Create a new blockchain from scratch

```
import hashlib

def hashGenerator(data):
    result=hashlib.sha256(data.encode())
    return result.hexdigest()
```

```
class Block:
    def __init__(self,data,hash,prev_hash):
        self.data=data
        self.hash=hash
        self.prev_hash=prev_hash
```

```
class Blockchain:
    def __init__(self):
      hashLast=hashGenerator('gen_last')
      hashStart=hashGenerator('gen_hash')

      genesis=Block('gen-data',hashStart,hashLast)
      self.chain=[genesis]

    def add_block(self,data):
        prev_hash=self.chain[-1].hash
        hash=hashGenerator(data+prev_hash)
        block=Block(data,hash,prev_hash)
        self.chain.append(block)

```

```
bc=Blockchain()
bc.add_block('1')
bc.add_block('2')
bc.add_block('3')

for block in bc.chain:
    print(block.__dict__)

```

```
{'data': 'gen-data', 'hash': '0a87388e67f16d830a9a3323dad0fdfa4c4044a6a6389cab1a0a37b651a5717b', 'prev_hash': 'bd6fecc16d509c74d23b04f00f936705e3eaa907b04b788720446076650184 77'}
{'data': '1', 'hash': 'e3e6c97161f3deaf01599fda60ba85593b07f70328bf228473d1d408f7400241', 'prev_hash': '0a87388e67f16d830a9a3323dad0fdfa4c4044a6a6389cab1a0a37b651a5717b'}
{'data': '2', 'hash': '47e8645e3c14bd4034a498aa88ea630bc0793375207bf90ca469792a5d9484e1', 'prev_hash': 'e3e6c97161f3deaf01599fda60ba85593b07f70328bf228473d1d408f7400241'}
{'data': '3', 'hash': '82084603decb1a14a8819dacaa86197659f1e150c4a50186e68043004b5a3c06', 'prev_hash': '47e8645e3c14bd4034a498aa88ea630bc0793375207bf90ca469792a5d9484e1'}
```

```
# How to mine a block
```

```
from hashlib import sha256
MAX_NONCE = 100000000000

def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()

def mine(block_number, transactions, previous_hash, prefix_zeros):
    prefix_str = '0'*prefix_zeros
    for nonce in range(MAX_NONCE):
        text = str(block_number) + transactions + previous_hash + str(nonce)
        new_hash = SHA256(text)
        if new_hash.startswith(prefix_str):
            print(f"Successfully mined bitcoins with nonce value:{nonce}")
            return new_hash

    raise BaseException(f"Couldn't find correct has after trying {MAX_NONCE} times")


```

```
if __name__=='__main__':
    transactions=" Alice ->Bob -> 20, James -> Jacob ->45"
    difficulty= 3 # try changing this to higher number and you will see it will take more time for mining as difficulty increases
    import time
    start = time.time()
    print("start mining")
    new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edbe038f81208fecf9a66be9a2b8321c6ec7', difficulty)
    total_time = str((time.time() - start))
    print(f"end mining. Mining took: {total_time} seconds")
    print(new_hash)


```

```
start mining
Yay! Successfully mined bitcoins with nonce value:604
end mining. Mining took: 0.0014979839324951172 seconds
000922034e04ce27731e27cf7cc18125f26d73cba526f335bf946b1eccd9592b
```

```
print(SHA256(transactions))
```

```
f9ff13061d2253e0681b186255f5cc6a1e94147004b1f5cb7e5b84805469203a
```

```
1
```

```
A
```

1