1. Please compare hash function and cryptographic hash function and give an example.

hash function一種從任何一種資料中建立小的數字「指紋」的方法。雜湊函式 把訊息或資料 (key) 壓縮成摘要，使得資料量變小，

將資料的格式固定下來。該函式將資料打亂混合，重新建立一個叫做 雜湊值（hash values，hash codes，hash sums，或hashes） 的指紋。

這個雜湊值就當作是陣列的索引，資料就儲存在這個索引的位置中。雜湊值通常用一個短的隨機字母和數字組成的字串來代表。

例如Roy=>(Hash function)=>4928mod5=3, list[3]='Roy'

cryptographic hash function，則是強調不可逆，例如SHA256。

2.

a. Can you print the private/public key with hex string representation? Please give us an example.

```
// privKey
const privKey = wallet.getPrivateKeyString();
console.log("privKey:", privKey);

// pubKey
const pubKey = wallet.getPublicKeyString();
console.log("pubKey:", pubKey);
```

```
privKey: 0x979eba610dcddf7b25d8ad6320040eff8aab5be413e7efcfaa43ec7a3fdf23b7
pubKey: 0x0d74b0d36f769dd6b3ad144e86e26818f0e595995546c5e4b8d1023dcff673c73ec0b99c487b9ff1216c42ebdf6a38ecb62327e17ab5705b5282a54ffa367ca8
address: 0x55e9e16c2a15d04c8101d6f9e65f768f86a31bc2
```

b. In addition, if we don't want to use the getAddressString() to get the address, how can we obtain the address by hashing the public key?

```
const public_key_hash  = keccak256(pubKey)
const address = "0x" + public_key_hash.slice(-40)
console.log("address:", address);
```

```
➜  ~ node key.js
privKey: 0xeaa1b6a4e6c0ac77e098fb212fe461a9be2e096a81790625e74d7450847199b2
pubKey: 0xfa11b25696a2faee9c884b7d976a745a7d0198fcb2c163422444eb04e170fdd0c7e00634057e8e6104d621a16c90582b53dff181f98916efae74f50396eb10e2
address: 0xc83a56d3496b79bbfd8b18d6d22a1983a4ffc0a9
```

c. There is a file called Keystore that is used to encrypt the private key and save in a JSON file. Can you generate a Keystore with the password "nccu"? You can find the details about Keystore below.

```
//JSON
const keystore = wallet.toV3("nccu");
console.log("keystore:",keystore);
```

```
➜  ethhw node 3.js
privKey: 0x4f9580797925f535dcd8b7ad884cd51d76f6388279abd83a4b613a0d8a4cc5ee
pubKey: 0x2f0442cb129618b6ff9d8bb5b01213f4f5f95cde466f831c72ced277bb00bcc7860fd399d6e6ab761d726457715f
b44c05c0e17aa7f3ac5a56e478d062eb2c6e
address: 0x1a0712b08cac4614aed0e77325dad17a0ae9ab2d
keystore: { version: 3,
  id: '172d35a8-f258-4fed-bfb2-a42628a7fbb2',
  address: '1a0712b08cac4614aed0e77325dad17a0ae9ab2d',
  crypto:
   { ciphertext:
      '303b33b342296de72034c4b154e8a0423220c54b7fd1fe49fcdae33445ddab4f',
     cipherparams: { iv: '9e3b5d905287f99513ae78b03d9f7de8' },
     cipher: 'aes-128-ctr',
     kdf: 'scrypt',
     kdfparams:
      { dklen: 32,
        salt:
         '1087841286771a91a03d0dee53e755752ac5d7f5cebe504b1acf94e9b60e8a96',
        n: 262144,
        r: 8,
        p: 1 },
     mac:
      '72a851844a92484f1176859f15037ae0c05c13bfe69a94d94701c0e2a94c116d' } }
```