Blockchain and Cryptocurrencies Assignment
Coursework Description and Assessment

The deadline for submission is the 22nd March (2pm). You should submit several pieces of python3 code and a single pdf report through DUO. Please don't zip files together - just attach them to the DUO submission. Level 3 students should do Tasks 1, 2 and 3. Level 4 students should do Tasks 1, 2, 3 and 4.

Marking: the marks are all allocated to the content of the report. However, you must submit the code so that I can check you have generated the values yourself and that you have used only the specified python modules. I may also look at the code in order to allocate partial credit if the values in the report are incorrect. You will not be marked for the "quality" of the code, but if it is not clear and easy for me to understand, I may not be able to award you marks you might otherwise have got. The report itself only needs to contain the elements requested below; you do not need to include a general account of your efforts.

Task 1. Mining puzzles
You may use the python3 modules `hashlib`, `ecdsa`, `json` and `time`.

1.1 Create simple code to mine a valid block using **proof of work.** (20 marks)

You must edit the python3 code in `pow_mining_puzzle.py`
The code provides a template block header, and code to compute the double hash in a standardized way. (For simplicity I use json to serialise the header, even though in bitcoin itself the header is stored as a fixed length binary blob.)

The hash with my user id (dcs0mjb) and nonce zero should be:

```
71142c9a662d4e0740a2c2817ba2c68cce9e9718781a0711c0ace472414e1c3e
```

You must:

- Change the 'coinbase_addr' field in the block header to your own user ID (e.g. xyxy55).
- Calculate the correct target value for a difficulty of 0.001
  (Normal bitcoin difficulties are much greater than one, but I am trying to make it easier for you, so don't be put off by the difficulty value being less than 1.)
- Find a nonce which makes the block valid.

1.2 Create code to mine a valid block building using **proof of stake.** (20 marks)

Edit the python3 code in `pos_mining_puzzle.py`
The code provides the header of the previous block. You may assume that the baseTarget does not change between blocks.

You must:

- Change the 'effective_balance' value to the digits from your own user ID (e.g. user ID is xyxy75, effective_balance should be 75).

- Generate an ECDSA key pair.
- Demonstrate that your pair work by signing the message “Hello World” and verifying the signature. You can use verify.py to check you are doing this right.
- Compute your hit value.
- Determine how long (in seconds) after the publication of the previous block you would be able to forge a new block.

You must submit the **two edited pieces of code** as a .py files, and **in your report** give the following information:

1. Your user ID
2. (1.1) The block hash target you calculated in hex.
3. (1.1) The nonce value (as an int) of the valid block.
4. (1.1) The number of (double) hashes your code performed to find this nonce value and the time taken.
5. (1.1) An estimate of how long it would take your code to mine a block at the initial bitcoin difficulty (i.e 1) and at the 2018 peak bitcoin difficulty of 7,454,968,648,263. Include your calculations, clearly laid out, as well as the final values.
6. (1.2) Your ECDSA public key (in hex)
7. (1.2) The signature of “Hello world” (in hex). I will use verify.py to check this, so make sure yourself that it works!
8. (1.2) The signature used in calculating your hit (in hex)
9. (1.2) Your hit value (in hex). Include your calculation, clearly laid out, as well as the final value.
10. (1.2) The time in seconds when you would be able to forge a new block. Include your calculation, clearly laid out, as well as the final value.

I will be checking your signatures and hashes, so you must serialize things the same way as I have done. Use the example code to check. Please make your code clean and clear and easy to follow, with comments as necessary. Please make it easy to find the above elements in the report.

Task 2. Interacting with bitcoin-testnet

Create an account at blockcypher.com and get a “Token”

In Python3, use the module `blockcypher` (https://github.com/blockcypher/blockcypher-python). Documentation at https://www.blockcypher.com/dev/bitcoin/?python#documentation-structure

- First generate a new address for the bitcoin testnet, using the code `blockcypher.generate_new_address(coin_symbol='btc-testnet', api_key= 'd62…')` where the api_key is the token you obtained with your blockcypher account.
- Inspect the returned object, from which you should be able to extract a private key, public key and address.
- Go to `https://coinfaucet.eu/en/btc-testnet/` and request some ‘free’ bitcoins (only on the testnet LOL). Copy and paste in your address from above as the recipient of the coins.
- Now you have some bitcoins, create a transaction. The file `blockcypher_example_transaction.py` contains a template code for creating a transaction. You will need to fill in input and output addresses, and keys, and include the correct api_key where appropriate.

- You can edit fields of the unsigned transaction before signing it. Inspect the unsigned transaction and see what fields it has. You will need to edit the `script_type` and `script` fields in the tasks below.

2.1: Using a blockchain explorer examine blocks in the bitcoin blockchain whose height ends with the same last two digits as your user ID (E.g. Block#432475 is a block ending in 75). Find three transactions to describe, which are contained in such blocks and which have different structures from each other. (15 marks)

2.2: Create a transaction sending 100 Satoshis to the address
`mpamtqLA66JFVSQNDaPHZ5xMiCz6T2MeNn` (10 marks)

2.3: Create a transaction writing your student id into the blockchain! (15 marks)
Create a transaction as in 2.2, but rather than specifying an address as the output, instead specify the following:

```
outputs = [{'value' : 0, 'script_type':"null-data", 'script':""}]
```

where the script string is the hex encoding of a suitable script to create a **proof of burn** transaction with your student id in it.

A list of script op codes is available at `https://en.bitcoin.it/wiki/Script`

An example proof of burn transaction is
https://chain.so/tx/BTCTEST/33a27d1167a74f251e549e8ab46875a69c70db58bbd03a8555160a47c070ab27

You must submit the **two pieces of code** that generated your transactions as a .py files, and **in your report** include the following information:

1. Your user ID
2. (2.1) Links blockchain.com for the three transactions you have selected, e.g.
https://www.blockchain.com/btc/tx/79b5a7eb708bba96c876ee7abca348919994d30943ee32c70cbe23326b4f8662
3. (2.1) Explain each of the three transactions, their input and outputs scripts, and what, if anything, you can infer from their structure.
4. (2.2) The bitcoin-testnet address you made the transactions from.
5. (2.2) The transaction ID in hex, and a hyperlink to the transaction on chain.so
E.g.
https://chain.so/tx/BTCTEST/87d55ede04cf6b90a1d168b9bee671079f29910f49f9355a682d15cc0b7253b8

6. (2.3) The transaction ID in hex, and a hyperlink to the transaction on chain.so
7. (2.3) The hex you used as the script, and an explanation of how you put that hex script together.

Please make your code clean and clear and easy to follow, with comments as necessary. Please make it easy to find the above elements in the report.

Task 3. (20 marks)

In your report explain in 4-500 words whether you think it would be wiser to invest in bitcoin, NXT or gold (real physical gold) at the moment. **Justify your stance as far as possible.** (I am not actually going to judge your investment advice, but rather whether you have considered all the different factors, suitably evaluated them and come to a coherent conclusion.)

Task 4. (**Level 4 students only,** 50 marks)

Choose a cryptocurrency *other than* Bitcoin, Ethereum, Litecoin, BitcoinCash, NXT.

Include in your report an analysis of your chosen currency covering:

- the technical differences between your chosen currency and bitcoin;
- the idea behind the currency (why was it developed?);
- how it has performed;
- how it is mined, and the level of activity of miners;
- any notable attacks or events in history of the currency;
- your assessment of the currency: do you think it is a significant improvement over, or has some other use or value to bitcoin?

This analysis should be approximately 1.5-2 sides of A4 at 11 point with sensible margins.