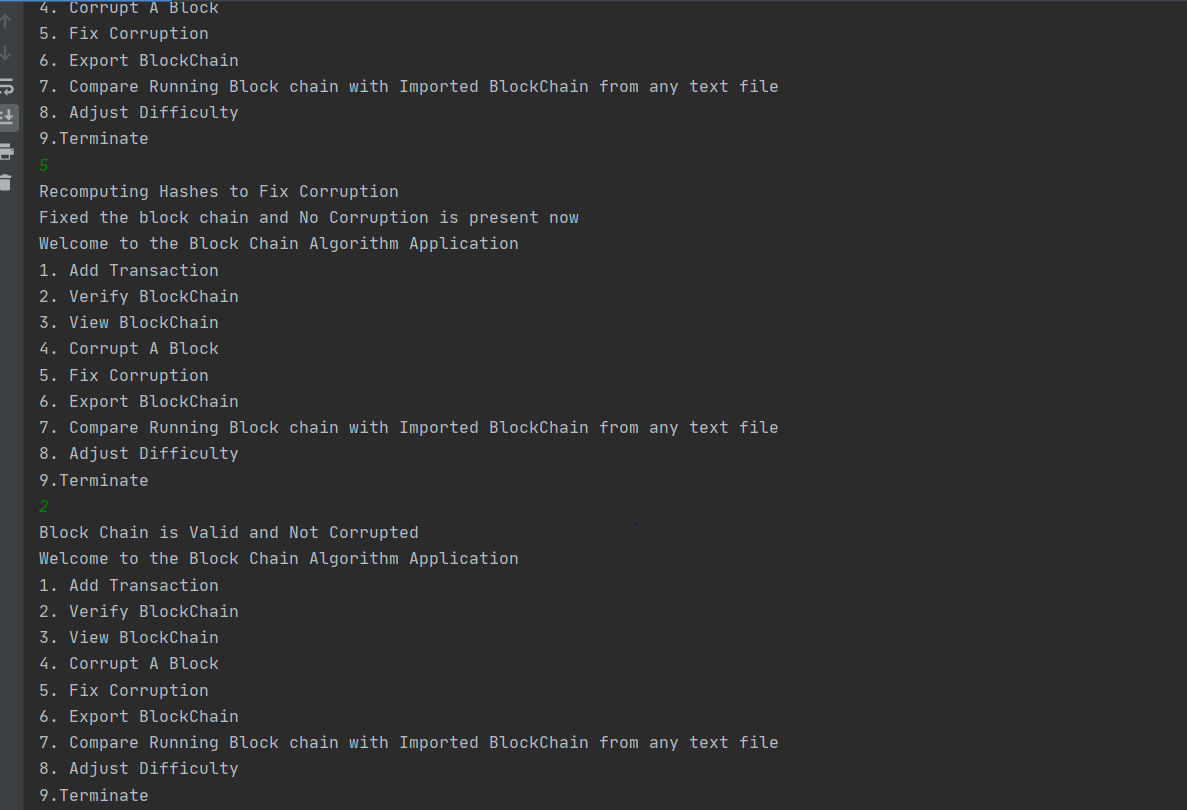
* After viewing, we can see the data field is changed to 3000.



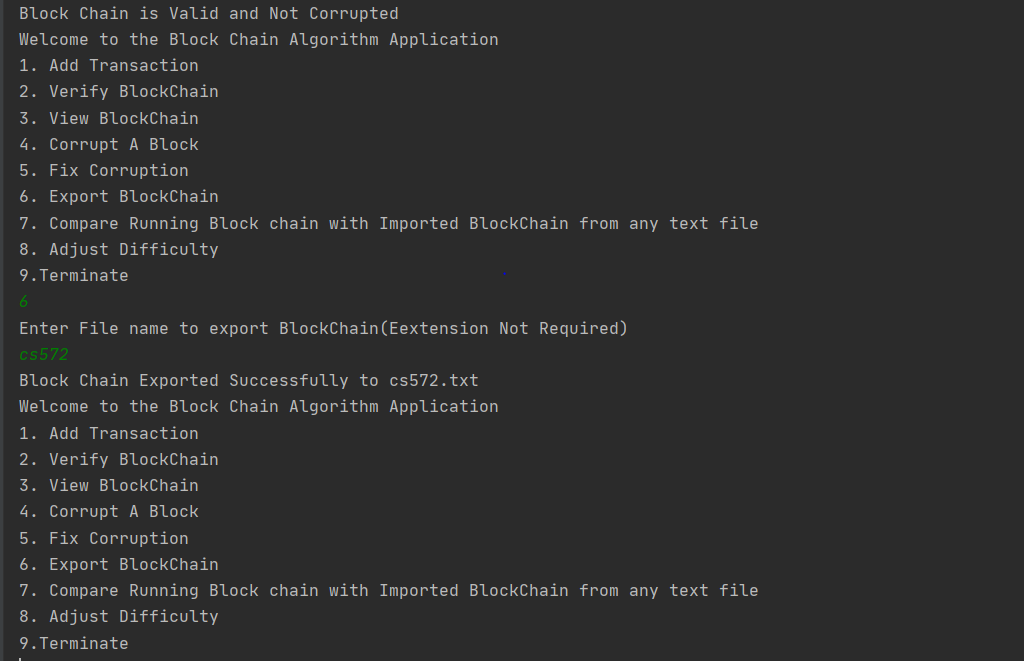
* Now if we verify the block it gives us that the block chain is corrupted at Block2 indexed from 0.



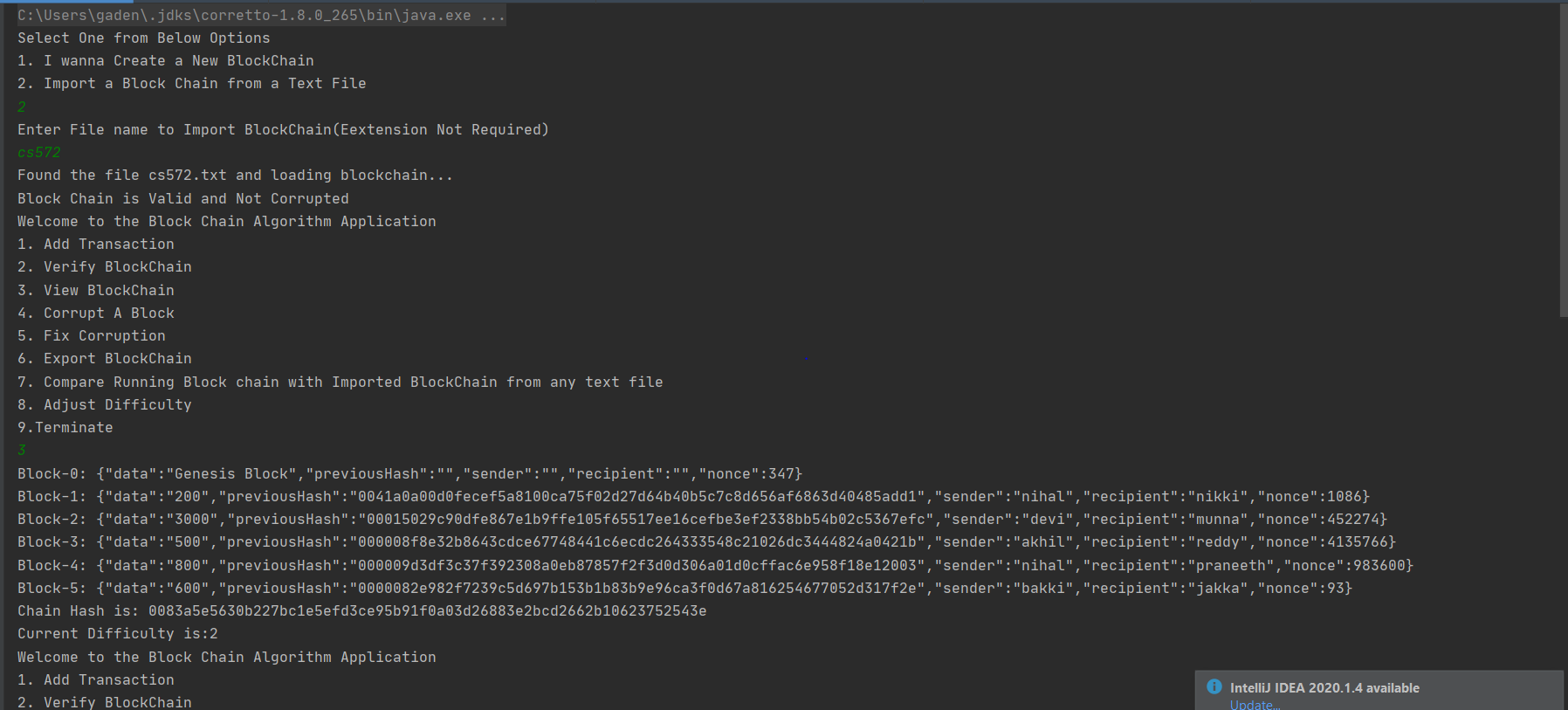
* We can fix the corruption by recomputing the hashes by using option 5. Fix Corruption and then verify it so that it gives us a valid blockchain.



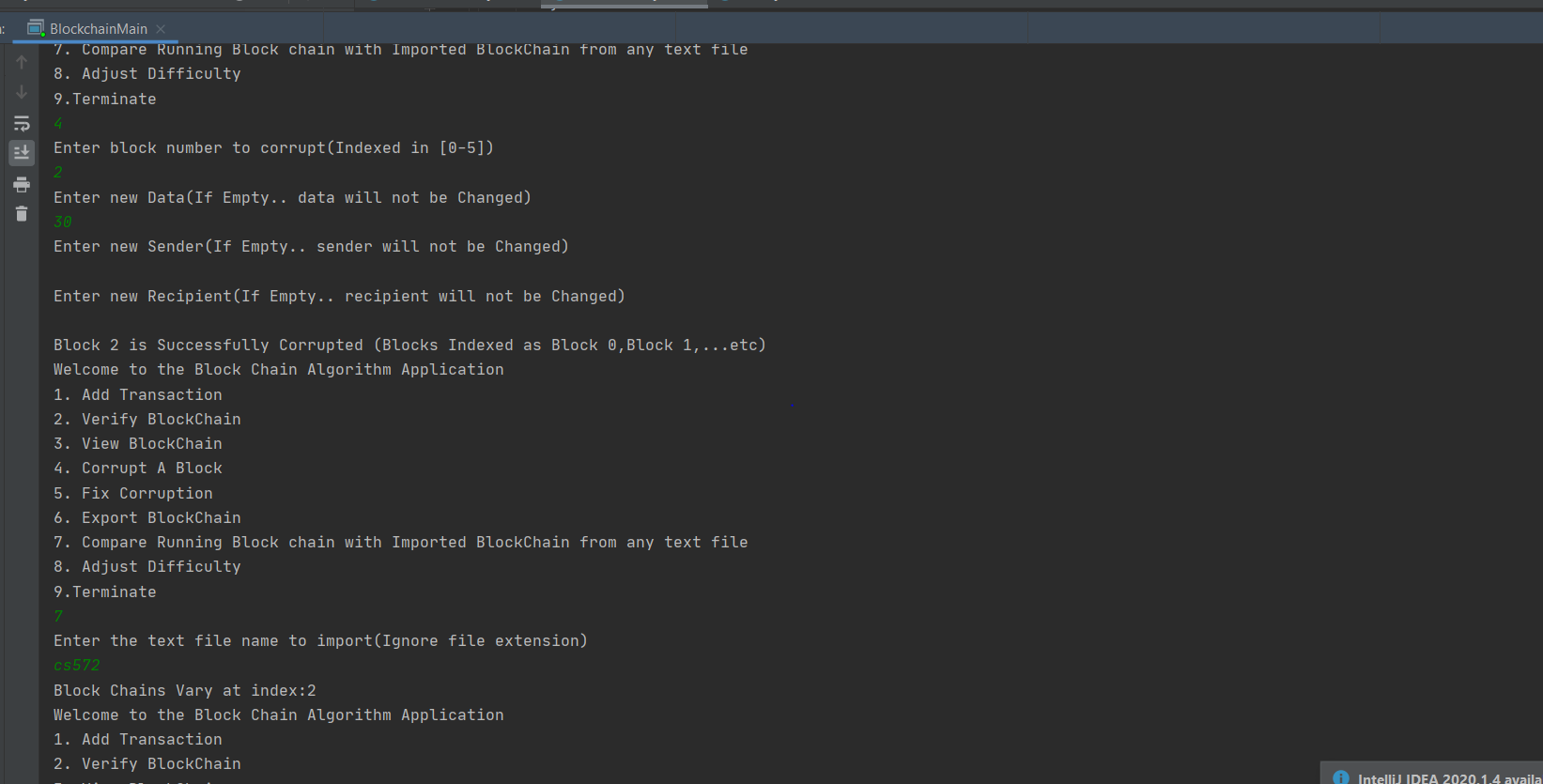
* Now let’s export this blockchain to a text file and name it as cs572.



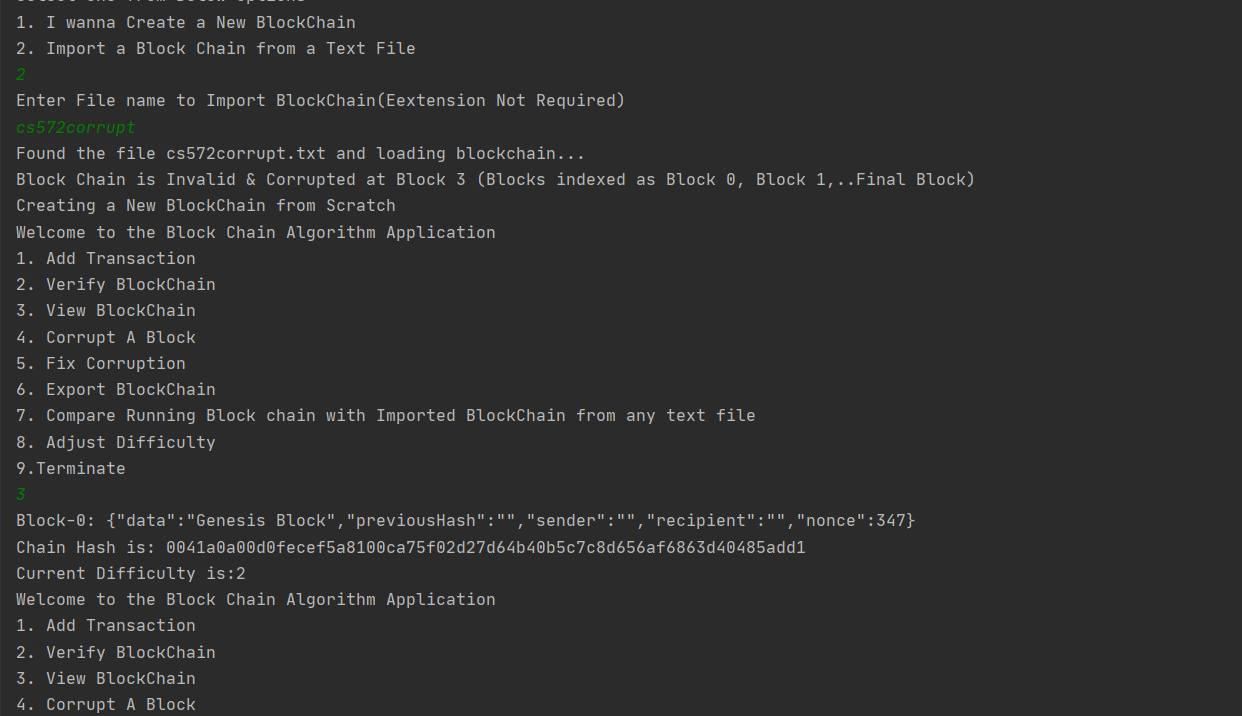
* Now Let’s get to file handling part. So that we can work on file handling from now on.
* Import cs572 and corrupt block 2 and compare it with cs572 again.



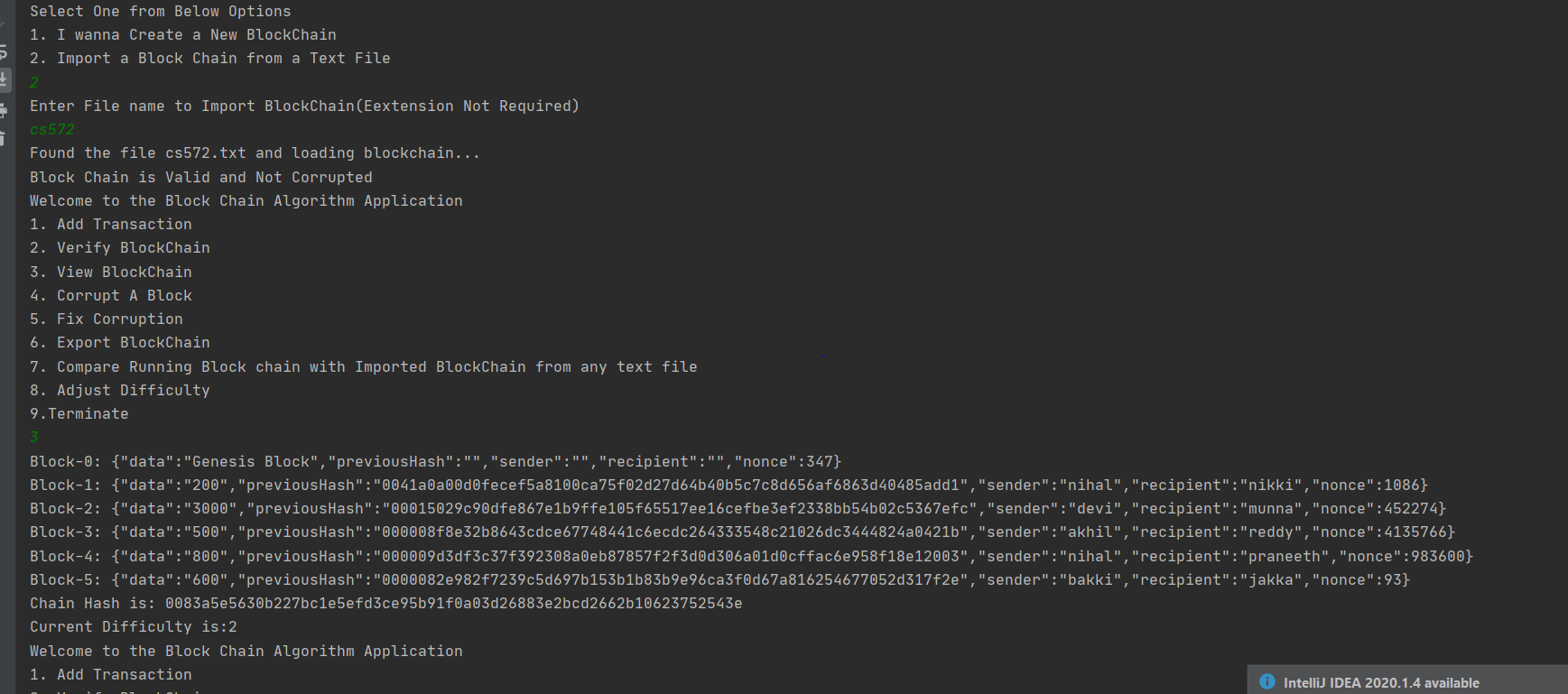
* We can see that comparison gave us that the blocks vary at index 2



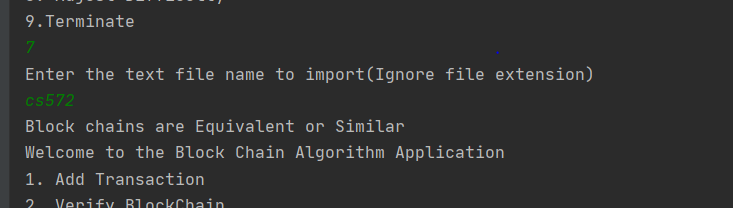
* We cannot import corrupted block as per the assignment so let’s import valid text file.
* If we import corrupted file then the program shows that the blockchain is corrupted at particular block and forces user to create a new block.
* Then if we view the block only genesis block is present as it is called by default at start of program.



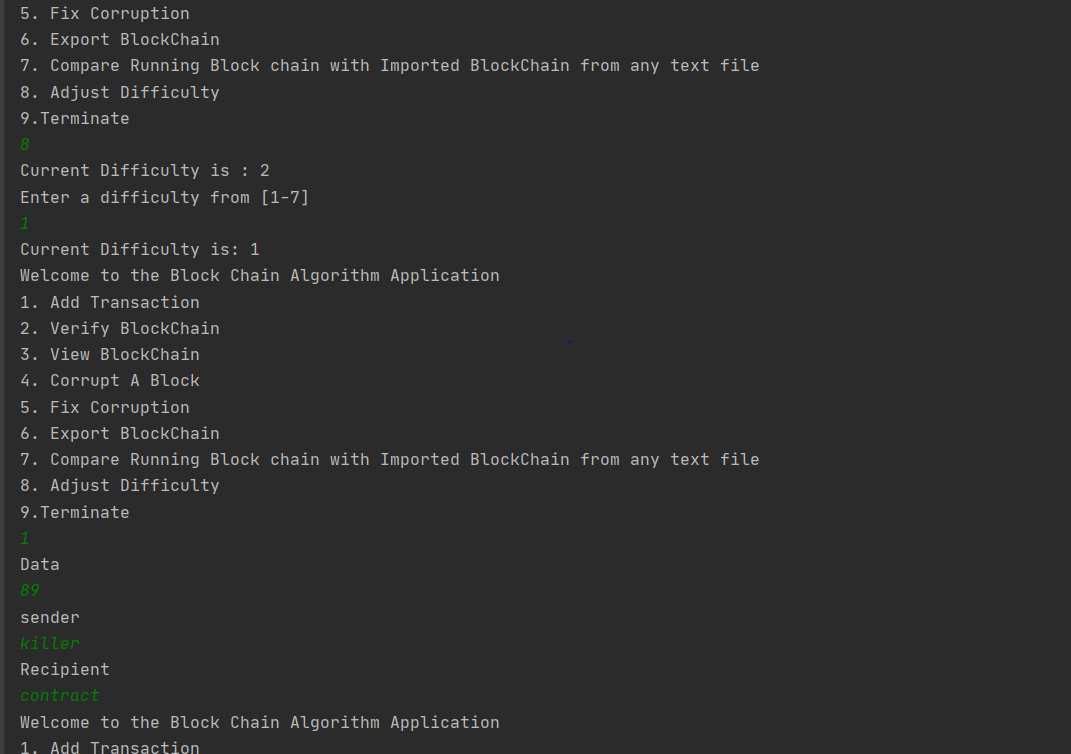
* Now let’s import valid block chain and view it. All the data will be retrieved and displayed accordingly.



* Now let’s us compare it with same text file using option 7. As both are same it gives us equivalent as output.



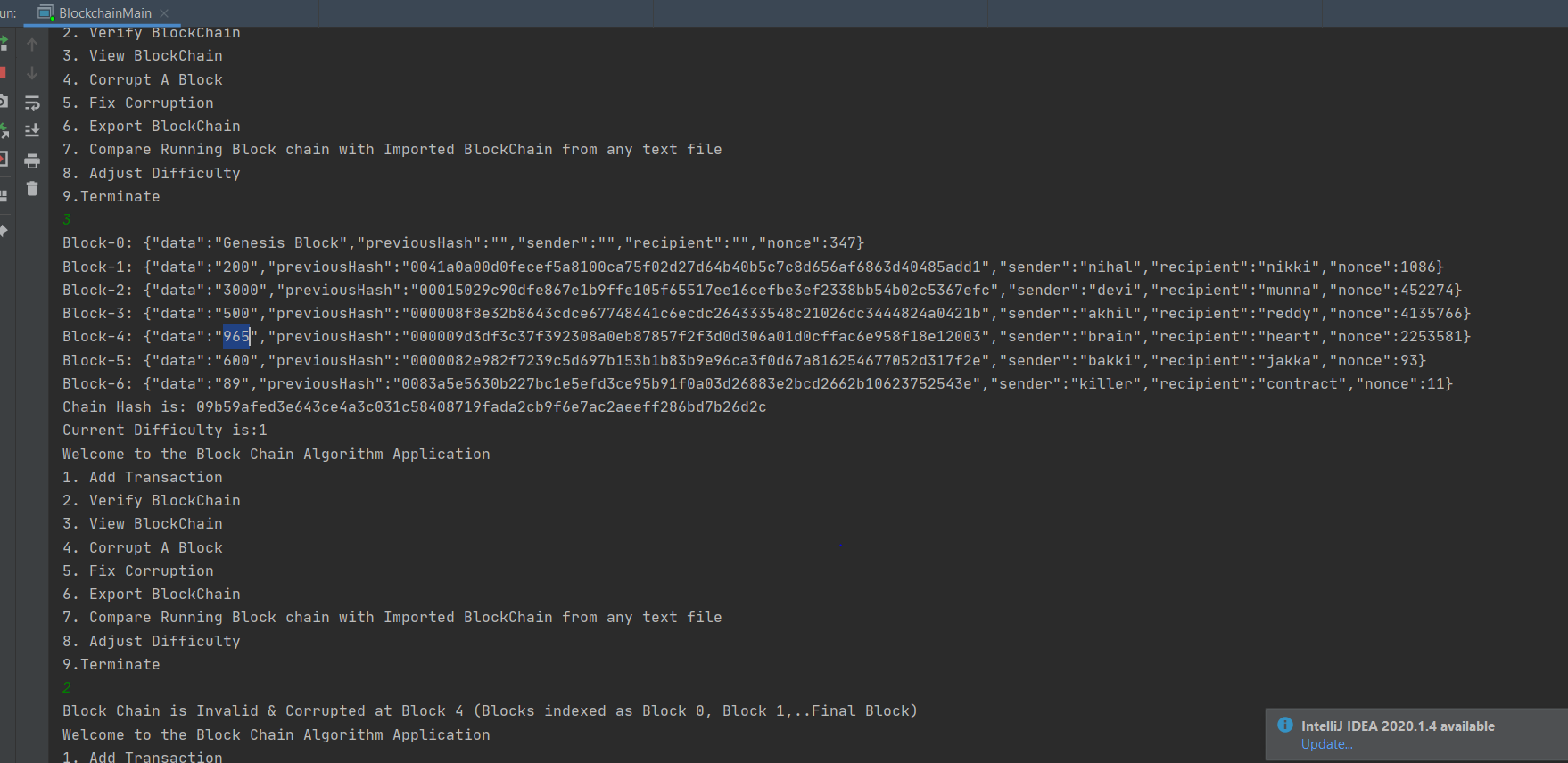
* Now let’s add one more transaction to it by adjusting difficulty to 1.
* The data will be added as block 6 as already blocks 0 to 5 are present.



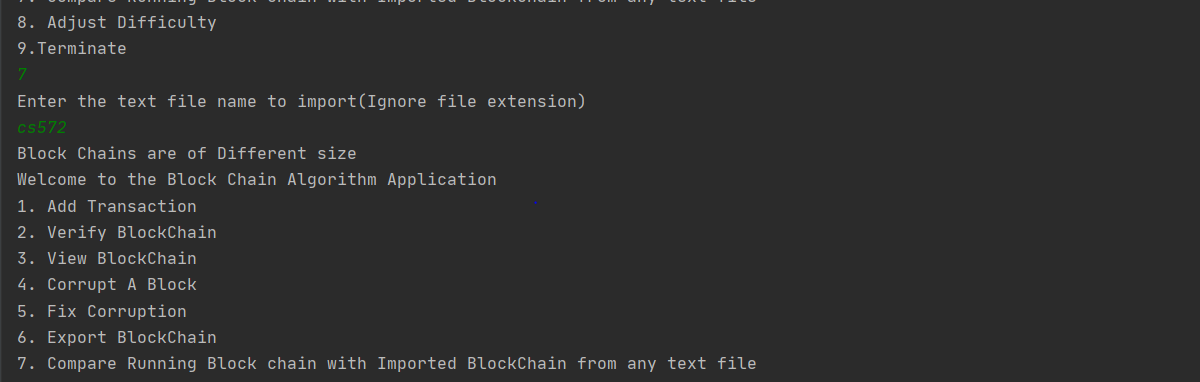
* We can view the block 6 added below.



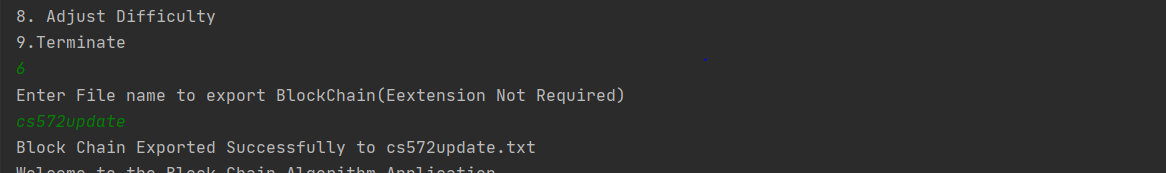
* Now let’s corrupt block-4 and view the block and verify it.
* It says that block is invalid and corrupted at block-4.



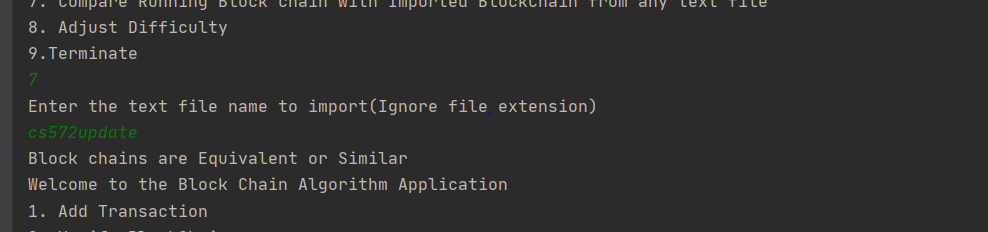
* Now let’s compare it with original text file cs572 so that the length of the file varies and displays the same.



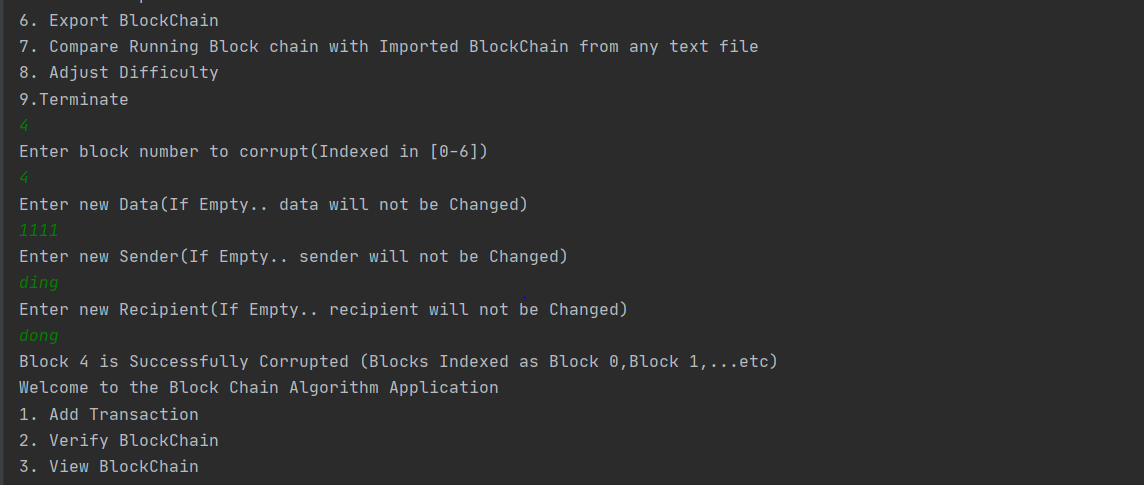
* Let’s verify, fix the corruption and export it to a file cs572update for reference.



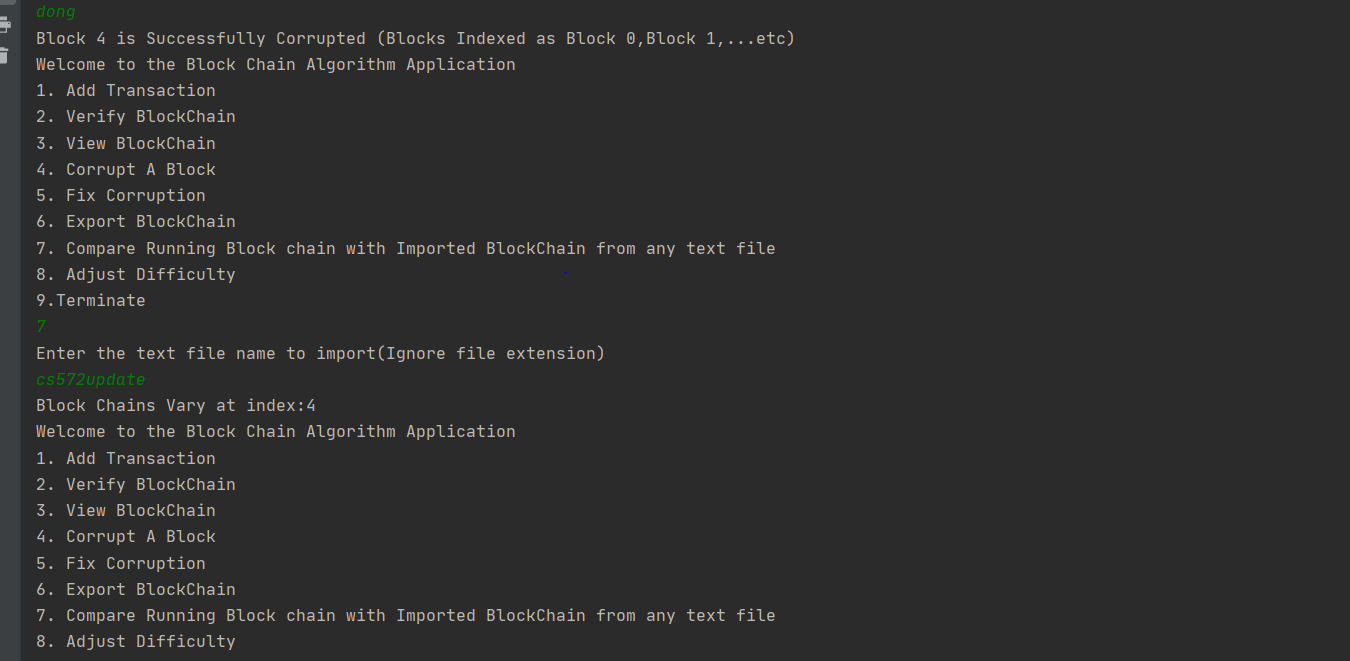
* Now if we compare present jvm blockchain with cs572update file it gives us equivalent.



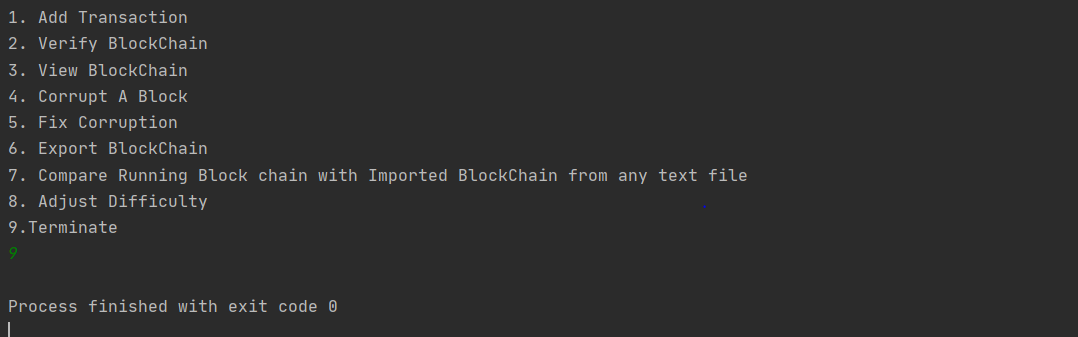
* Now let’s corrupt block-4 and see if the compare gives us correct solution.



* By comparing jvm blockchain with cs572update file blockchains vary at index-4.



* Last but not the least option 9. Terminate the program.



* With this all the functionality is tested.
* Project is zipped and included with necessary jar files.

**Thank you**