Task 1

1. My User ID is: **bzqs27**

2. The **block hash target** I calculated was the following:
   3e7fc180000000000000000000000000000000000000000000000000000000

3. In order to attain a valid block I used a **nonce value** of 2943088.

4. The number of double hashes performed to achieve this nonce value is 2943089.
   The real-time taken to obtain the valid block is 63.70 seconds.

5. Estimate of how long it would take your code to mine a block at:

   (a) The initial bitcoin difficulty, 1: 92959.95 seconds.
   The following is an equation for the expected time to mine a block:
   $t = \frac{n}{R}$ where n is the number of hashes and R is the hash rate.
   The initial target defined at difficulty 1 is
   00000000FFFF0000000000000000000000000000000000000000000000000000
   This target has $4(8) = 32$ leading zeros, and as a result we would expect approximately 2^{32} hashes to be performed.
   Since earlier, I performed 2943089 in 63.70 seconds, we have a hash rate of approximately $R = \frac{2943089}{63.70}$.
   Using the above formula, we have:
   $t = 2^{32} \cdot \frac{63.70}{2943089} = 92959.95$

   (b) The 2018 peak bitcoin difficulty, $7,454,968,648,263$: $4.09 \cdot 10^{17}$ seconds.
   The target at difficulty $7,454,968,648,263$ is
   00000000000000000025c1910000001580000000000000000000000000000000
   This target has $4(18) + 2 = 74$ leading zeros, and as a result we would expect approximately 2^{74} hashes to be performed.
   Using the above calculated hash rate:
   $R = \frac{2943089}{63.70}$.
   As well as the above formula, we have:
   $t = 2^{74} \cdot \frac{63.70}{2943089} = 4.09 \cdot 10^{17}$

6. The ECDSA public key used is the following:
   0d059e547e831b396f67b41e96bc9f2366dbf3d26b6bcd9ac268a3cff9e81625
   aa38fc9cf387a752dd6d843a0afce44d9cd7ba56212f6f73ed019fcdc6889cf1

7. The **signature** of the message '**Hello world**' is the following:
   b0d338cc3c5a02c9eecd202bf2407a259e2b99fa72e6f12d8f7baa83f676a39a
   705a75ff0810d80387862b89f822b024fbacf69d7606d52dc0e602c5a0c86285

8. The signature used in calculating the below hit value is:
   d9e232243bd09634968c9363af96576a5d09c66caa64095a738a50db2f2f3b2a
   31258f5292e770fe16bdb73eba4921c010c5eedbef0398a434887688e2b79577

9. The **hit value**: **d8c1a43cd397bd9b**
   We obtain the hit value by performing the following steps:

- Sign the generation signature:
  9737957703d4eb54efdff91e15343266123c5f15aaf033292c9903015af817f1
  using our ECDSA private (signing) key,
  041bd9e2bd0dd67995f3ada692f38f5a8926d33b5fed7e0018de696522c9d09e
- The resulting signature is stated in (8). We hash the resulting signature (using SHA-256), to obtain the following:
  d8c1a43cd397bd9b8fdb96d80a874426c9a054997e212c90e4458227df79fd0f
- The hit value is then the first 8 bytes of the above, since each hexadecimal character consists of 4 bits, this is the first 16 characters:
  d8c1a43cd397bd9b

10. The time when I will be able to find a new block is 165541515.56 seconds.
    We obtain the time in the following manner:

    - Once our target value T is greater than the hit value, we may find a new block.
    - The target value T is $T = T_b \cdot S \cdot B_e$, where T_b is the base target, B_e is the effective balance and S is the time in seconds since the last block was found.
      In our case, $T = 1229782938247303 \cdot S \cdot 67$.
    - By the above, $T > \textit{hit value}$ occurs when $1229782938247303 \cdot S \cdot 67 > \textit{hit value}$
    - The decimal value of our hit value is 15618945563852520859.
    - So $T > S$ occurs when $S > \frac{15618945563852520859}{1229782938247303 \cdot 67} = 189.56$. So we're able to find a new block immediately after 189.56 seconds.
    - Since the timestamp of the previous block is 165541326, the time we will be able to produce a new block is the sum $165541326 + 189.56 = 165541515.56$.

Task 2

1. My User ID is: **bzqs27**

2. Transaction links:

   (a) https://www.blockchain.com/btc/tx/b17fc28c1dc15ca4b05bf4784419c145b14e79ece54c793

   (b) https://www.blockchain.com/btc/tx/4716f13815add4e071b38f5e0f9cb6ba0eab094c68a047b

   (c) https://www.blockchain.com/btc/tx/f602ce0777e04345c4e30aae1a8af0942e422f89b2832a3

3. (a) Link (a) presents a transaction at [Block #67].
   This transaction involves 50 BTC (newly generated coins) being awarded to the address **17uWtnxcJdpH7pEXXmr1SLrohVxe18B3fD**. Since no other transactions occur in the block there are no fees, this is a very early transaction occuring at 2009-01-10 22:21:12. Since this is a CoinBase transaction it needs no particular, however the miner can encode arbitrary data - in this case '04ffff001d0173'. The output script is the following

   PUSHDATA(65)[04ea258025b3a5fa9352ab84951c1141affb5d7e0364ab18ed65d7f970b9af1c d7f9550cb69a016480d825e4e112ba9f16c717ec3a421c02155de4425a2066d208] CHECKSIG

   This is a Pay To Pubkey, or P2PK script that locks an output to a public key (a simpler version of the more commonly used P2PKH). The PUSHDATA(65)[PUBLIC KEY] section of the script pushes the $65 - \textit{byte}$ public key to the user. To unlock it, you need only provide a valid signature. When the script runs the CHECKSIG opcode it compares the signature with the public key, and pushes a 1 on the stack if it is valid. This ensures that only a miner with the correct public key would have access to the block.

(b) Link (b) presents a transaction at [Block #500067].
Similarly to the above this transaction also presents newly generated coins being awarded to an address. In this case 16.72698693 BTC are awarded, at 2018-12-26 13:06:54. Since mid-2016 the award for mining a bitcoin has been 12.5 BTC, thus the additional 4.22698693 BTC are for fees, for other transactions in the block. This indicates there are a large number of transactions, as the fee is relatively large compared to the reward value.

Since mid-2012, as a result of BIP 0034, the CoinBase script must start with a push of the height of the block, this is in order to prevent CoinBase transactions from having the same transaction ID. In this case we have that the CoinBase script hex is:

0363a10741d68e2247eee3b541d68e22478976a12f4254432e544f502ffabe6d6d3081b836aaba4e
c4a30706e4102b8d2ff056bc061059377dd1682e527605fcff8000000000000000f8000e6c1654010
000000000.

By the protocol the first byte '03' indicates that the next 3 bytes present the height of the block. This enforces that each transaction is unique. In our case these are, $63a107$, converting this from little-endian gives us $07a163$ and in turn converting this to decimal gives us 500067 - the height of the block. The remainding string of hexadecimal is arbitrary, and essentially meaningless in our case.

We have the following 2 output scripts:

DUP HASH160 PUSHDATA(20)[ba507bae8f1643d2556000ca26b9301b9069dc6b]
EQUALVERIFY CHECKSIG

This is a Pay To Pubkey Hash or P2PKH script , a more commonly used locking script than the aforementioned P2PK script. It follows similar principles to P2PK however, rather than using the miner's public key, it uses the hash of the miner's public key. P2PKH allows us to fund an arbitrary transaction, no matter how complicated, using a 20-byte hash. Pay-to-Pubkey-hash addresses are similarly a 20-byte hash of the public key. EQUALVERIFY returns True if its inputs the Total In/ Total Out of the transaction) are exactly equal, and if they are not equal it marks the transaction as invalid.

RETURN PUSHDATA(36)[aa21a9edbaf8d06742c94e5d7dab23cbbe93e4757f02b19724913cf
5abf820654c3bba76]

This script pushes 36-bytes, i.e. 72 hexadecimal characters to the stack. In this case the data pushed is the Merkle root of the witness tree. This is part of Segregated Witness a protocol upgrade introduced in 2017, which changes the way that data is stored. It enables a larger number of transactions within the $1MB$ blocks, $1MB$ was the maximum block size at the time.

(c) Link (c) presents a transaction at [Block #56476].
This transaction consists of a payment from address **3PbJsixkjmjzsjCpi4xAYxxaL5Nn xrbF9B** to address **3MBiG7WKXeYE8GRaYzXWeR3sr8aY171o3K** of 0.0299 BTC.

This transaction has multiple input and outputs (2 in each case).

The address **3PbJsixkjmjzsjCpi4xAYxxaL5NnxrbF9B**, provides two inptus. One of which is a value of 0.02419667 BTC, and another is 0.09040641 BTC. Thus the total input value is 0.11460308 BTC.
A value of 0.08120308 BTC is sent back to the address **3PbJsixkjmjzsjCpi4xAYxxaL5NnxrbF9B**, I think of this as the original address receiving 'change' for paying too much, and a value of 0.0299 BTC is transferred to the address **3MBiG7WKXeYE8GRaYzXWeR3sr8aY171o3K**. The output value amounts to 0.11110308 BTC, the remaining 0.0035 BTC is spent as miner fees. Thus the total input value is equal to the total output value, which we require.

The 2 inputs are both of the following form:

ScriptSig: PUSHDATA(22)[0014f7bae6ee31d59e79da8e857fb63429637f9e0d57]
WItness: $02473044022056f21b60c6d7443546ceff68db83f16fd7bbf94\ldots$

This another segwit transaction, and as expected we see the traditional scriptSig alongside 'Witness' encoding which is introduced by this protocol upgrade. Both input scripts are of the same form, simple with different 'Witness' encodings.

The 2 outputs are:
HASH160 PUSHDATA(20)[d5d7b573229379eb66f0cb98c112b715b32e2731] EQUAL
HASH160 PUSHDATA(20)[f03e6bf9b389bbd5d5669ff55c4dba30de995535] EQUAL

This output makes use of Pay-to-Script-Hash, here the 20 bytes of data being pushed is a redeem script. P2SH allow transactions to be sent to a script hash (address starting with 3) instead of a public key hash (addresses starting with 1). When redeeming the values, a HASH160 is performed on the script used in the transaction, and this is compared with the script provided by the recipient. To spend bitcoins sent via P2SH, the recipient must provide a script matching the script hash and data which makes the script evaluate to true. This adds extra security to the transaction.

4. The bitcoin-testnet address I made the transactions from is:
   mqCYbVBd7jYKtGLoJMbfwWyi1PFFrxuVGA

5.
   - The transaction ID in hex is:
     f9e69cf2326a23dd65942bfcc0b15db73d38d7abb1f0ea9778f018da054eabb6
   - This is a hyerlink to the transaction:
     https://chain.so/tx/BTCTEST/f9e69cf2326a23dd65942bfcc0b15db73d38d7abb1f0ea9778f

6.
   - The transaction ID in hex is:
     a83c94074a3c9205ca68361466ae6cf5197ad94b988c7c4be6f97b0618322312
   - This is a hyerlink to the transaction:
     https://chain.so/tx/BTCTEST/a83c94074a3c9205ca68361466ae6cf5197ad94b988c7c4be6f

7. The hex script used is: **6a4c06627a71733237**
   The general format of this hex script is:
   OP_RETURN + OP_PUSHDATA1 + number of bytes + encoded user ID.

Using https://en.bitcoin.it/wiki/Script, we see that 'A standard way of attaching extra data to transactions is to add a zero-value output with a scriptPubKey consisting of *OP_RETURN* followed by exactly one pushdata operation'.
Thus I chose to use *OP_RETURN* alongside *OP_PUSHDATA1*, the description of *OP_PUSHDATA1* tells us the next byte contains the number of bytes to be pushed onto the stack. Since there are 6 characters in **bzqs27**, the next byte is the hex encoding of 6, which is 06.
The hex encoding my user ID is 627a71733237, thus we have the following:
The hex code for *OP_RETURN* is 6a.
The hex code for *PUSHDATA1* is 4c.
The number of bytes is 6, which has hex code 06.
The hex encoding of my user ID **bzqs27** is 627a71733237.
Concatenating the above information together gives the above script, **6a4c06627a71733237**.

Task 3

Gold has an intrinsic value due to its properties as an element and its uses in industry. There is a finite amount of gold, and no one is making anymore. In times of economic uncertainty people often invest in gold due to its reliable reputation, increasing its value. Gold is a physical object and as a result, no one can steal it from miles away as they can with cryptocurrencies. Throughout history, gold and money have been seen as synonymous, and thus Gold is seen as a reliable and low-risk store of value. However, gold has a history of low capital gains, and its value rarely increases significantly.

Bitcoin utilises cryptographic techniques to ensure that bitcoins can't be copied or counterfeited - we can verify that a bitcoin is real in seconds. The number of bitcoins is capped at 21 million, and many people expect that the value of bitcoin will increase as the number of bitcoins entering the system decreases. What happens to bitcoin when there is no more bitcoin left to mine is uncertain, and people are very wary about what will happen to their 'virtual savings'. Bitcoin is seen as less universal than some currencies, as many countries refuse to use it and have even made it illegal. Bitcoin is reliant on 3rd party software for wallets and exchanges, and people/organisations are constantly trying to exploit vulnerabilities in these 3rd parties. Not only this but bitcoin exchanges are not regulated by the government and generally do not provide enough insurance and security to be used to store money in the same way as a bank.

NXT utilises 'Proof of Stake' meaning the hardware requirements to 'forge' NXT are much less than to 'mine' Bitcoin. The only way of attaining NXT in a Proof of Stake system is from a number of people who already have them, initially 1 billion NXT were created and distributed across 73 investors. Many of these initial investors still hold significant proportions of the NXT and have huge influence over the system. Unlike Bitcoin, NXT does not rely on third parties and instead provides its own wallet software that uses cold storage. NXT facilitates a range of applications that make use of the blockchain, and it could secure its future as more than just a cryptocurrency. The future of NXT is also very uncertain due its low market cap.

All cryptocurrencies are seen as a financial risk, and in NXT and Bitcoin, this is supported by a history of massive fluctuations in value. Whilst an investment in Gold is unlikely to reap the huge financial gain that an investment in Bitcoin or NXT could, it is also less likely to produce the sudden losses that investments in cryptocurrency often do. Whilst many cryptocurrencies have become obsolete with a lack of backing and advancements in technology, Gold will always have some intrinsic value. As a result, I feel Gold is the best long-term investment, but the best investment on a case-to-case basis is dictated by the investment parameters.