CS572-Project -1

Submitted to

Prof. David Brennan

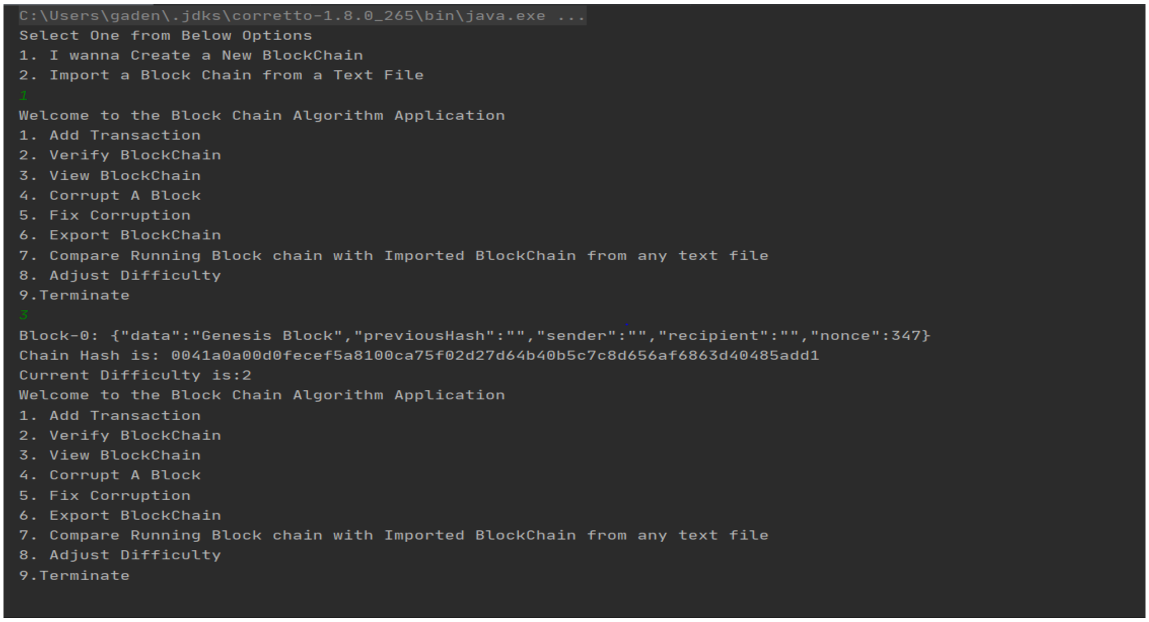
Submitted by

Nihal Reddy Gade

#310363

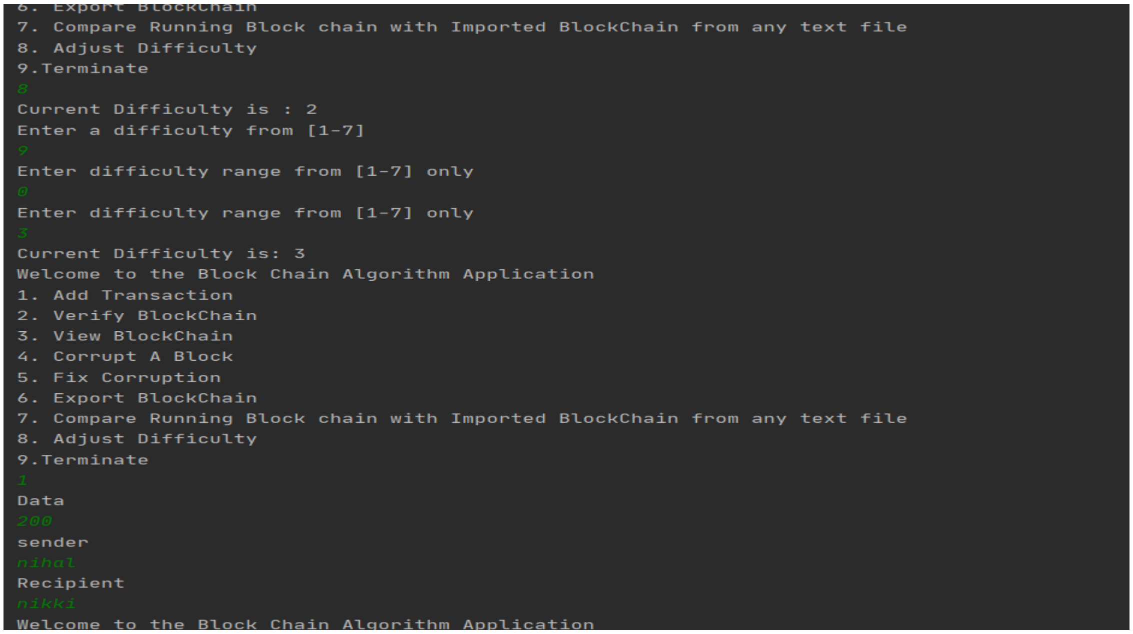
<u>Attempting operations on all 9 options:</u>

- 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 <u>3. View Blockchain.</u>



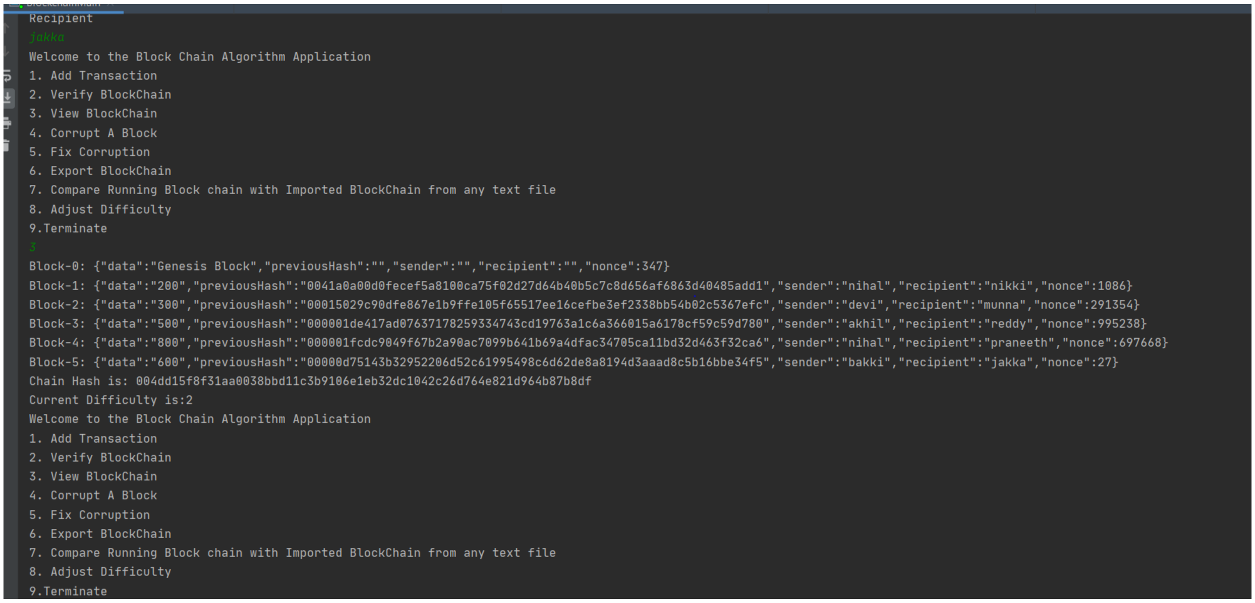
```
C:\Users\gaden\.jdks\corretto-1.8.0_265\bin\java.exe ...
Select One from Below Options
1. I wanna Create a New BlockChain
2. Import a Block Chain from a Text File
1
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
3
Block-0: {"data":"Genesis Block","previousHash":"","sender":"","recipient":"","nonce":347}
Chain Hash is: 0041a0a00d0fecef5a8100ca75f02d27d64b40b5c7c8d656af6863d40485add1
Current Difficulty is:2
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
```

- Now Let us add few more blocks. We can change the current difficulty [1-7] by using option <u>8. Adjust Difficulty.</u> 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.



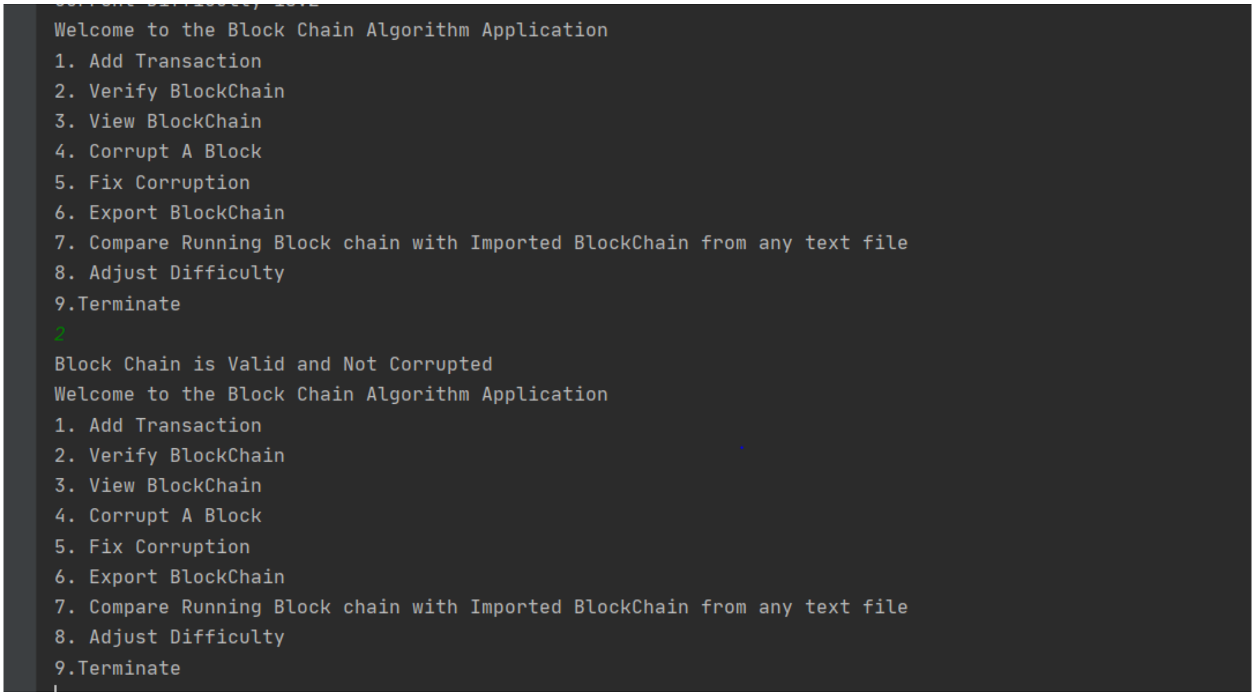
```
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
8
Current Difficulty is : 2
Enter a difficulty from [1-7]
9
Enter difficulty range from [1-7] only
0
Enter difficulty range from [1-7] only
3
Current Difficulty is: 3
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
1
Data
200
sender
nihal
Recipient
nikki
Welcome to the Block Chain Algorithm Application
```

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



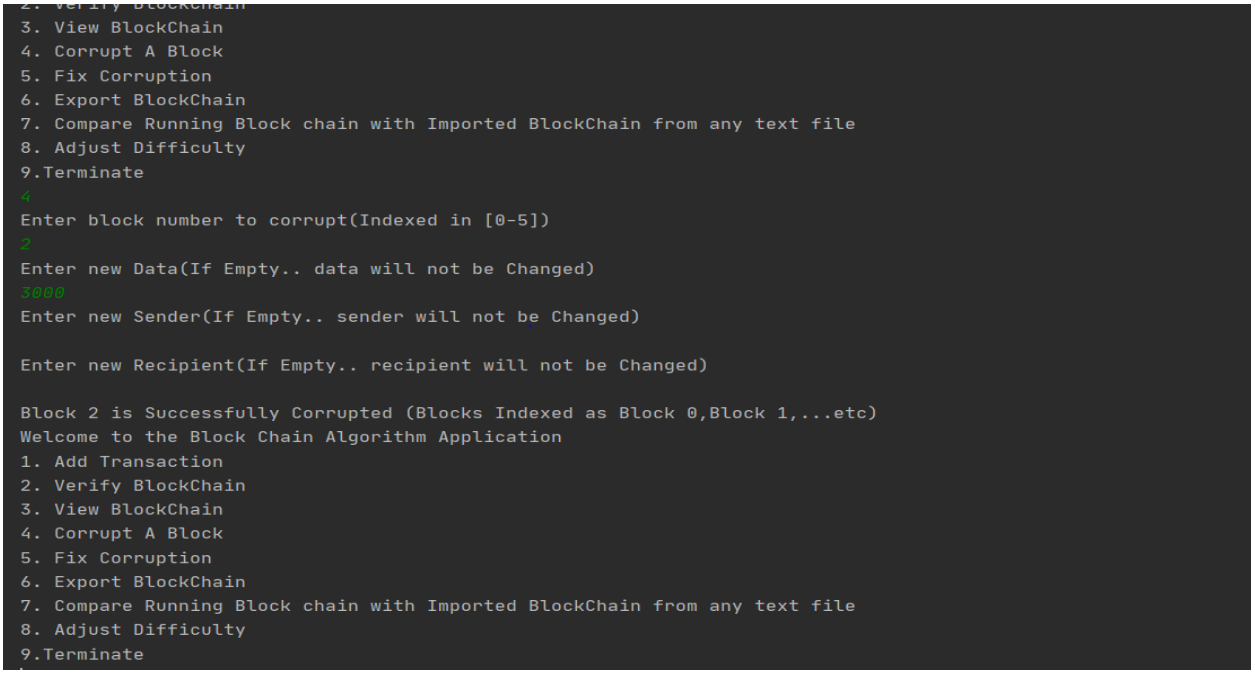
```
Recipient
jakka
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
3
Block-0: {"data":"Genesis Block","previousHash":"","sender":"","recipient":"","nonce":347}
Block-1: {"data":"200","previousHash":"0041a0a00d0fecef5a8100ca75f02d27d64b40b5c7c8d656af6863d40485add1","sender":"nihal","recipient":"nikki","nonce":1086}
Block-2: {"data":"300","previousHash":"00015029c90dfe867e1b9ffe105f65517ee16cefbe3ef2338bb54b02c5367efc","sender":"devi","recipient":"munna","nonce":291354}
Block-3: {"data":"500","previousHash":"000001de417ad07637178259334743cd19763a1c6a366015a6178cf59c59d780","sender":"akhil","recipient":"reddy","nonce":995238}
Block-4: {"data":"800","previousHash":"000001fcdc9049f67b2a90ac7099b641b69a4dfac34705ca11bd32d463f32ca6","sender":"nihal","recipient":"praneeth","nonce":697668}
Block-5: {"data":"600","previousHash":"00000d75143b32952206d52c61995498c6d62de8a8194d3aaad8c5b16bbe34f5","sender":"bakki","recipient":"jakka","nonce":27}
Chain Hash is: 004dd15f8f31aa0038bbd11c3b9106e1eb32dc1042c26d764e821d964b87b8df
Current Difficulty is:2
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
```

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



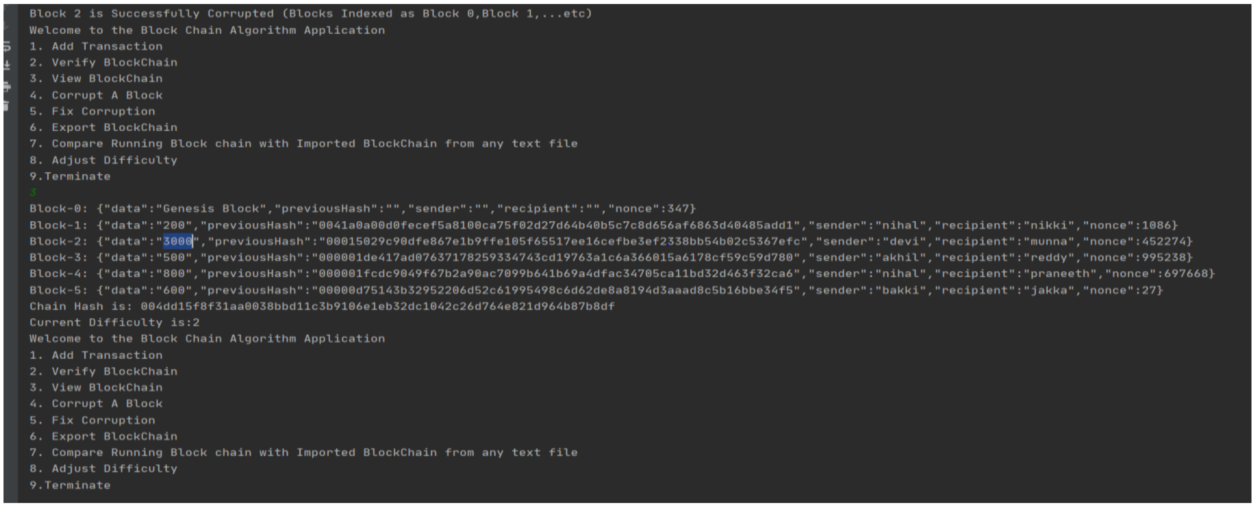
```
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
2
Block Chain is Valid and Not Corrupted
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
```

- 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.



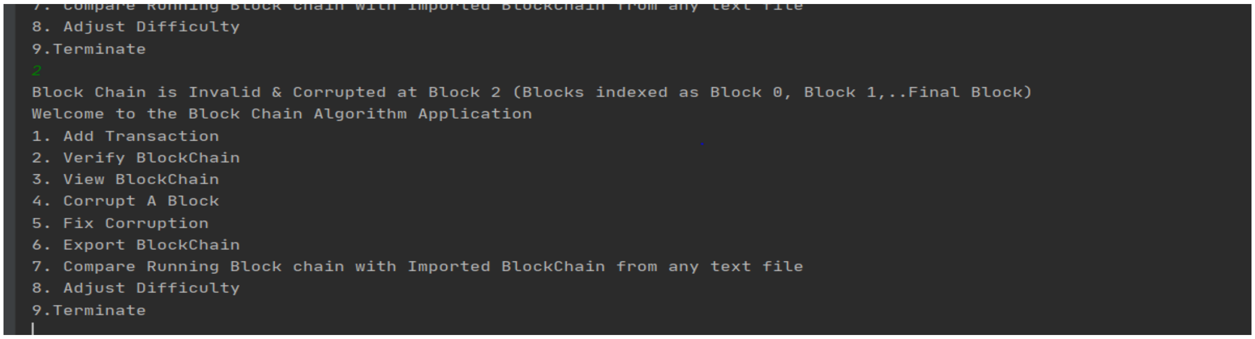
```
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
4
Enter block number to corrupt(Indexed in [0-5])
2
Enter new Data(If Empty.. data will not be Changed)
3000
Enter new Sender(If Empty.. sender will not be Changed)

Enter new Recipient(If Empty.. recipient will not be Changed)

Block 2 is Successfully Corrupted (Blocks Indexed as Block 0,Block 1,...etc)
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
```

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



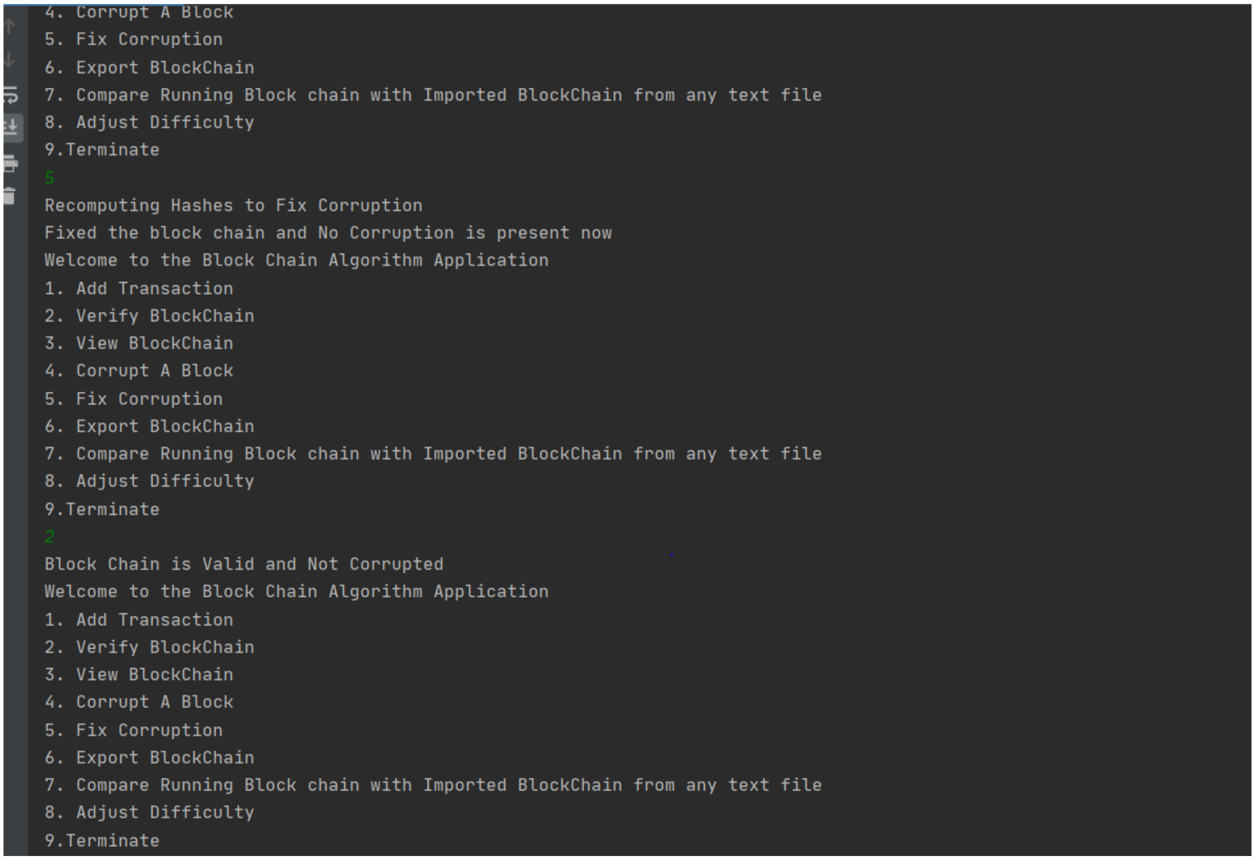
```
Block 2 is Successfully Corrupted (Blocks Indexed as Block 0,Block 1,...etc)
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
3
Block-0: {"data":"Genesis Block","previousHash":"","sender":"","recipient":"","nonce":347}
Block-1: {"data":"200","previousHash":"0041a0a00d0fecef5a8100ca75f02d27d64b40b5c7c8d656af6863d40485add1","sender":"nihal","recipient":"nikki","nonce":1086}
Block-2: {"data":"3000","previousHash":"00015029c90dfe867e1b9ffe105f65517ee16cefbe3ef2338bb54b02c5367efc","sender":"devi","recipient":"munna","nonce":452274}
Block-3: {"data":"500","previousHash":"000001de417ad07637178259334743cd19763a1c6a366015a6178cf59c59d780","sender":"akhil","recipient":"reddy","nonce":995238}
Block-4: {"data":"800","previousHash":"000001fcdc9049f67b2a90ac7099b641b69a4dfac34705ca11bd32d463f32ca6","sender":"nihal","recipient":"praneeth","nonce":697668}
Block-5: {"data":"600","previousHash":"00000d75143b32952206d52c61995498c6d62de8a8194d3aaad8c5b16bbe34f5","sender":"bakki","recipient":"jakka","nonce":27}
Chain Hash is: 004dd15f8f31aa0038bbd11c3b9106e1eb32dc1042c26d764e821d964b87b8df
Current Difficulty is:2
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
```

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



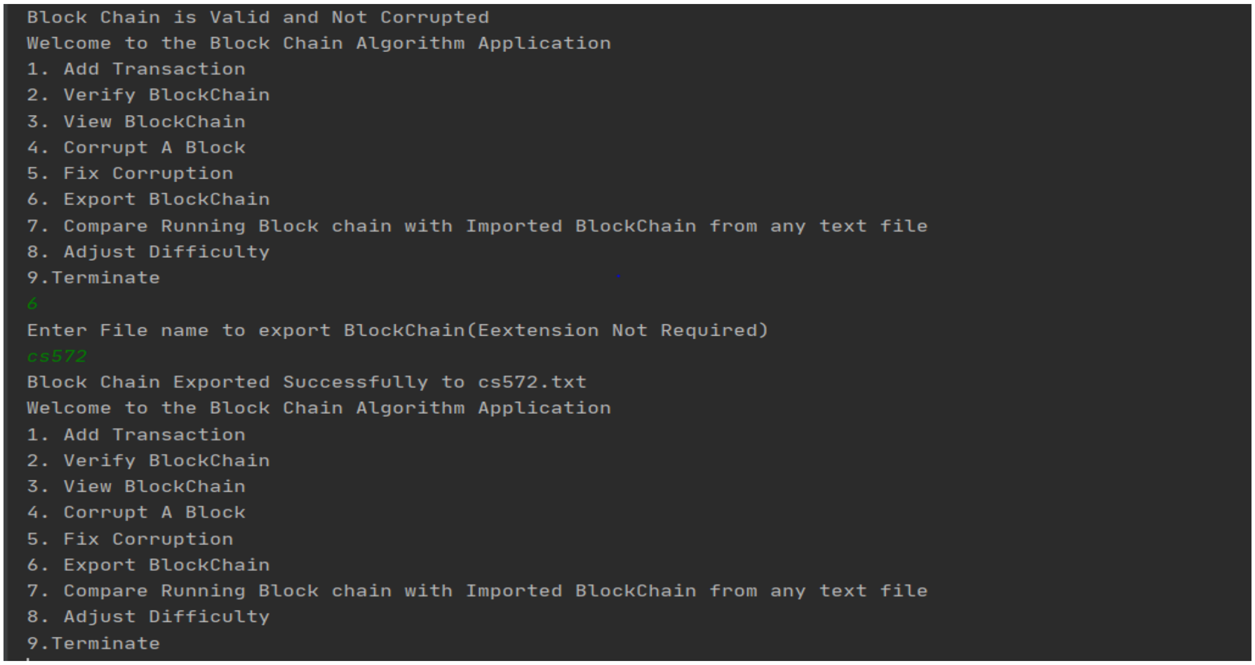
```
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
2
Block Chain is Invalid & Corrupted at Block 2 (Blocks indexed as Block 0, Block 1,..Final Block)
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
```

- 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.



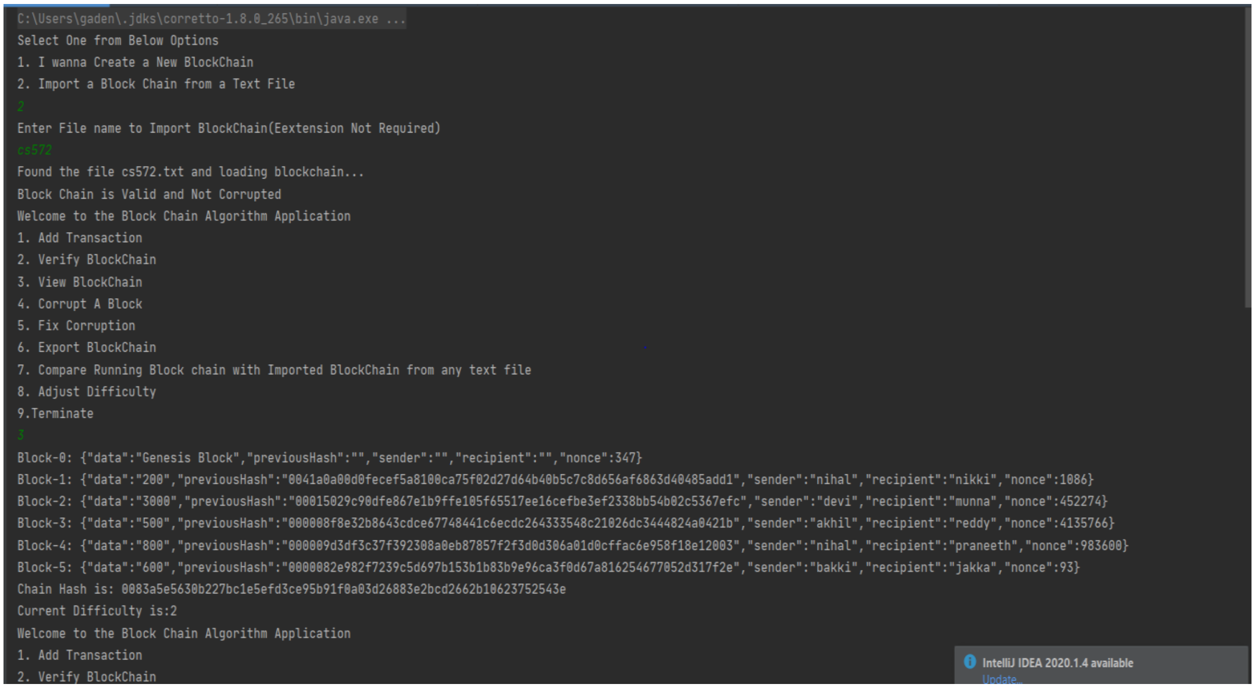
```
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
5
Recomputing Hashes to Fix Corruption
Fixed the block chain and No Corruption is present now
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
2
Block Chain is Valid and Not Corrupted
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
```

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



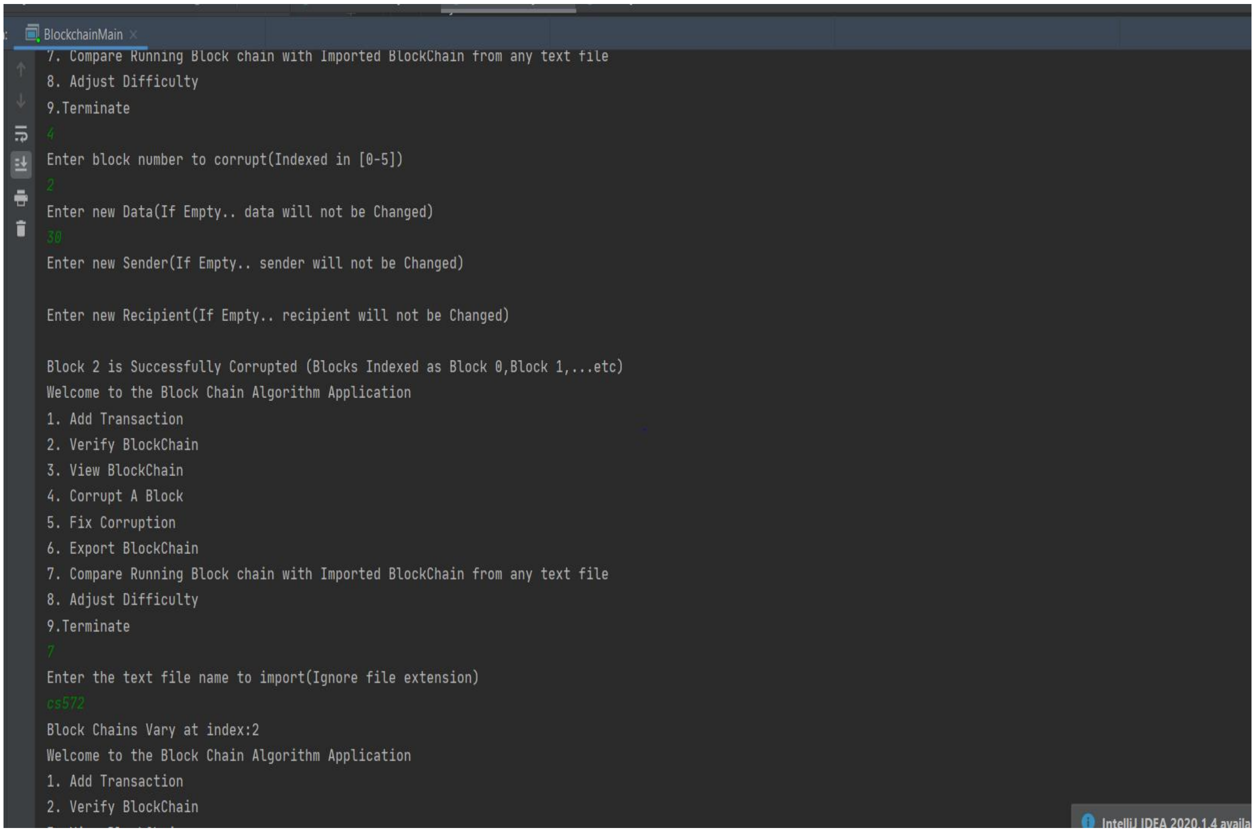
```
Block Chain is Valid and Not Corrupted
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
6
Enter File name to export BlockChain(Eextension Not Required)
cs572
Block Chain Exported Successfully to cs572.txt
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
```

- 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.



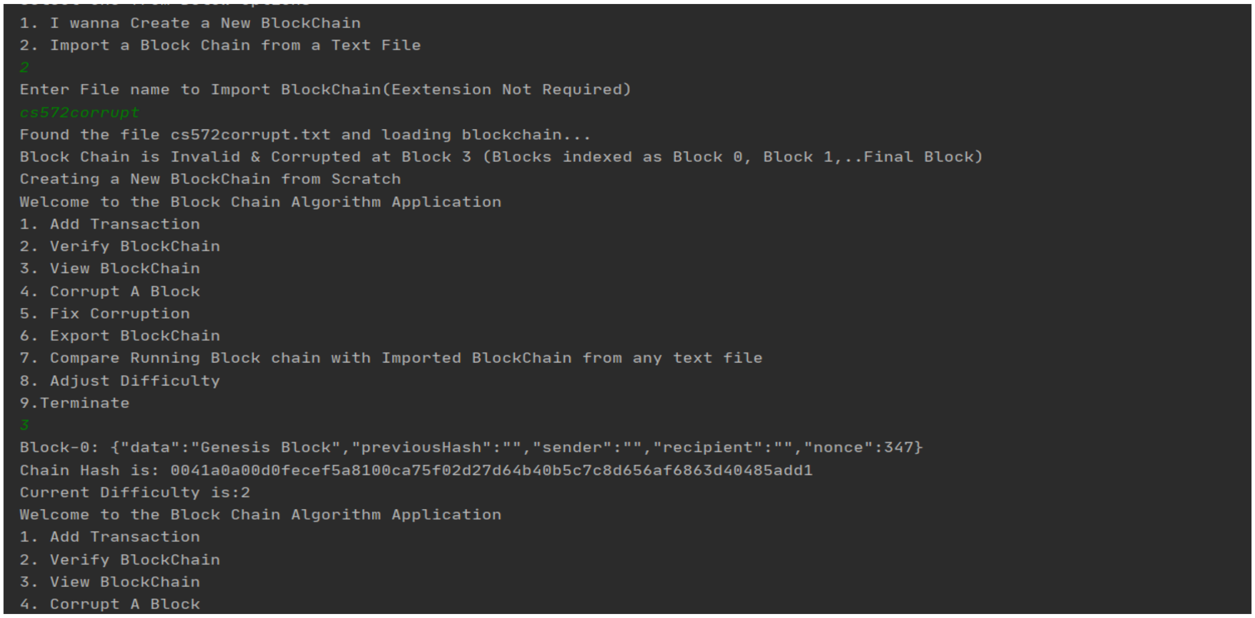
```
C:\Users\gaden\.jdks\corretto-1.8.0_265\bin\java.exe ...
Select One from Below Options
1. I wanna Create a New BlockChain
2. Import a Block Chain from a Text File
2
Enter File name to Import BlockChain(Eextension Not Required)
cs572
Found the file cs572.txt and loading blockchain...
Block Chain is Valid and Not Corrupted
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
3
Block-0: {"data":"Genesis Block","previousHash":"","sender":"","recipient":"","nonce":347}
Block-1: {"data":"200","previousHash":"0041a0a00d0fecef5a8100ca75f02d27d64b40b5c7c8d656af6863d40485add1","sender":"nihal","recipient":"nikki","nonce":1086}
Block-2: {"data":"3000","previousHash":"00015029c90dfe867e1b9ffe105f65517ee16cefbe3ef2338bb54b02c5367efc","sender":"devi","recipient":"munna","nonce":452274}
Block-3: {"data":"500","previousHash":"000008f8e32b8643cdce67748441c6ecdc264333548c21026dc3444824a0421b","sender":"akhil","recipient":"reddy","nonce":4135766}
Block-4: {"data":"800","previousHash":"000009d3df3c37f392308a0eb87857f2f3d0d306a01d0cffac6e958f18e12003","sender":"nihal","recipient":"praneeth","nonce":983600}
Block-5: {"data":"600","previousHash":"0000082e982f7239c5d697b153b1b83b9e96ca3f0d67a816254677052d317f2e","sender":"bakki","recipient":"jakka","nonce":93}
Chain Hash is: 0083a5e5630b227bc1e5efd3ce95b91f0a03d26883e2bcd2662b10623752543e
Current Difficulty is:2
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
```

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



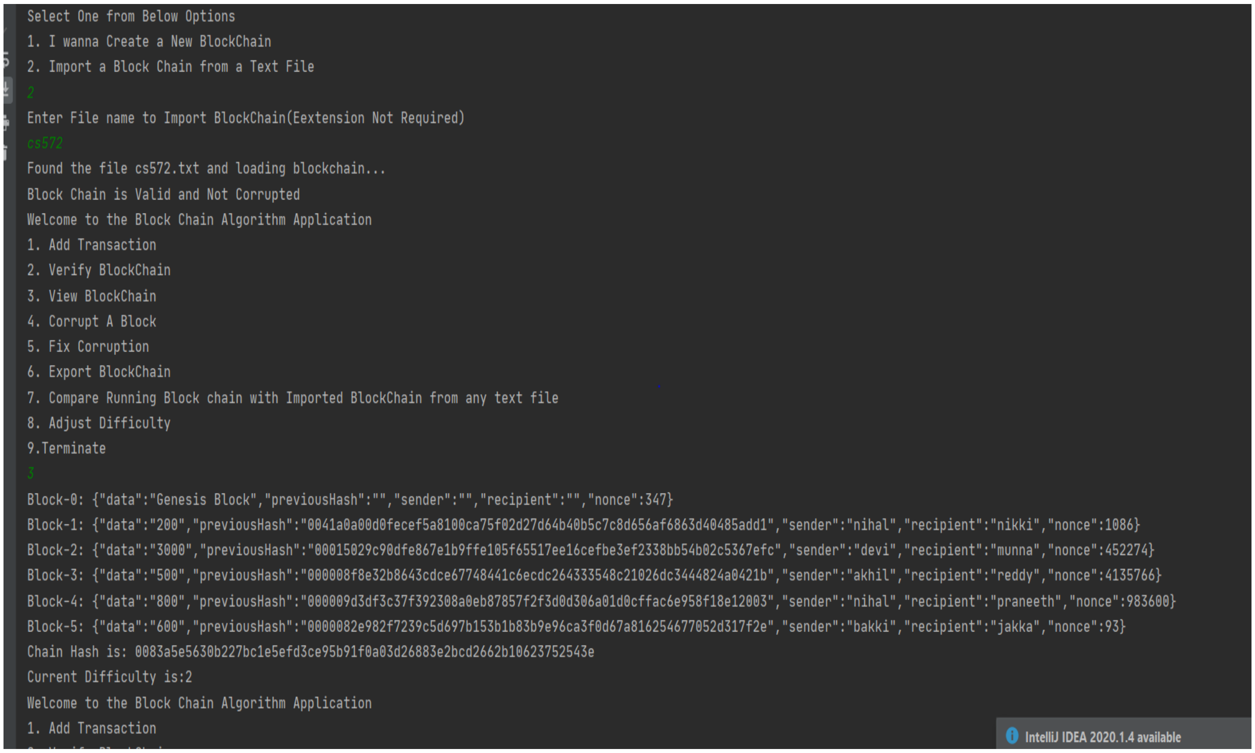
```
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
4
Enter block number to corrupt(Indexed in [0-5])
2
Enter new Data(If Empty.. data will not be Changed)
30
Enter new Sender(If Empty.. sender will not be Changed)

Enter new Recipient(If Empty.. recipient will not be Changed)

Block 2 is Successfully Corrupted (Blocks Indexed as Block 0,Block 1,...etc)
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
7
Enter the text file name to import(Ignore file extension)
cs572
Block Chains Vary at index:2
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
```

- 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.



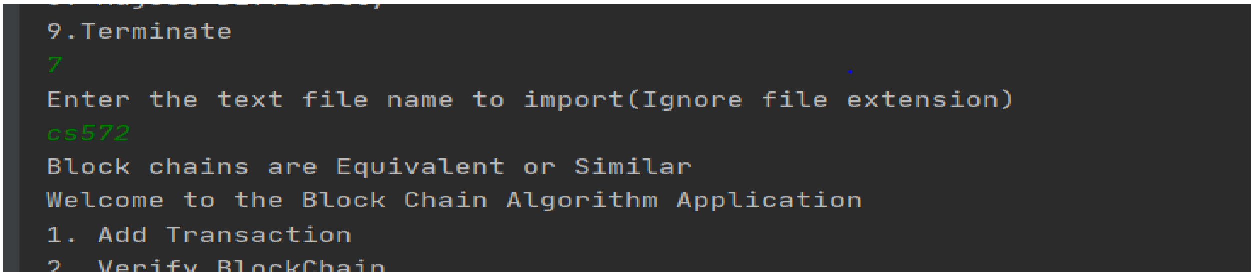
```
1. I wanna Create a New BlockChain
2. Import a Block Chain from a Text File
2
Enter File name to Import BlockChain(Eextension Not Required)
cs572corrupt
Found the file cs572corrupt.txt and loading blockchain...
Block Chain is Invalid & Corrupted at Block 3 (Blocks indexed as Block 0, Block 1,..Final Block)
Creating a New BlockChain from Scratch
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
3
Block-0: {"data":"Genesis Block","previousHash":"","sender":"","recipient":"","nonce":347}
Chain Hash is: 0041a0a00d0fecef5a8100ca75f02d27d64b40b5c7c8d656af6863d40485add1
Current Difficulty is:2
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
```

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



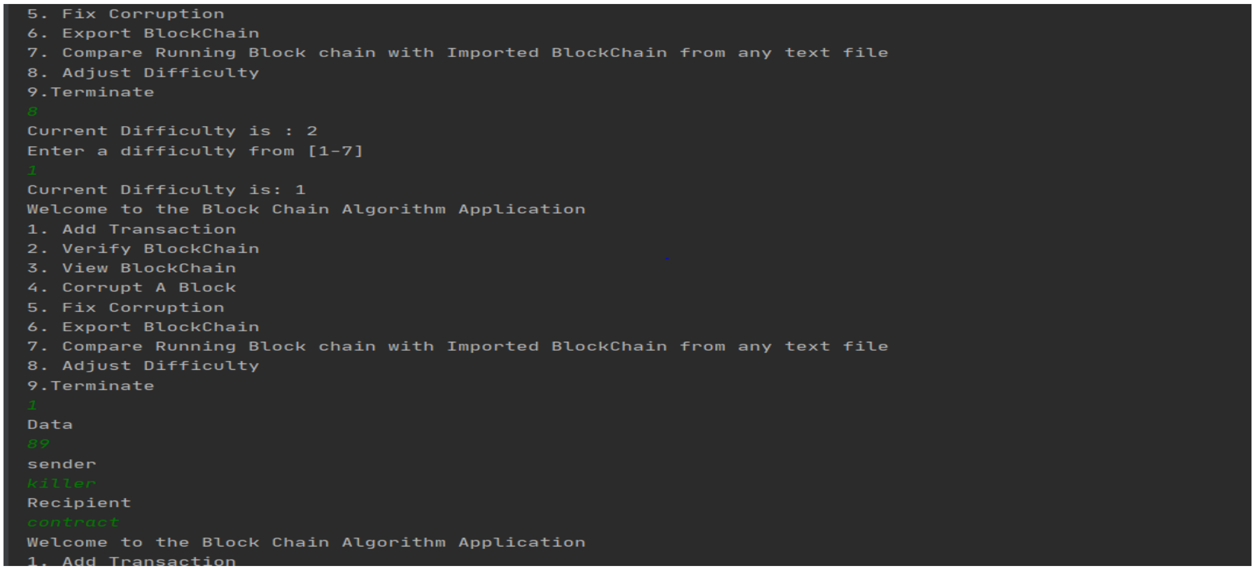
```
Select One from Below Options
1. I wanna Create a New BlockChain
2. Import a Block Chain from a Text File
2
Enter File name to Import BlockChain(Eextension Not Required)
cs572
Found the file cs572.txt and loading blockchain...
Block Chain is Valid and Not Corrupted
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
3
Block-0: {"data":"Genesis Block","previousHash":"","sender":"","recipient":"","nonce":347}
Block-1: {"data":"200","previousHash":"0041a0a00d0fecef5a8100ca75f02d27d64b40b5c7c8d656af6863d40485add1","sender":"nihal","recipient":"nikki","nonce":1086}
Block-2: {"data":"3000","previousHash":"00015029c90dfe867e1b9ffe105f65517ee16cefbe3ef2338bb54b02c5367efc","sender":"devi","recipient":"munna","nonce":452274}
Block-3: {"data":"500","previousHash":"000008f8e32b8643cdce67748441c6ecdc264333548c21026dc3444824a0421b","sender":"akhil","recipient":"reddy","nonce":4135766}
Block-4: {"data":"800","previousHash":"000009d3df3c37f392308a0eb87857f2f3d0d306a01d0cffac6e958f18e12003","sender":"nihal","recipient":"praneeth","nonce":983600}
Block-5: {"data":"600","previousHash":"0000082e982f7239c5d697b153b1b83b9e96ca3f0d67a816254677052d317f2e","sender":"bakki","recipient":"jakka","nonce":93}
Chain Hash is: 0083a5e5630b227bc1e5efd3ce95b91f0a03d26883e2bcd2662b10623752543e
Current Difficulty is:2
Welcome to the Block Chain Algorithm Application
1. Add Transaction
```

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



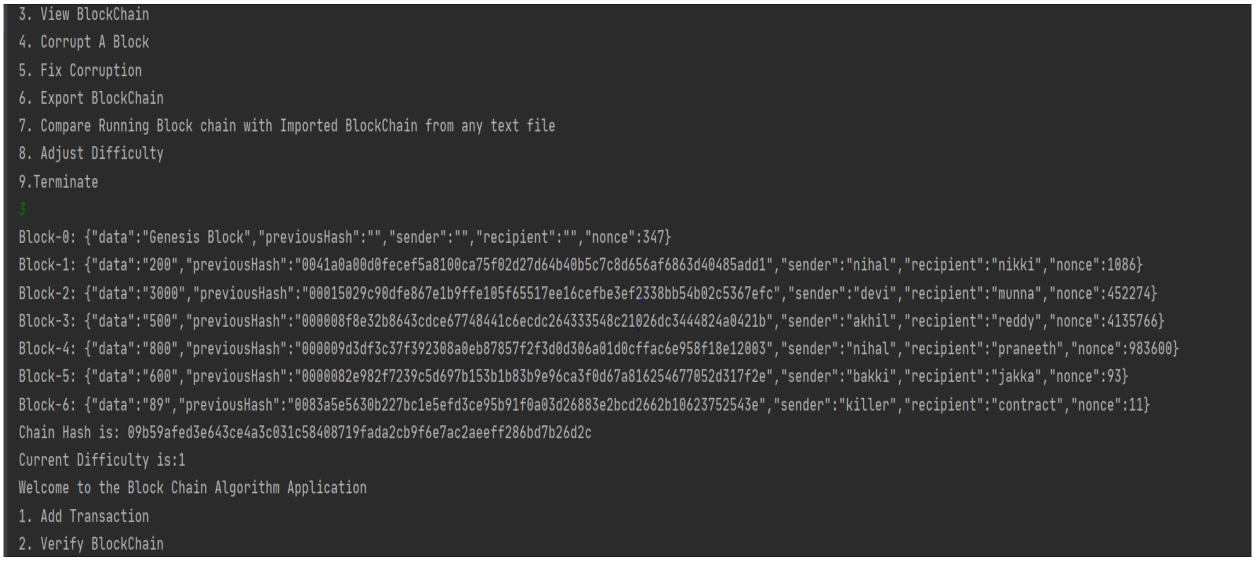
```
8. Adjust Difficulty
9.Terminate
7
Enter the text file name to import(Ignore file extension)
cs572
Block chains are Equivalent or Similar
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
```

- 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.



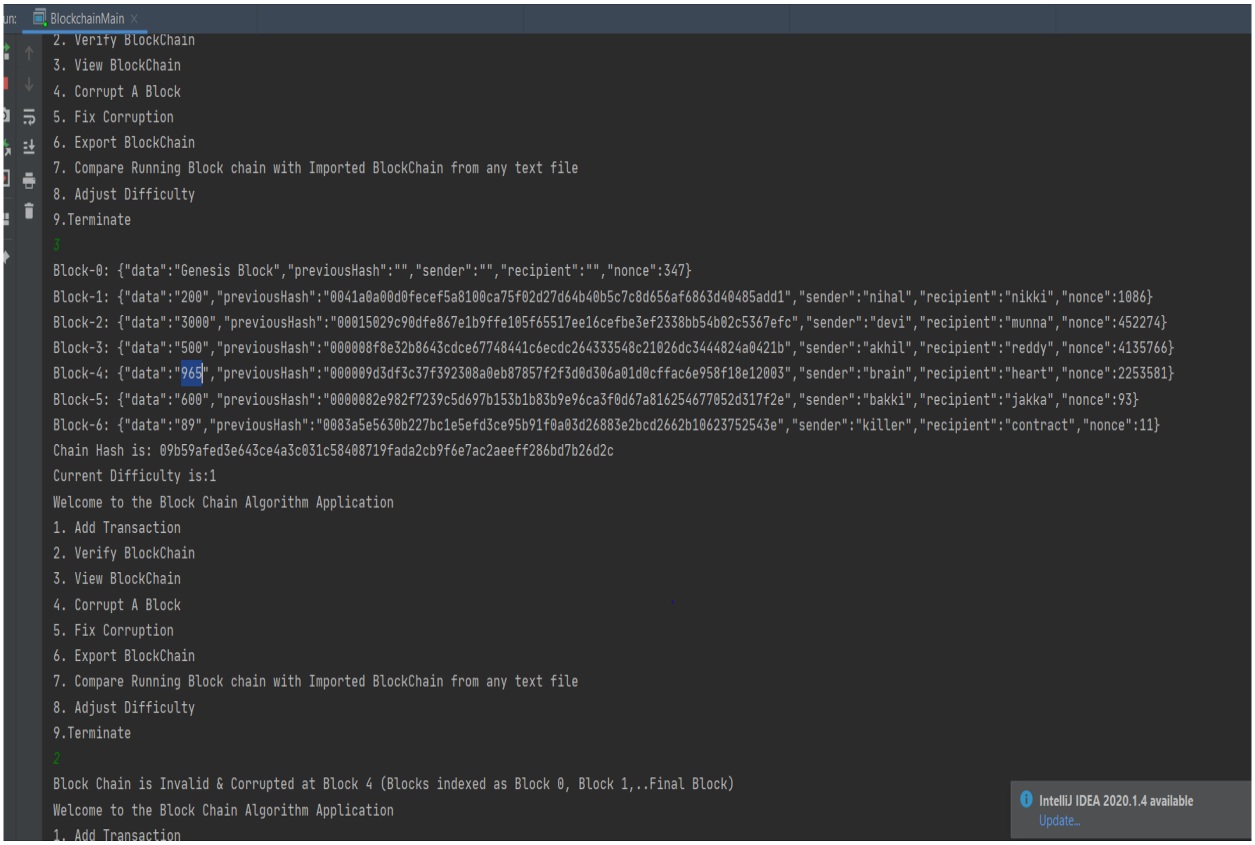
```
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
8
Current Difficulty is : 2
Enter a difficulty from [1-7]
1
Current Difficulty is: 1
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
1
Data
89
sender
killer
Recipient
contract
Welcome to the Block Chain Algorithm Application
1. Add Transaction
```

- We can view the block 6 added below.



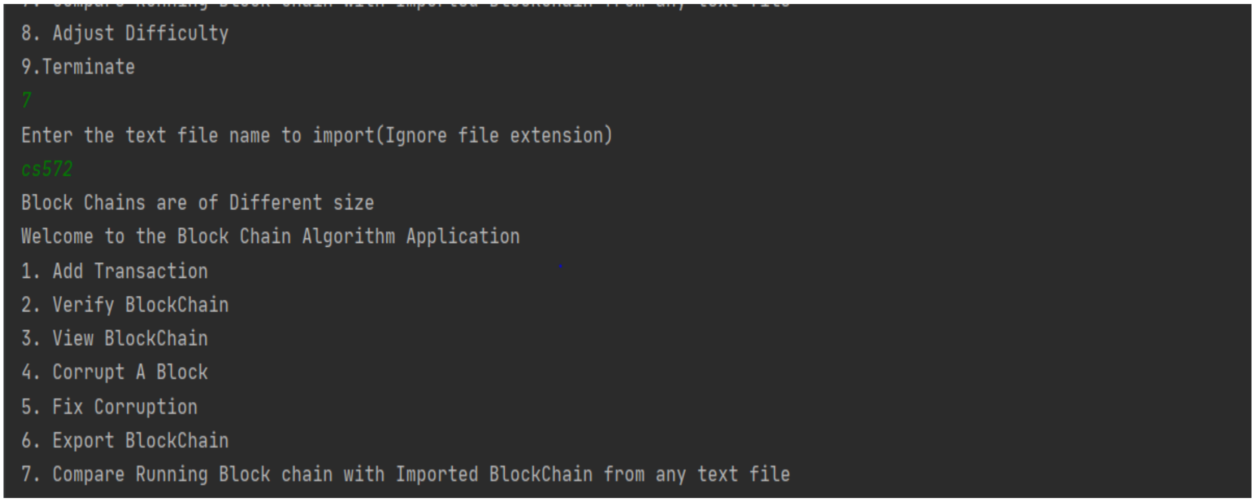
```
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
3
Block-0: {"data":"Genesis Block","previousHash":"","sender":"","recipient":"","nonce":347}
Block-1: {"data":"200","previousHash":"0041a0a00d0fecef5a8100ca75f02d27d64b40b5c7c8d656af6863d40485add1","sender":"nihal","recipient":"nikki","nonce":1086}
Block-2: {"data":"3000","previousHash":"00015029c90dfe867e1b9ffe105f65517ee16cefbe3ef2338bb54b02c5367efc","sender":"devi","recipient":"munna","nonce":452274}
Block-3: {"data":"500","previousHash":"000008f8e32b8643cdce67748441c6ecdc264333548c21026dc3444824a0421b","sender":"akhil","recipient":"reddy","nonce":4135766}
Block-4: {"data":"800","previousHash":"000009d3df3c37f392308a0eb87857f2f3d0d306a01d0cffac6e958f18e12003","sender":"nihal","recipient":"praneeth","nonce":983600}
Block-5: {"data":"600","previousHash":"0000082e982f7239c5d697b153b1b83b9e96ca3f0d67a816254677052d317f2e","sender":"bakki","recipient":"jakka","nonce":93}
Block-6: {"data":"89","previousHash":"0083a5e5630b227bc1e5efd3ce95b91f0a03d26883e2bcd2662b10623752543e","sender":"killer","recipient":"contract","nonce":11}
Chain Hash is: 09b59afed3e643ce4a3c031c58408719fada2cb9f6e7ac2aeeff286bd7b26d2c
Current Difficulty is:1
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
```

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



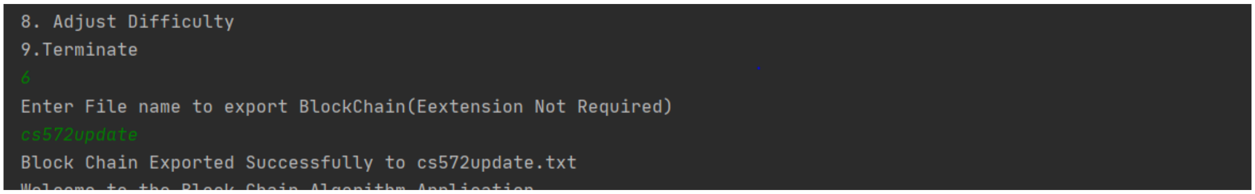
```
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
3
Block-0: {"data":"Genesis Block","previousHash":"","sender":"","recipient":"","nonce":347}
Block-1: {"data":"200","previousHash":"0041a0a00d0fecef5a8100ca75f02d27d64b40b5c7c8d656af6863d40485add1","sender":"nihal","recipient":"nikki","nonce":1086}
Block-2: {"data":"3000","previousHash":"00015029c90dfe867e1b9ffe105f65517ee16cefbe3ef2338bb54b02c5367efc","sender":"devi","recipient":"munna","nonce":452274}
Block-3: {"data":"500","previousHash":"000008f8e32b8643cdce67748441c6ecdc264333548c21026dc3444824a0421b","sender":"akhil","recipient":"reddy","nonce":4135766}
Block-4: {"data":"965","previousHash":"000009d3df3c37f392308a0eb87857f2f3d0d306a01d0cffac6e958f18e12003","sender":"brain","recipient":"heart","nonce":2253581}
Block-5: {"data":"600","previousHash":"0000082e982f7239c5d697b153b1b83b9e96ca3f0d67a816254677052d317f2e","sender":"bakki","recipient":"jakka","nonce":93}
Block-6: {"data":"89","previousHash":"0083a5e5630b227bc1e5efd3ce95b91f0a03d26883e2bcd2662b10623752543e","sender":"killer","recipient":"contract","nonce":11}
Chain Hash is: 09b59afed3e643ce4a3c031c58408719fada2cb9f6e7ac2aeeff286bd7b26d2c
Current Difficulty is:1
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
2
Block Chain is Invalid & Corrupted at Block 4 (Blocks indexed as Block 0, Block 1,..Final Block)
Welcome to the Block Chain Algorithm Application
1. Add Transaction
```

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



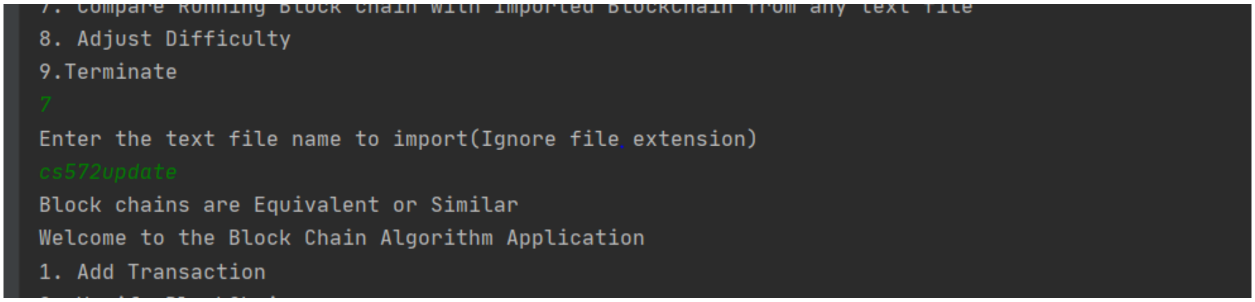
```
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
7
Enter the text file name to import(Ignore file extension)
cs572
Block Chains are of Different size
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
```

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



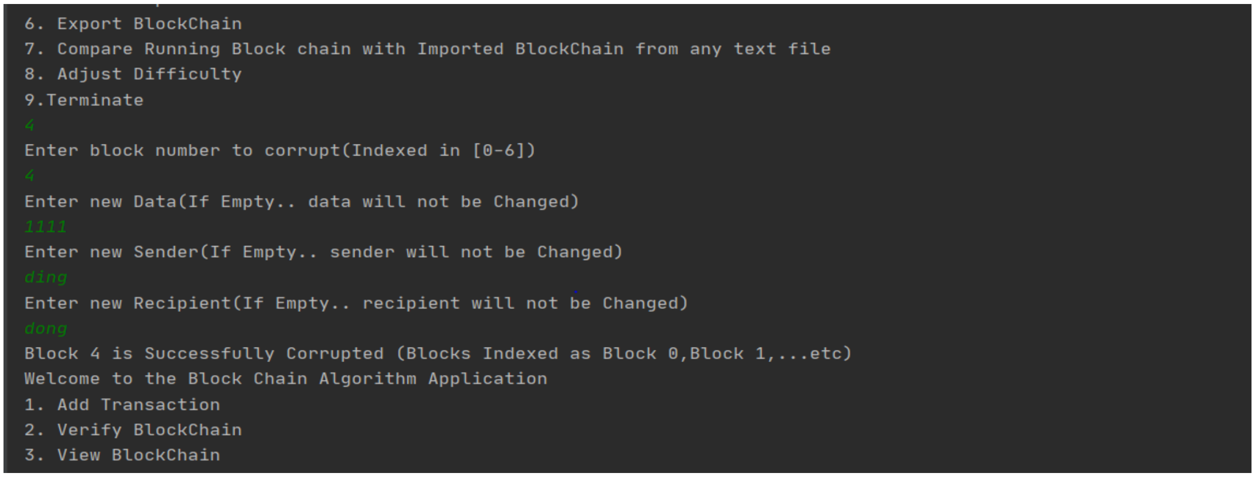
```
8. Adjust Difficulty
9.Terminate
6
Enter File name to export BlockChain(Eextension Not Required)
cs572update
Block Chain Exported Successfully to cs572update.txt
Welcome to the Block Chain Algorithm Application
```

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



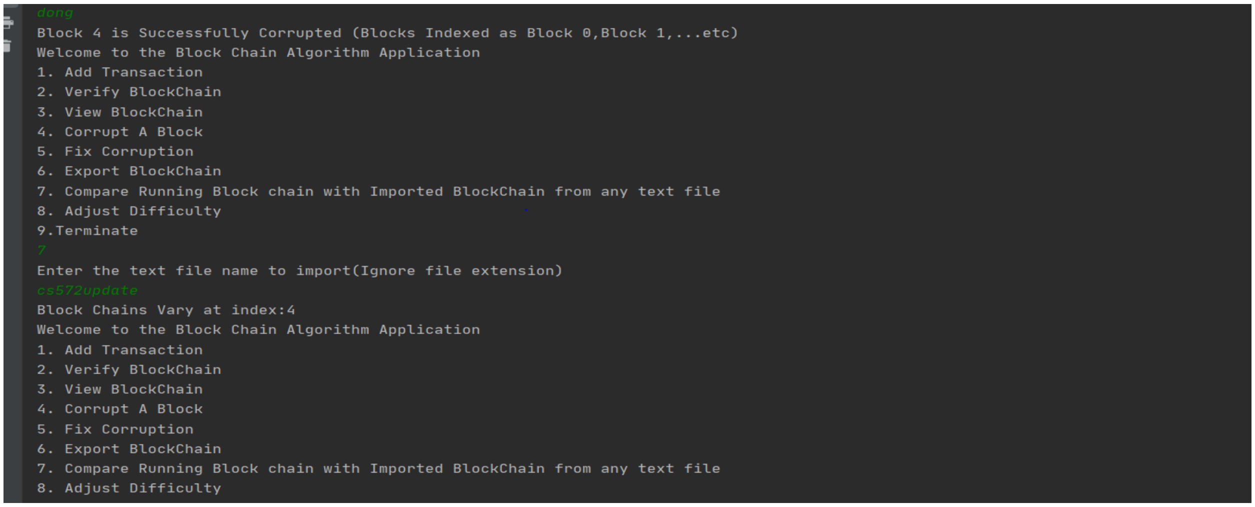
```
7. Compare Running Block chain with Imported Blockchain from any text file
8. Adjust Difficulty
9.Terminate
7
Enter the text file name to import(Ignore file extension)
cs572update
Block chains are Equivalent or Similar
Welcome to the Block Chain Algorithm Application
1. Add Transaction
```

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



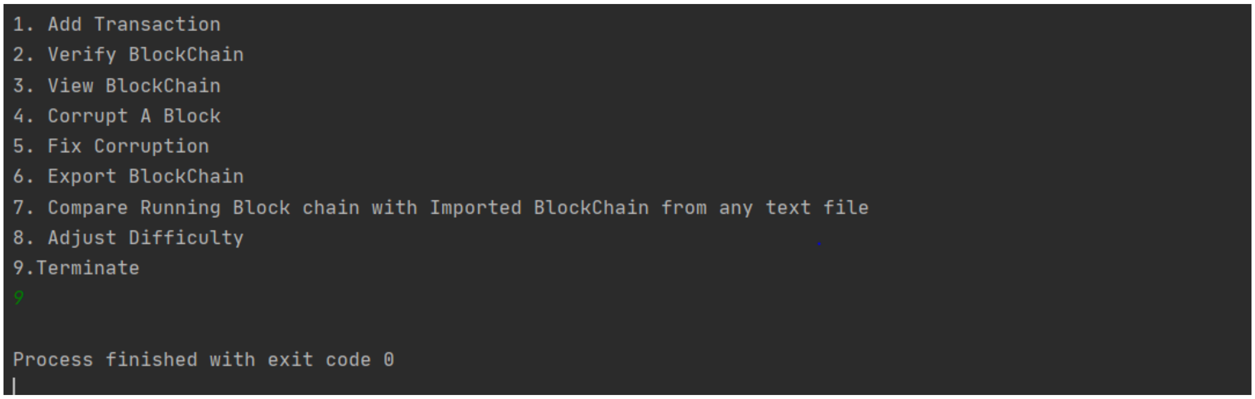
```
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
4
Enter block number to corrupt(Indexed in [0-6])
4
Enter new Data(If Empty.. data will not be Changed)
1111
Enter new Sender(If Empty.. sender will not be Changed)
ding
Enter new Recipient(If Empty.. recipient will not be Changed)
dong
Block 4 is Successfully Corrupted (Blocks Indexed as Block 0,Block 1,...etc)
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
```

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



```
dong
Block 4 is Successfully Corrupted (Blocks Indexed as Block 0,Block 1,...etc)
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
7
Enter the text file name to import(Ignore file extension)
cs572update
Block Chains Vary at index:4
Welcome to the Block Chain Algorithm Application
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
```

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



```
1. Add Transaction
2. Verify BlockChain
3. View BlockChain
4. Corrupt A Block
5. Fix Corruption
6. Export BlockChain
7. Compare Running Block chain with Imported BlockChain from any text file
8. Adjust Difficulty
9.Terminate
9

Process finished with exit code 0
```

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

Thank you