All answers highlighted in Yellow

Problem 1

a. A transaction ID is a double-entry bookkeeping ledger. A transaction ID contains information about the inputs and outputs between the two parties involved regarding the bitcoins being transferred.
b. The transaction fee for the transaction in BTC is the implied fee that is taken and amounts to the difference between the input and output. The transaction fee incentivizes the miner to include it in a block and putting that block on the block chain ledger.
c. The total value of the block chain containing my transfer was 14,487.20964744 BTC.
d. It took approximately 30 hours to have 3 confirmations because the time between the execution of my transaction and the first confirmation was 10 minutes.

Problem 2

a. The bitcoin addresses of potentially 3 other students in our class are
   1DuA2rvcJmgBLs9TVE14chN94mwsnW8xEe
   1PQHK5ivZGyRf83TDZsvhnmecxKgCK4JXa
   1FqQV1R3H8dftDupCjCoW7b6kSVxoBZBmh
b. I traced the bitcoin back to 19tQyNBkgBQv93mk7L9UYgYc8tho2mzn7o in transaction
   a9ce101b0ecdf0f8e955033f68adeb8c8a201594acefa93327d57922f7ddefcd
c. The sender's probable location is on the East Coast close to the state of Denver based on the information in the block chain.

Problem 3

a. An evil wallet might be created by memorizing all the private keys that would otherwise be stored in some sort of wallet. The most secure way to place bitcoins in an evil wallet is in cold storage.
b. My confidence in the security of my bitcoin wallet is actually lower than that of storing it in my bank. This is so because in the case that my current bank (Bank of America) collapses, I doubt that people would accept bitcoin more readily. The price of a bitcoin has also proven to be much more volatile over the past five years than most world currencies. Since it is not backed by a government, there is always the risk that a more technologically advanced currency could replace it or that the bitcoin system is hacked.

Problem 4

I verified it by wolfram alpha compared to the hexadecimal converted to decimal from the code. It looks like they are exactly equal.

Decimal from code after hexadecimal converter:

115792089237316195423570985008687907853269984665640564039457584007908834671663

From Wolfram Alpha Decimal Computation:

115792089237316195423570985008687907853269984665640564039457584007908834671663

Problem 5

All the things that I need to trust if I am going to send money to the key generated by running keypair.go are that it implements it correctly, the computer's accuracy in computation, the private key is not a commonly used number, and that the overall implementation of the code is truly random.

Problem 6

generateVanityAddress (pattern string) (*btcec.Publickey, *btcec.PrivateKey) {

generatekeyPair(*btcec.Publickey, *btcec.PrivateKey);

}

Problem 7

My bitcoin vanity address: 1PeterbRjoyDWfxtPQS2txeke8yxaYfXp5

Problem 8

My vanity address is less secure than the first address I generated because it contains personal information and is more likely to contain dictionary words. Therefore, if someone were to hack my address they would be more likely to hack my vanity address.

Problem 9

Transaction ID be037c9fe011353bbde94adfa52bb1eb9637b9ed997c2da248977b8f95a0085b

Problem 10

Transaction ID c43fd82b034b6516d5424a0206972c0eadeff8ed9aa93d357aa296a687fc9faf

Problem 11

Transaction ID 3ea041a39f8a5b05b0c6848aa5f5c07433f2b0f7db8e4000400703ba42a1dc7f

f774f516d88f0070ab91cddc688a1e06bdd34d7b01d55fe06f0aa3fdee650e02