

CryptoCurrencies Project#1

Alireza Arbabi

810199334

Section 1) Bitcoin Address Generator

Q1&2

In the AddressGenerator.py class, there is a class called **PublicAddressGenerator**, which can be used to create bitcoin Address. By putting **False** in the argument of the class constructor, an address will be generated, and it will be printed with the corresponding private and public key (the x & y points on the secp256k1 curve). By putting **True** in the argument of the class constructor, the program asks for a 3 character vanity string, and then the program tries to create the corresponding vanity Address.

Q1- Simple Address:

```
alireza@alireza:/media/alireza/C0BEE86EBE85F02/UT/Lessons/Term6/Crypto/Projects/P1$ python3 AddressGenerator1.py
private key: 1ddde3818859bbbf9c6d273669a0607f36b43cae502315ea90d9fcd8247f9f51
WIF Address: 4iznTrweKwFc0atnK7r1AHYTH6zSD9nZVH8Kp3nz7oHSyu74aT
public key(x , y): (5760521774898071108218450675151455341595881182040583407763146040340619445741, 78344610854722498518420633679590242264371415693973919019463670048845601097566)
Address: mqXPrsQfx42xMd5trUgrqqN1u866A193FB
alireza@alireza:/media/alireza/C0BEE86EBE85F02/UT/Lessons/Term6/Crypto/Projects/P1$
```

Q2- Vanity Address:

```
alireza@alireza:/media/alireza/C0BEE86EBE85F02/UT/Lessons/Term6/Crypto/Projects/P1$ python3 AddressGenerator1.py
Enter your vanity string (3 char): zql
private key: 09dc50e8c3e910c3d3824779729da2707fd5dd183a40fa42ecf120724f9a3728
WIF Address: 4j5hgo3dhtbhuMk8Pw456TJ1Fq3okaFpNaqTomykcekazQYPu1
public key(x , y): (36899279086099797980690573164977193859101170017289608890988711635144476784149, 83580453543266980018482490919872697019834995474985656276112144869339029245768)
Address: mzqILenPjbt45trKEUdcRcm6ID1yVNZYp
alireza@alireza:/media/alireza/C0BEE86EBE85F02/UT/Lessons/Term6/Crypto/Projects/P1$
```

Section 2) Creating Transaction

Q1-P1)

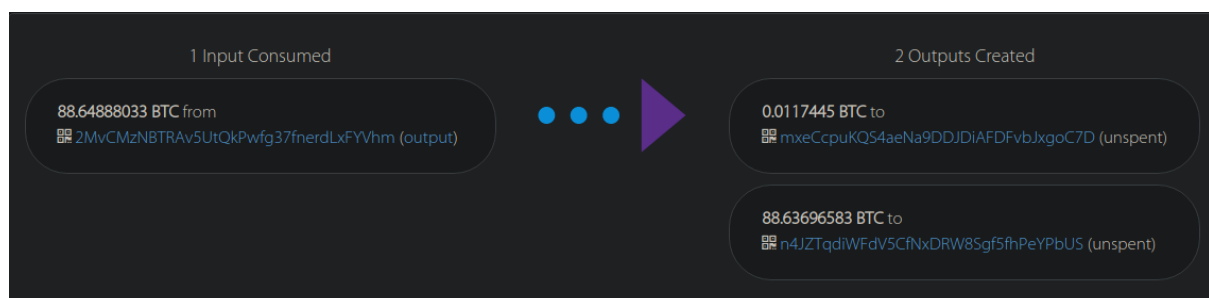
Our main address data:

```
private_key: 931cPqJ5igurei6mQSQYfte9buw3KGF6VERz9hDekukF5Dcwf23
address: mxecCpuKQS4aeNa9DDJDiaFDFvbJxgoC7D
```

Our second address data:

```
private_key: 93S78wP3T666D8wLtccQ79bsys631vSR64TMLKVishP6wG9m8Yp
address: myonnzBhpPdX6q5JymPg3jrkW2hbQmF4B
```

Now here we get some coins from coinfaucet.eu to the main user address:



Here is the [Link](#) to the transaction page. After receiving bitcoin from the faucet, we were asked to create a transaction with 2 outputs, an unspendable output, and an output which can be spent by everyone. We can create the first output by only putting OP_RETURN script in the public

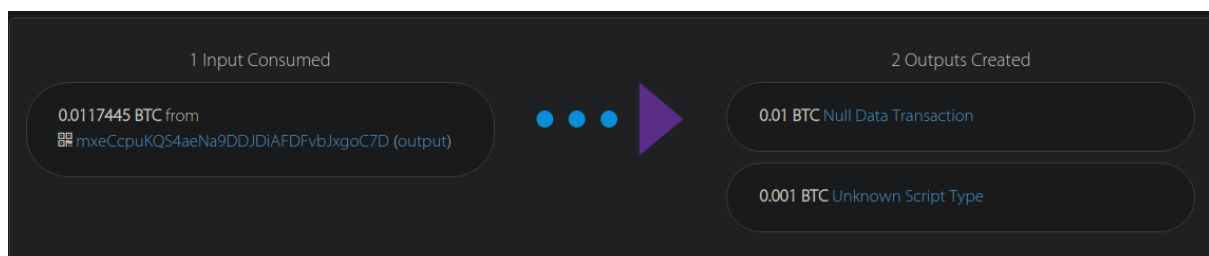
script of the first output, and we can create the second output by putting only OP_CHECKSIG script in the public script of the second output.

We can execute this transaction by running Q1P1.py:

```
alirez@alireza:/media/alireza/C08EE86EE85F02/UT/Lessons/Term6/Crypto/Projects/P1$ python3 Q1P1.py
mxCcPuKQ54aeNa9DDJDIAFDfVbJxgoC7D
940198c68078d8f7c65492606f3f316db59b842d2b1f6b6634a2b681d9b1e76426c59ae41ac75c08961e3d2a8926a91bedd56cb318cc0902a32bb35931b2fcbe1
bbc41aacf38a9f4321dccc22e4b2282d1732b97f8cf2865b20ff58739cb013db2
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "717708b1bb42a25f349bdd98eaa1beda361bce4a900f1c97e6cf5158f5860fed",
    "addresses": [
      "mxCcPuKQ54aeNa9DDJDIAFDfVbJxgoC7D"
    ],
    "total": 100000,
    "fees": 74450,
    "size": 210,
    "vsize": 210,
    "preference": "high",
    "relayed_by": "2a01:5ec0:1000:16d1:313a:e8e1:f994",
    "received": "2023-05-19T12:12:24.7330996Z",
    "ver": 1,
    "double_spend": false,
    "vin_sz": 1,
    "vout_sz": 2,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash": "5e6c82a0afb6618e9ff86cc2b2b6dcd10fb7ef10d1efb39b9079adcc46e2e2",
        "output_index": 0,
        "script": "483045022100cc8c978fdd7f94272b9626f5dcbd1ca408e5e7a231f506c6e75d5064e20c4d0b02286c797eb9c5f27f33aca2a19f0d908f24dc4f57438923ad53535d376ef8f055cab60141045c24daec13ef214b3b05392a83c065f77c9ebb8947a3d3010bf5e6df038558369ceb163f7303e018280fd92fc23d8119acadb2f393136d4cc1a8171b9d59d84f",
        "output_value": 1174450,
        "sequence": 4294967295,
        "addresses": [
          "mxCcPuKQ54aeNa9DDJDIAFDfVbJxgoC7D"
        ],
        "script_type": "pay-to-pubkey-hash",
        "age": 2434107
      }
    ],
    "outputs": [
      {
        "value": 1000000,
        "script": "6a",
        "addresses": null,
        "script_type": "null-data"
      },
      {
        "value": 1000000,
        "script": "ac",
        "addresses": null,
        "script_type": "unknown"
      }
    ]
  }
}
```

So here is the new transaction with the above features: ([Link](#))

Transaction Hash: 717708b1bb42a25f349bdd98eaa1beda361bce4a900f1c97e6cf5158f5860fed

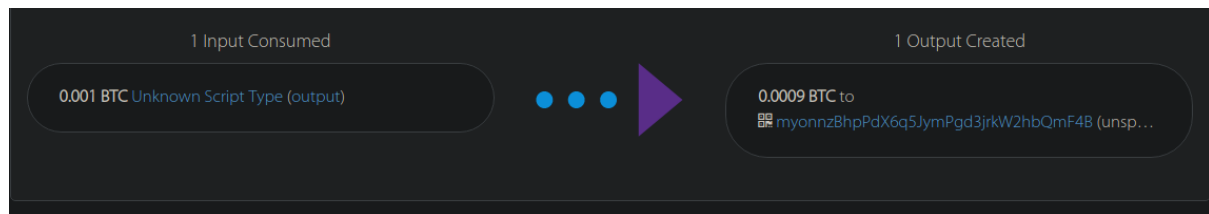


As described before, the second output of the latest transaction can be used by everyone. So we send it back to our another address (address 2) by creating another transaction. This is the output of the Q1P2.py file:

```
alirez@alireza:/media/alireza/C08EE86EE85F02/UT/Lessons/Term6/Crypto/Projects/P1$ python3 Q1P2.py
myonnzBhnpDx6q5JymPgD3JrkW2hbQmF4B
045c24daec13ef214b3b05392a83c065f77c9ebb8947a3d3010bf5e6df038558369ceb163f7303e018280fd92fc23d8119acadb2f393136d4cc1a8171b9d59d84f
f202d0aa5974e971e3b08d13991b0d66bae5ee01a82fb0f08023a92f8e407d9
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "5e642fe24ba79f326aac43d9b49bdd172769db4483d048b90103d25e8317a401",
    "addresses": [
      "myonnzBhnpDx6q5JymPgD3JrkW2hbQmF4B"
    ],
    "total": 90000,
    "fees": 10000,
    "size": 224,
    "vsize": 224,
    "preference": "low",
    "relayed_by": "91.98.52.47",
    "received": "2023-05-26T08:32:30.507247759Z",
    "ver": 1,
    "double_spend": false,
    "vin_sz": 1,
    "vout_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash": "717708b1bb42a25f349bdd98eaa1beda361bce4a900f1c97e6cf5158f5860fed",
        "output_index": 1,
        "script": "483045022100cc8c978fdd7f94272b9626f5dcbd1ca408e5e7a231f506c6e75d5064e20c4d0b02286c797eb9c5f27f33aca2a19f0d908f24dc4f57438923ad53535d376ef8f055cab60141045c24daec13ef214b3b05392a83c065f77c9ebb8947a3d3010bf5e6df038558369ceb163f7303e018280fd92fc23d8119acadb2f393136d4cc1a8171b9d59d84f",
        "output_value": 100000,
        "sequence": 4294967295,
        "script_type": "unknown",
        "age": 2434108
      }
    ],
    "outputs": [
      {
        "value": 90000,
        "script": "76a914c8a1677f8771c7583d86c451c545719d0a00174188ac",
        "addresses": [
          "myonnzBhnpDx6q5JymPgD3JrkW2hbQmF4B"
        ],
        "script_type": "pay-to-pubkey-hash"
      }
    ]
  }
}
```

Here is the overview of the transaction above: ([Link](#))

Transaction hash: 5e642fe24ba79f326aac43d9b49bdd172769db4483d048b90103d25e8317a401



Q1-P2)

3 new generated addresses:

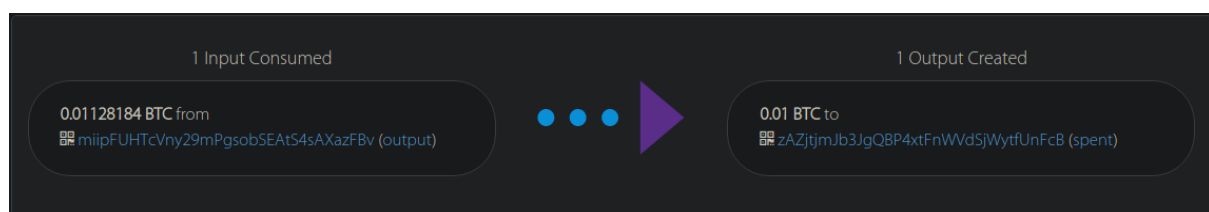
```
private_key 0 : 93R6C1uspBV9RfTNXjcl0EFZXA494HMjZBNUpKueZhPx2iFVAWG
address: miipFUHTcVny29mPgSobSEAtS4sAXazFBv
private_key 1 : 92h8FzaffymGV8JZgYtTwNmeK2QNVXvKEms7yVtDUoyRtHpG2YN
address: mjLRVq15q6Qwpn4fQyzSLsjzYCq6aLRmjs
private_key 2 : 91rEuvZai6xBjbSAX4qdL4tWBj8ANktypVhKkjwQciUfB71noZx
address: mhPavQtru2CbGb3acy14ArLC8w95MRBSvw
```

First we should send some coins to an address, so we receive 0.11 from a faucet to the “miipFUHTcVny29mPgSobSEAtS4sAXazFBv” address. Now we have to create a 2-of-3 P2MS transaction. We can do it by executing the Q2P1.py file:

```
alirez@alirez: /media/alirez/CORRE060EE85F02/UT/Lessons/Term0/Crypto/Projects/P1$ python3 Q2P1.py
miipFUHTcVny29mPgSobSEAtS4sAXazFBv
049e6f446635d9b49c7493f4f2d820cbc46144ffbc95f8a56eb60aa30a3b3aed318fde4d7f8845d73ff96bd1486c9d277566515e1dd9dc1263002dc0393c1207
f1141aeaf38d9f43210cc22e4baa82d1732b97f8cfbda3b20f158739cb013db2
201 Created
{
  "tx": {
    "block height": -1,
    "block index": -1,
    "hash": "4ad64a8fc862f43495af8bd713f689eb18450d77e6b23fcd5188cc99b02dbca9",
    "addresses": [
      "miipFUHTcVny29mPgSobSEAtS4sAXazFBv",
      "ZAZjtjmJb3JgQBp4xtFnWVdSjWytUnFcB"
    ],
    "total": 1000000,
    "fees": 128184,
    "size": 399,
    "vsize": 399,
    "preference": "high",
    "relayed by": "28015sec0:1000:76ae:7450:ba8b:cd1e:854e",
    "received": "2023-05-10T13:26:29.597388435Z",
    "ver": 1,
    "double spend": false,
    "vin_sz": 1,
    "vout_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
        "prev hash": "ba776a8df2131ed019d6213167d14b273dcdb109f5421b00eb9f96766ce06d",
        "output index": 0,
        "script": "4730440220683f9559c1a07de9cd0cf1b1500baf130617e5f903afe7c46fd2fda132ead302204e22ff14170a1d0751e5f8794514cbs5bac923cfd9d209f2a213fc7490d87014049e6f446635d9b49c7493f4f2d820cbc46144ffbc95f8a56eb60aa30a3b3aed318fde4d7f8845d73ff96bd1486c9d277566515e1dd9dc1263002dc0393c1207",
        "output value": 1128184,
        "sequence": 4294967295,
        "addresses": [
          "miipFUHTcVny29mPgSobSEAtS4sAXazFBv"
        ],
        "script type": "pay-to-pubkey-hash",
        "age": 2434112
      }
    ],
    "outputs": [
      {
        "value": 1000000,
        "script": "5241049e6f446635d9b49c7493f4f2d820cbc46144ffbc95f8a56eb60aa30a3b3aed318fde4d7f8845d73ff96bd1486c9d277566515e1dd9dc1263002dc0393c120741049e6f446635d9b49c7493f4f2d820cbc46144ffbc95f8a56eb60aa30a3b3aed318fde4d7f8845d73ff96bd1486c9d277566515e1dd9dc1263002dc0393c1207",
        "addresses": [
          "ZAZjtjmJb3JgQBp4xtFnWVdSjWytUnFcB"
        ],
        "script type": "pay-to-multi-pubkey-hash"
      }
    ]
  }
}
```

Link of the transaction below: [Link](#)

Transaction Hash: 4ad64a8fc862f43495af8bd713f689eb18450d77e6b23fcd5188cc99b02dbca9



Now we have to spend that money using 2 of 3 users whose public keys were put in the transaction above. We can do this by running Q2P2.py file as it shown below:

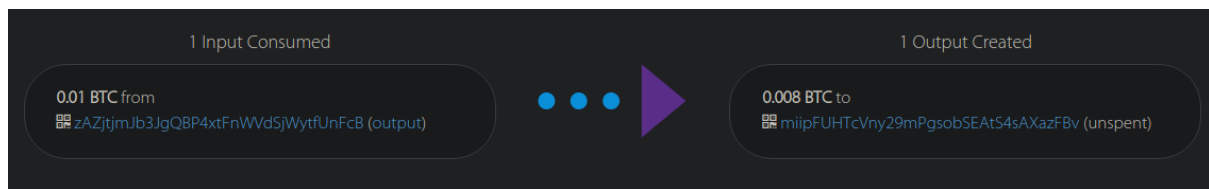
```

* alireza@alireza:~/media/alireza/C08E08E0E0E0F02/UT/Lessons/Term6/Crypto/Projects/PL1 python3 Q2P2.py
miipFUHTcVny29mPgSobSEAtS4sAXazFBv
849eef4466399d49c7493f4f72b20c2bc46144ffbc95f6b56ab69aa38a3b3aed31bfe4d7f0845d73ff96bd1486c9d277566515e1dda9dc12630d2c0393c1207
f1141aac13809f4321dc22e4baa5d173207f6cfb3a3b20f158739cb135b2
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "23b1a612d5ab9a4c4f229aee0c4dd559f9bd92345b330d049a94dae12d241032",
    "addresses": [
      "miipFUHTcVny29mPgSobSEAtS4sAXazFBv",
      "zAZjtjmJb3jgQB4xtFnWvD5jWytUnFCB"
    ],
    "total": 800000,
    "fees": 280000,
    "size": 231,
    "vsize": 231,
    "preference": "high",
    "relayed_by": "91.98.59.121",
    "received_by": "2023-05-19T14:34:25.641390981Z",
    "ver": 1,
    "double_spend": false,
    "vin_sz": 1,
    "vout_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash": "4ad648f8c62f43495af8bd713f689eb1845ed77e6b23fcd518cc99b62bdea9",
        "script": "00483045022100b67ddacf5402f51b5abf354ea49556cb3d65267f762b00f7b8c70be7f35c129a02201bc3e921d633a6253b33dab7c48f02848ef6b1ba46526920695114e955700601473044022044e879b0064323a9982786520b40aae1a596140bedd
fcb5fd2827aaad1d61e9020147beef5a237ecf6cf671de4db52f29956179949f4b391d2894c1438d28e16fb01",
        "output_value": 1000000,
        "sequence": 4291987259,
        "addresses": [
          "zAZjtjmJb3jgQB4xtFnWvD5jWytUnFCB"
        ],
        "script_type": "pay-to-multi-pubkey-hash",
        "age": 2434113
      }
    ],
    "outputs": [
      {
        "value": 800000,
        "script": "76a9142326a756092cbb75f42022417207372d1e390f6e88ac",
        "addresses": [
          "miipFUHTcVny29mPgSobSEAtS4sAXazFBv"
        ],
        "script_type": "pay-to-pubkey-hash"
      }
    ]
  }
}

```

[Here](#) is the link of the transaction above, As it shown below:

Transaction hash: 23b1a612d5ab9a4c4f229aee0c4dd559f9bd92345b330d049a94dae12d241032



Q1-P3)

Private key: 92h8F'za f ymGV8JZgYtTwNmeK2QNVXvKEms7yVtDUoyRtHpG2YN

Prime number 1: 89

Prime number 2: 13

The locking script to achieve what the question wants is as follows:

```

def customized_locking_script(sum_nums:bytes, sub_nums:bytes):
    return [OP_2DUP, OP_ADD, OP_HASH160, Hash160(sum_nums), OP_EQUALVERIFY, OP_SUB, OP_HASH160, Hash160(sub_nums), OP_EQUAL]
# return [OP_2DUP, OP_ADD, sum_nums, OP_EQUALVERIFY, OP_SUB, sub_nums, OP_EQUAL]

def customized_unlocking_script(prime_num1:bytes, prime_num2:bytes):
    return [prime_num1, prime_num2]

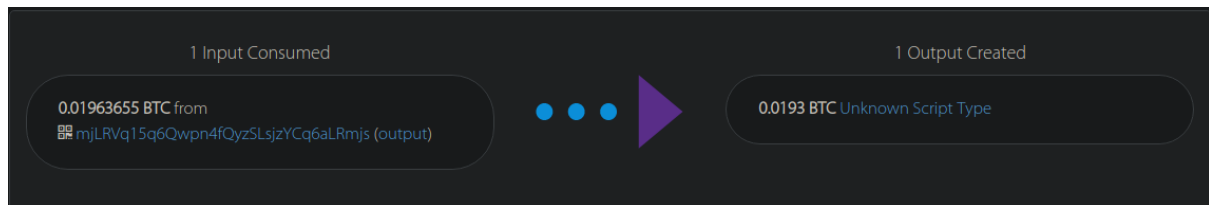
```

We can run the questioned script by running the file Q3P1.py, and the output is as follows:

```
alireza@alireza:/media/alireza/C08EE86EBE85F02/UT/Lessons/Term6/Crypto/Projects/P1$ python3 Q3P1.py
mjLRVq15q6Qwpn4fQyzSLsjzYCq6aLRmjs
04a9e48d490803b7665174a3eaae1c71ab2e091ec86ee89321bda62d8cd7e02ec5eb2c1cd619c055026bce3ed21acfb7dbe5dfe6eb6096ff58b944ee22a09f947a
91cb1aacf38a9f4321dccc22e402292d1732b97f0cfcd3a3b50ff58739cb013db2
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "b5029f1b5b52233bccf357ce69a66ed15ea0a64e296e8a1eb2a93a74e289266c",
    "addresses": [
      "mjLRVq15q6Qwpn4fQyzSLsjzYCq6aLRmjs"
    ],
    "total": 1930000,
    "fees": 33655,
    "size": 247,
    "vsize": 247,
    "preference": "high",
    "relayed_by": "91.98.219.95",
    "received": "2023-05-26T17:38:35.064521415Z",
    "ver": 1,
    "double_spend": false,
    "vin_sz": 1,
    "vout_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash": "66c465d841577e7a84f56c5f79e38a24465088e7308582af6bb23ba6e57f123",
        "output_index": 1,
        "script": "47384402208502e63affra443bcba10c24426d719d256cb77076990fae90551776efc20e8220463c1a5d6f39e6eb2821410d3a29c0b4a2baab840165faa950732a1cce8c4f60141049e480490803b7665174a3eaae1c71ab2e091ec86ee89321bda62d8cd7e02ec5eb2c1cd619c055026bce3ed21acfb7dbe5dfe6eb6096ff58b944ee22a09f947a",
        "output_value": 1963655,
        "sequence": 4294967295,
        "addresses": [
          "mjLRVq15q6Qwpn4fQyzSLsjzYCq6aLRmjs"
        ],
        "script_type": "pay-to-pubkey-hash",
        "age": 2435578
      }
    ],
    "outputs": [
      {
        "value": 1930000,
        "script": "6e93a9149b4a30911c0bcc2921cd153a2956da397b5793a48094a914c936b4fc84f72b040357e8063b0955d996eb79c4f87",
        "addresses": null,
        "script_type": "unknown"
      }
    ]
  }
}
```

[Here](#) is the link of the transaction Above, As it shown below:

Transaction hash: b5029f1b5b52233bccf357ce69a66ed15ea0a64e296e8a1eb2a93a74e289266c

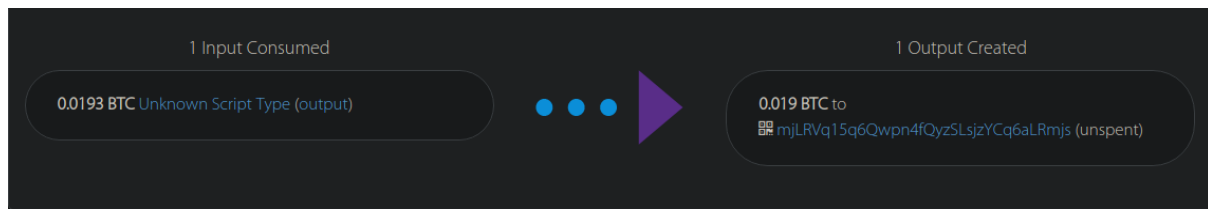


Now we have to spend this transaction by using those 2 prime numbers which was defined before, By running Q3P2.py file we have this transaction with hash:

d1116083573093ec2acbe7566b4be90529e6843b7b7b4cf3cf1bd06db5551976

```
alireza@alireza:/media/alireza/C08EE86EBE85F02/UT/Lessons/Term6/Crypto/Projects/P1$ python3 Q3P2.py
mjLRVq15q6Qwpn4fQyzSLsjzYCq6aLRmjs
04a9e48d490803b7665174a3eaae1c71ab2e091ec86ee89321bda62d8cd7e02ec5eb2c1cd619c055026bce3ed21acfb7dbe5dfe6eb6096ff58b944ee22a09f947a
91cb1aacf38a9f4321dccc22e402292d1732b97f0cfcd3a3b50ff58739cb013db2
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "d1116083573093ec2acbe7566b4be90529e6843b7b7b4cf3cf1bd06db5551976",
    "addresses": [
      "mjLRVq15q6Qwpn4fQyzSLsjzYCq6aLRmjs"
    ],
    "total": 1900000,
    "fees": 30000,
    "size": 89,
    "vsize": 89,
    "preference": "high",
    "relayed_by": "91.98.219.85",
    "received": "2023-05-26T18:01:05.366552818Z",
    "ver": 1,
    "double_spend": false,
    "vin_sz": 1,
    "vout_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash": "b5029f1b5b52233bccf357ce69a66ed15ea0a64e296e8a1eb2a93a74e289266c",
        "output_index": 0,
        "script": "0159010d",
        "output_value": 1930000,
        "sequence": 4294967295,
        "script_type": "unknown",
        "age": 2435579
      }
    ],
    "outputs": [
      {
        "value": 1900000,
        "script": "76a91429e2a2aef4bf8b9e59fac7fa50af4f34125519088ac",
        "addresses": [
          "mjLRVq15q6Qwpn4fQyzSLsjzYCq6aLRmjs"
        ],
        "script_type": "pay-to-pubkey-hash"
      }
    ]
  }
}
```

Here is the schematic of the corresponding transaction:



Section 3) Mining a block

Step1) Find my block information:

My block num: 9334

My block hash:

00000000261b4765edf334510ef4e167cbaf58406af6281891b4d57595c10fc9

Step2) create coinbase transaction:

Transaction_input = 32 bytes 0

Transaction_input_index = 0xFFFFFFFF

Coinbase_Data = 810199334AlirezaArbabi (ASCII to hex)

Output script = Our address to receive block reward

Step3) Calculate Merkle Root. As we have only 1 transaction, the merkle root will be the SHA256 hash of the coinbase transaction.

Step4) Mine desired block. Now we have to put prev_block_hash, merkle_root, and nonce into the block header. We have to find the nonce in a way that the hash of the block header becomes lower than $2^{(256 - 16)}$.

By running the `BlockMiner.py` code, we have:

[illegible]