

Question 1

Part 1 (generate bitcoin private key)

For this part, we generate a random 256 bit(32 byte) number for private key.

```
random_number = random.getrandbits(256)
private_key = random_number.to_bytes(32, byteorder="big")
```

To generate the Wallet Import Format (WIF) of a private key, the following steps can be followed:

1. Start with the private key in its hexadecimal format.
2. Add the test network prefix (0xEF) to the beginning of the private key (0x80 for main network).
3. Perform a double hash operation on the extended private key using a cryptographic hash function, such as SHA-256.
4. Take the first 4 bytes of the resulting hash and append them to the extended private key.
5. Encode the extended private key, including the appended checksum, into a base58 encoding algorithm.
6. The resulting string is the WIF representation of the private key.¹

By following these steps, the private key can be converted into a WIF format that is commonly used for importing private keys into cryptocurrency wallets.

To generate a public key based on a given private key, the following steps can be followed:

1. Utilize the ECDSA (Elliptic Curve Digital Signature Algorithm) to generate a public key corresponding to the provided private key.
2. Add the public key prefix (0x04) to the beginning of the generated public key. This prefix distinguishes uncompressed public keys.
3. Apply a hash function, such as SHA-256, followed by RIPEMD160, to the public key. This process results in a hashed value.
4. Prepend the test network prefix (0x6f) to the hashed value obtained in the previous step. This prefix is used to identify the network or purpose for which the public key is intended.
5. Calculate the double hash of the extended key using SHA-256, obtaining a new hash result.
6. Append the first 4 bytes (checksum) of the double hash to the end of the extended key.
7. The resulting string is the public key derived from the given private key.

By following these steps, the public key can be obtained from a private key.

¹ For more info visit https://en.bitcoin.it/wiki/Wallet_import_format

```
WIF:
9289tv65ADDci2VrHf71YR6qarf76bvrueN1FSo98PwaRpkd8Q
Public key:
mzR5eMMtwgcNt7SogR61Jd1Jk8vj6qKnVC
```

Figure 1 test result

Part 2 (generate bitcoin address)

In the following section, we focus on generating a vanity address based on a user-defined input string. A vanity address is an address that contains a specific subset of characters, starting from the second character and ending at the $[2+n]$ th character of the Bitcoin public key address. To determine the number of attempts required to achieve the desired vanity address, we implement a code that iteratively searches for a matching address.

Please note that when the input string is longer than three characters, the calculation process can be time-consuming due to the increased complexity of finding a matching subset.

```
PS E:\UT\Semester 6\Crypto\HW\CA> python -u "e:\UT\Semester 6\Crypto\HW\CA\q1_p2.py"
mir
10,000 addresses generated so far!
20,000 addresses generated so far!
30,000 addresses generated so far!
40,000 addresses generated so far!
50,000 addresses generated so far!
60,000 addresses generated so far!
70,000 addresses generated so far!
80,000 addresses generated so far!
90,000 addresses generated so far!
100,000 addresses generated so far!
110,000 addresses generated so far!
120,000 addresses generated so far!
130,000 addresses generated so far!
140,000 addresses generated so far!
150,000 addresses generated so far!
160,000 addresses generated so far!
170,000 addresses generated so far!
180,000 addresses generated so far!
190,000 addresses generated so far!
200,000 addresses generated so far!
210,000 addresses generated so far!
220,000 addresses generated so far!
Found vanity address after 229607 attempts!
Vanity Address: mmir5fG6s4nDcVmiKveP92nkA7BD4sNLZK
Vanity Private Key (WIF): p3GFWshBtu4iSFwM6hVN5WcLSSD8sEpHwBhPpAnJwEgpCAEjaGd
```

Figure 2. Vanity address generation result starting with "mir"

Found vanity address after 229607 attempts!

Vanity Address: m**mir**5fG6s4nDcVmiKveP92nkA7BD4sNLZK

Vanity Private Key (WIF):

93GFWshBtu4iSFwM6hVN5WcLSSD8sEpHwBhPpAnJwEgpCAEjaGd

Question 2

To acquire a certain amount of Bitcoin on the test network, we can utilize a faucet website².

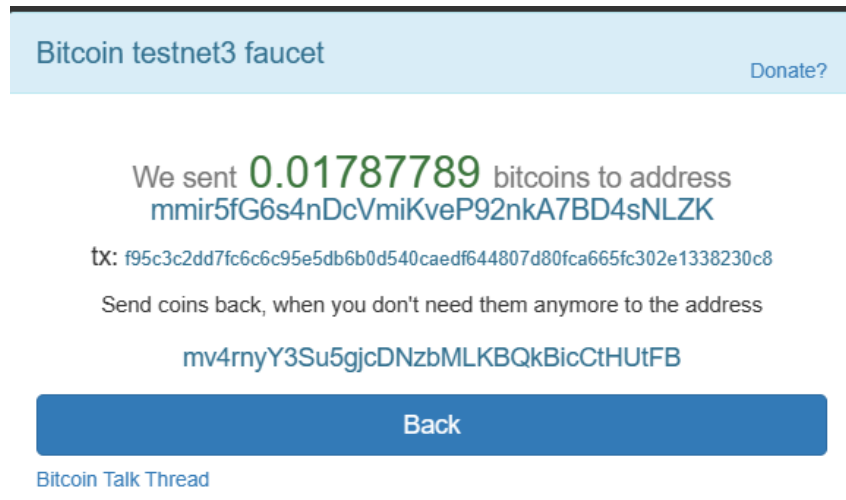


Figure 3. Bitcoin Received on the Provided Address

```
txid_to_spend =  
( 'f95c3c2dd7fc6c95e5db6b0d540caedf644807d80fca665fc302e1338230c8' )
```

After initiating a transaction, it is essential to wait for a certain number of blocks to be added to the blockchain to ensure the transaction is securely recorded and considered final. Generally, a common practice is to wait for at least six blocks, which provides a sufficient level of confirmation for the transaction.

We initiated a transaction with one input and two outputs. Figure 4 illustrates the structure of this transaction. The purpose of the two outputs is as follows:

1. The first output, also shown in Figure 4, is spendable and can be used to transfer the specified Bitcoin amount to the provided address. This output allows the Bitcoin to be sent to the designated recipient.
2. The second output, as depicted in Figure 4, is not spendable and is utilized to return the Bitcoin amount to the faucet address. This ensures that any remaining funds or change from the transaction are returned to the original source.

² <https://coinfaucet.eu/en/btc-testnet>

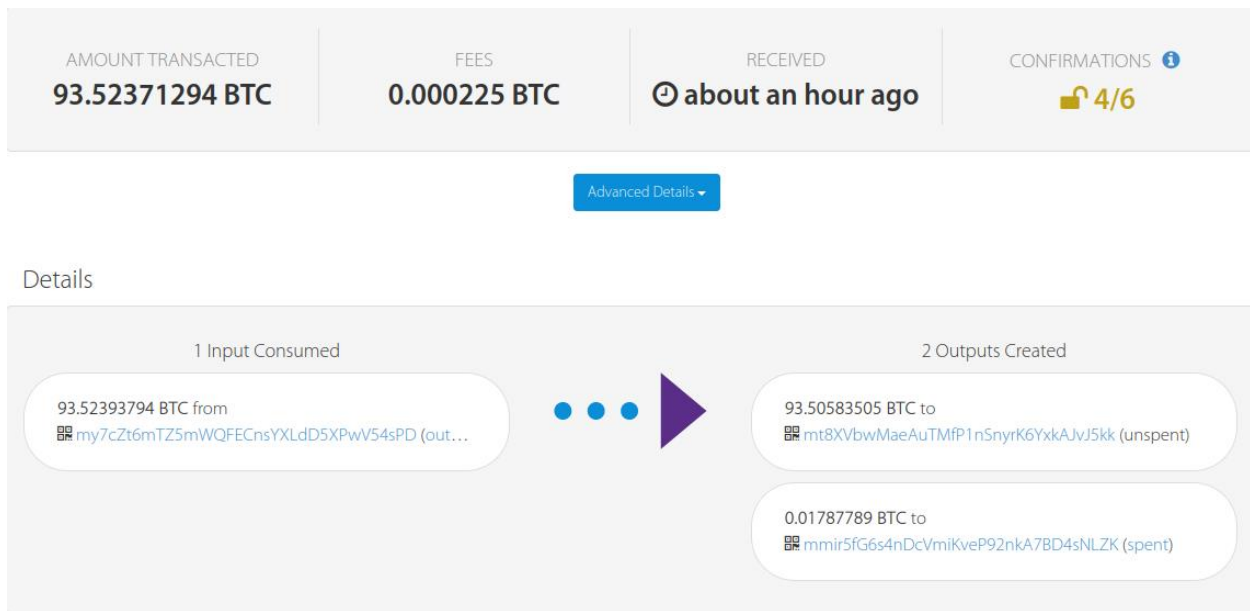


Figure 4

To complete the transaction.py file, we utilized a specific [website](#) to study our transaction and analyze the script used within it. This website provided valuable insights and information regarding the transaction structure and associated scripts as shown in figure 5.

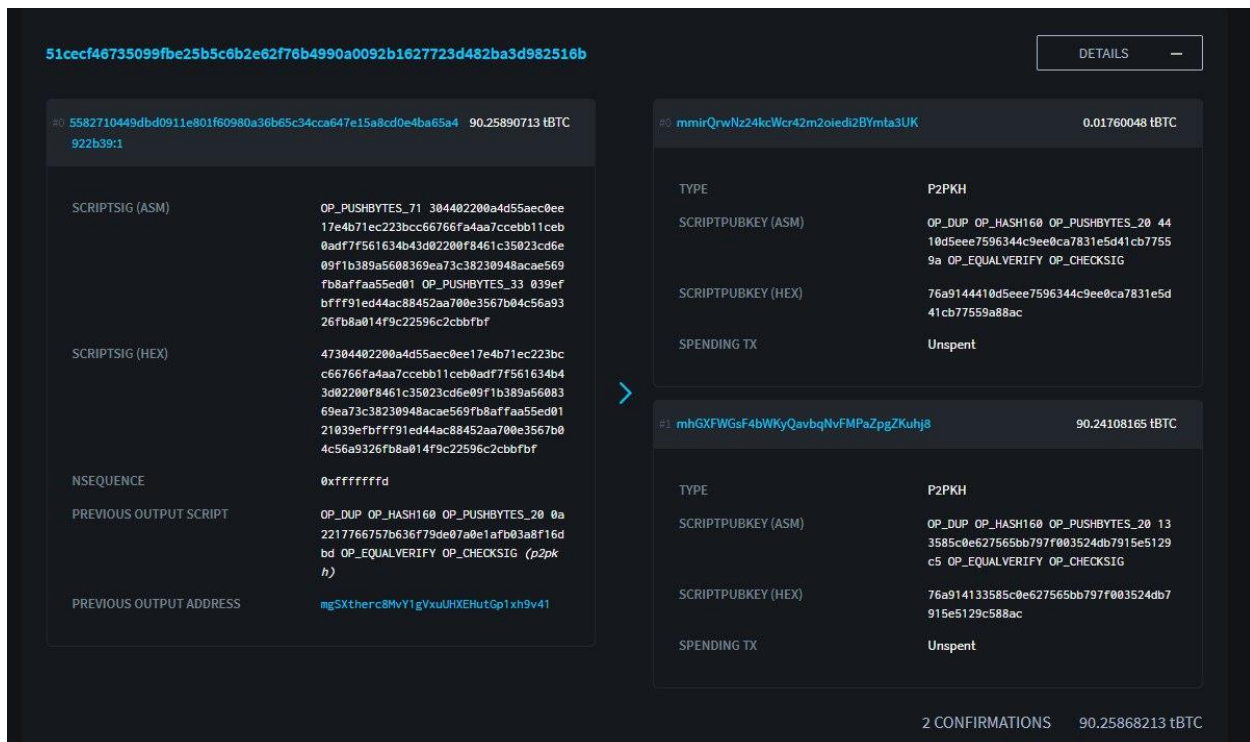
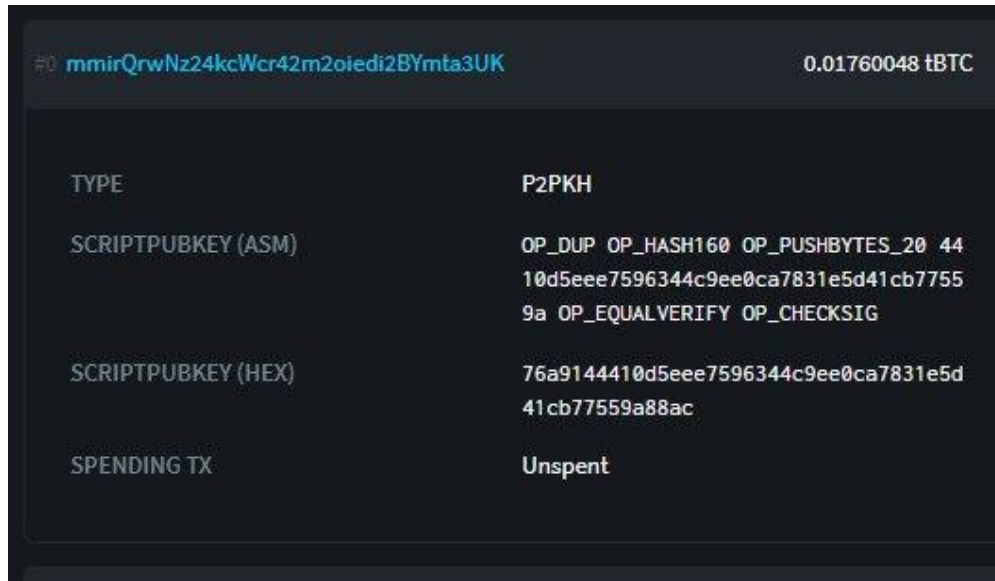


Figure 5

Part 1

Utilizing the provided script, we implemented the necessary logic using script shown in figure 6 within the transaction.py file. This logic allows us to spend the unspent portion of the faucet transaction and generate a new transaction with one input and two outputs.

The first output is designated as spendable, allowing the specified Bitcoin amount to be sent to the desired address. The second output is configured as not spendable, serving the purpose of returning any remaining funds or change back to the faucet address.



The screenshot shows a Bitcoin transaction output. At the top, it displays the output index '#0', the address 'mmirQrwNz24kcWcr42m2oiedi2BYmta3UK', and the amount '0.01760048 tBTC'. Below this, a table lists the details of the output:

TYPE	P2PKH
SCRIPTPUBKEY (ASM)	OP_DUP OP_HASH160 OP_PUSHBYTES_20 44 10d5eee7596344c9ee0ca7831e5d41cb7755 9a OP_EQUALVERIFY OP_CHECKSIG
SCRIPTPUBKEY (HEX)	76a9144410d5eee7596344c9ee0ca7831e5d41cb77559a88ac
SPENDING TX	Unspent

Figure 6

Upon executing the transaction.py code, the output is obtained, which is illustrated in Figure 7 of the report.

```
mohammad@kali:~/07/crypt$ python3 q2.pl.py
address base58 = mmir5fG6s4nDcVmKveP92nkA7BD4sNLZK
public key = 04a83991a39d5eedec95c592a878cccf115568a9dc4e6baf872a3999ea682e82d4247ad87ccea1d3e51e8458ce52d25a9558b9faf851241a7839b144478d9a
private key = dd928c268d211bda0149bb04cb9488613e1515c98c4cf8c5c867c31465b69544
response status code = 201
response reason = Created
Report:
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "b9baa009551b44a091d4445be9bdf737afce282fa7c4620ffca5bcfb4296974",
    "addresses": [
      "mmir5fG6s4nDcVmKveP92nkA7BD4sNLZK"
    ],
    "total": 177788,
    "fees": 10001,
    "size": 289,
    "vsize": 289,
    "preference": "low",
    "relayed_by": "188.118.96.41",
    "received": "2023-05-28T13:58:02.970264091Z",
    "ver": 1,
    "double_spend": false,
    "win_sz": 1,
    "vout_sz": 2,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash": "f95c3c2dd7fc6c95e5db6b8d548caedf644887d88fca665fc302e1338238c8",
        "output_index": 1,
        "script": "423844822a65af7bd091d67f1cbb56aebdcdf7eced22c26c3eac0b2d64c3af9498dd3ac68228138b6cc825a5492734d2329949a4bad5f753e16d5a1f8e25d47bf0227e32c76814184e83991a39d5eedec95c592a878cccf115568a9dc4e6baf872a3999ea682e82d4247ad87ccea1d3e51e8458ce52d25a9558b9faf851241a7839b144478d9a",
        "output_value": 177789,
        "sequence": 4294967295,
        "addresses": [
          "mmir5fG6s4nDcVmKveP92nkA7BD4sNLZK"
        ],
        "script_type": "pay-to-pubkey-hash",
        "age": 2435762
      }
    ],
    "outputs": [
      {
        "value": 1788888,
        "script": "51",
        "addresses": null,
        "script_type": "unknown"
      },
      {
        "value": 77788,
        "script": "88",
        "addresses": null,
        "script_type": "unknown"
      }
    ]
  }
}
```

Figure 7

The transaction.py code execution yielded the following output:

- "addresses": ["mmir5fG6s4nDcVmKveP92nkA7BD4sNLZK"]
- "preference": "low"
- "total": 1500000
- "received": "2023-05-28T13:58:02.970264091Z"
- "double_spend": false
- "script_type": "pay-to-pubkey-hash"
- Txid = b9baa009551b44a091d4445be9bdf737afce282fa7c4620ffca5bcfb4296974

By monitoring our transaction on the Bitcoin blockchain, it is evident that the funds have been successfully expended, as illustrated in Figure 8 on the faucet website.

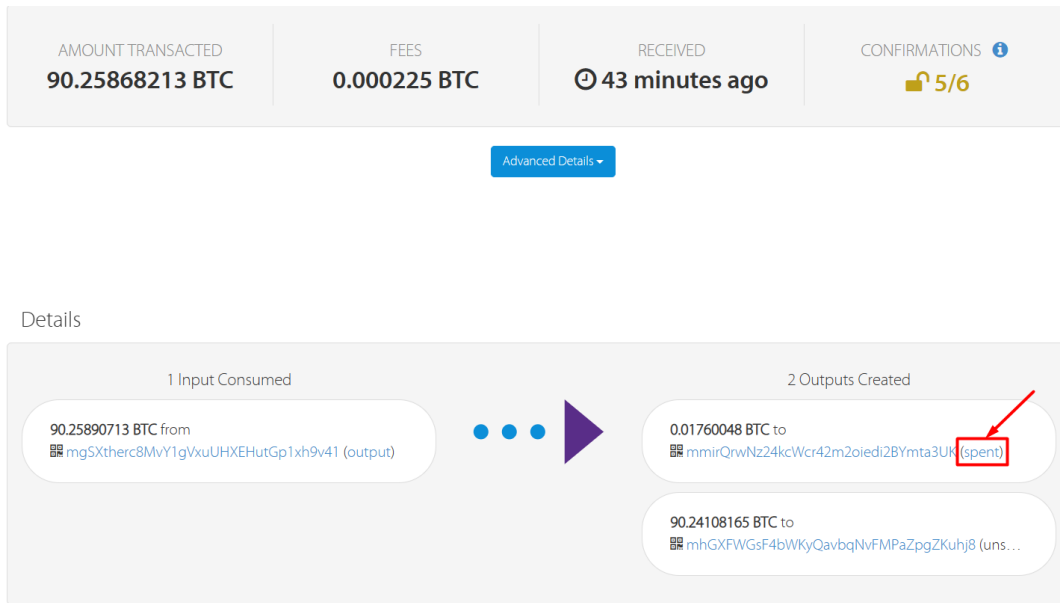


Figure 8

The transaction information is as shown in figure 9.

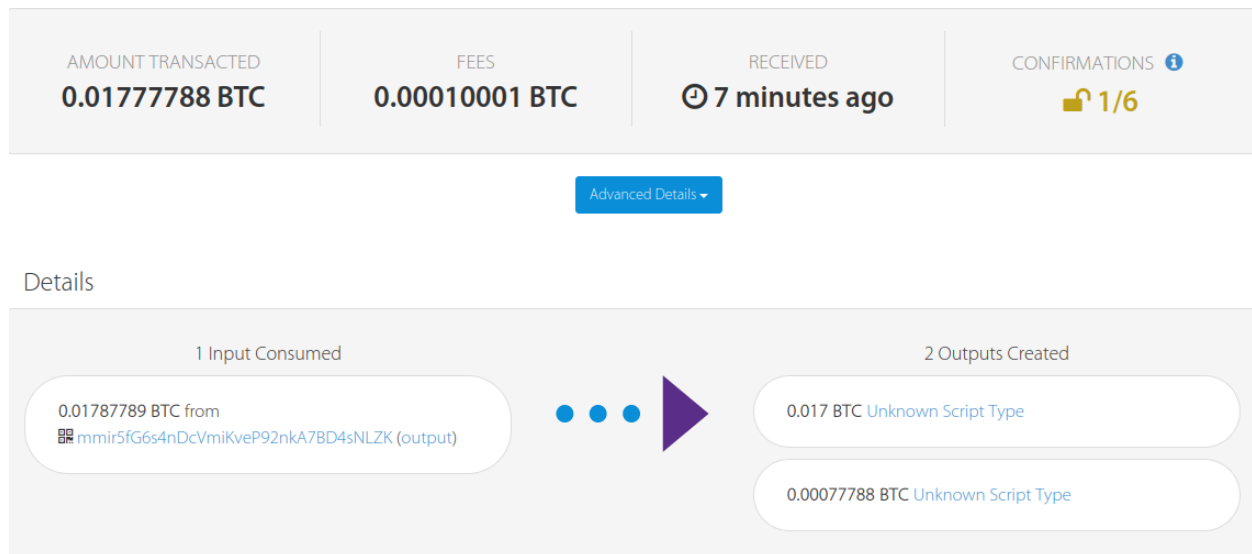


Figure 9

In another transaction, we spend the spendable output of this transaction and return it to our original address as P2PKH output.(q2_p1_2.py file)

The address is : b9baa009551b44a091d4445be9bdf737afce282fa7c4620ffca5bcfbb4296974

```
mohta3b@mohta3b:~/UT/Crypto$ python3 q2_p1.2.py
address base58 = mmir5fG6s4nDcVmiKveP92nkA7BD4sNLZK
public key = 04e83991a39d5eedec95c592a8787cccf115568a9dc4e6baf872a3999ea602e82d4247ad07ccee1a1d3e51e0450ce52d25a9558b9fa0f051241a7839b144478d9a
private key = dd020c268d211bda0149bb04cb9488613e1515c50c4cf8c5c007c31465b69544
response status code = 201
response reason = Created
Report:
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "aea8ad8d04415c59624fa300dfd7a7c0d7a4a99409945e72d58ec8840379dfbd",
    "addresses": [
      "mmir5fG6s4nDcVmiKveP92nkA7BD4sNLZK"
    ],
    "total": 1600000,
    "fees": 100000,
    "size": 85,
    "vsize": 85,
    "preference": "high",
    "relayed_by": "188.118.96.41",
    "received": "2023-05-28T14:15:14.846680446Z",
    "ver": 1,
    "double_spend": false,
    "vin_sz": 1,
    "vout_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash": "b9baa009551b44a091d4445be9bdf737afce282fa7c4620ffca5bcfb4296974",
        "output_index": 0,
        "output_value": 1700000,
        "sequence": 4294967295,
        "script_type": "unknown",
        "age": 2435766
      }
    ],
    "outputs": [
      {
        "value": 1600000,
        "script": "76a91444108f2fd841adc2f4cb4746cd4e97c1e89d912488ac",
        "addresses": [
          "mmir5fG6s4nDcVmiKveP92nkA7BD4sNLZK"
        ],
        "script_type": "pay-to-pubkey-hash"
      }
    ]
  }
}
```

Figure 10 : test code result

The transaction.py code execution yielded the following output:

- "addresses": ["mmir5fG6s4nDcVmiKveP92nkA7BD4sNLZK"]
- "preference": "high"
- "received": "2023-05-28T14:15:14.846680446Z"
- "double_spend": false
- "script_type": "unknown"
- Txid = aea8ad8d04415c59624fa300dfd7a7c0d7a4a99409945e72d58ec8840379dfbd

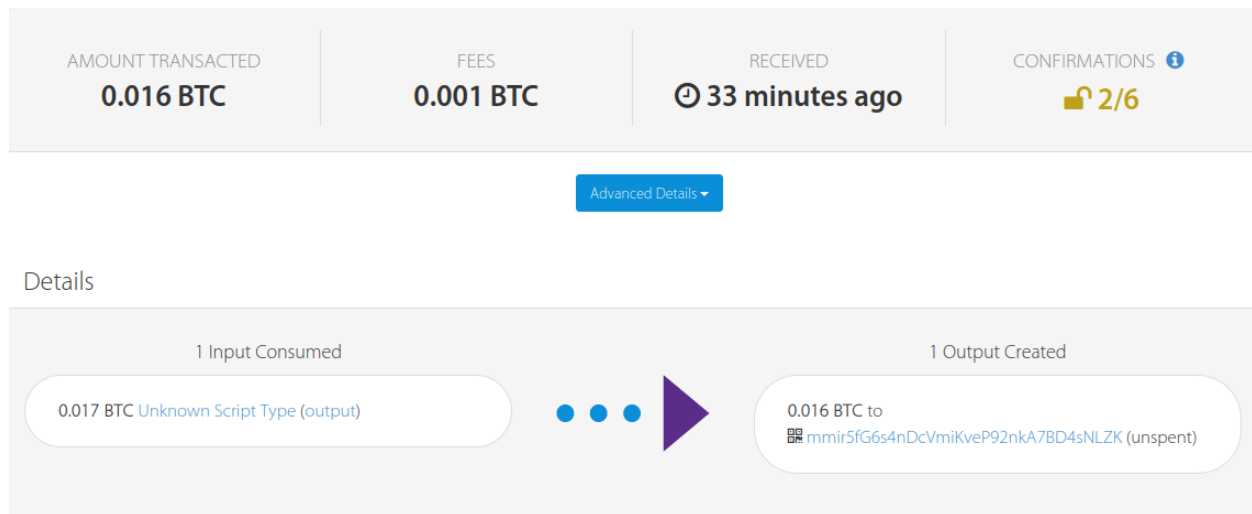


Figure 11: test result

Part 2

First we generate 3 bitcoin address on test network using the codes of the last part.

```
Address 0:
WIF: 923CPJacNiH4crjnz9hTH6cB7kb38CeiURRMoL1GTUFq9v64w8c
public key: b'muUpXeFJEex17XMo6Pvh5CH6AeWPuaMsRj'

Address 1:
WIF: 93JBuQXNHqZpvAZ7ySzjQRENUfpP8FVge9EzmRmxmUYVKiUDAwj
public key: b'mmykEvAKJKdB9nGYMNkZW42cTz3La8zQA1'

Address 2:
WIF: 93Rpjdp6hmRJRHqwM9kUZy8Mr5L8y6eoSNmSS6u4xMstQVcAZQD
public key: b'moy21KeffeHKUKbwNWVMdJjDzp5X11HAN6'
```

Figure 12

Address 1:

WIF: 923CPJacNiH4crjnz9hTH6cB7kb38CeiURRMoL1GTUFq9v64w8c
address: muUpXeFJEex17XMo6Pvh5CH6AeWPuaMsRj

Address 2:

WIF: 93JBuQXNHqZpvAZ7ySzjQRENUfpP8FVge9EzmRmxmUYVKiUDAwj
address: mmykEvAKJKdB9nGYMNkZW42cTz3La8zQA1

Address 3:

WIF: 93Rpjdp6hmRJRHqwM9kUZy8Mr5L8y6eoSNmSS6u4xMstQVcAZQD
address: moy21KeffeHKUKbwNWVMdJjDzp5X11HAN6

```

address bases5 = mnlr5f66s4dcvnlkveP92nka7BD45NLZK
public key = 84a3991a13d5eeec9c5c592a877ccf1155b8a9dc1d8107723999a602a024247a087ccea1d3e1ae450e52d25955089f8a1851241a7839b1444789a
private key = d082a260211b0ea1498a94c9489313a5515c8c4fbc85e007c31465869a944
txout scriptPubkey = [OP_2, b'muqWx3Eox170u6pH5SCH6ANpuHsRj', b'moyKEVAC3Kd8nG5YMNxW42C7z3La8zQA1', b'moy21KxeffHUKkUbmVWVmj3DzpsX11Ha66', OP_3, OP_CHECKMULTISIG]
response status code = 201
response reason = created
Report:
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "338149f9edc415f3a5ae7eab736d7856e72f49a95116ff58066a5f6d471412f",
    "addresses": [
      "mnlr5f66s4dcvnlkveP92nka7BD45NLZK",
      "xKjE3LntBxoH43KewB962hjKPRxqjpb"
    ],
    "total": 1550000,
    "fees": 50000,
    "size": 386,
    "vsize": 386,
    "preference": "high",
    "relayed_by": "213.142.159.95",
    "received": "2023-05-28T16:10:06.287458489Z",
    "ver": 1,
    "double_spend": false,
    "vin_sz": 1,
    "vout_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash": "eeabadd084415c59624fa380d7d7c8d7a04a89409945e7d58ec8848379dfbd",
        "output_index": 0,
        "script": "47384482856d5593d9dd16251f786ef182c3cf35a2244b66af568745fe78bcb1a6d6820077795cb5c1713842c3d937e1ccf86753701411715086c33f5f11b8d3609ce14104e83991a395eedec95c592a877ccf115568a9dc4e6ba872a3999ea0e282d4247a07ccea1d3e1ae450e52d25955089f8a1851241a7839b1444789a",
        "output_value": 1600000,
        "sequence": 4249067295,
        "addresses": [
          "mnlr5f66s4dcvnlkveP92nka7BD45NLZK"
        ],
        "script_type": "pay-to-pubkey-hash",
        "age": 2435768
      }
    ],
    "outputs": [
      {
        "value": 1550000,
        "script": "3226d7555785805464a495703137584d6f36587668354383641655785614d73526a226d6d796b4576414ba4b6442396e475946deb5a7343263547a334c61387a5141132126d6f793214b05666665484b554862774e57564d6446a447a78355831148416e3653aa",
        "addresses": [
          "xKjE3LntBxoH43KewB962hjKPRxqjpb"
        ],
        "script_type": "pay-to-multiple-pubkey-hash"
      }
    ]
  }
}

```

The transaction.py code execution yielded the following output:

- "addresses": ["mmir5fG6s4nDcVmiKveP92nkA7BD4sNLZK",
"zKjE3LnTBxoMi4JKeW8Mo962hjKPRXq1pB"]
- "preference": "high"
- "received": "2023-05-28T16:10:06.287458409Z"
- "double_spend": false
- "script_type": "unknown"
- Txid = 5301497edc415f3a5ae7e4b736de7056e72f4d9a95116ff58068a5f0d471412f



Figure 14 test result

Then we send back this money to it's original address in another transaction.

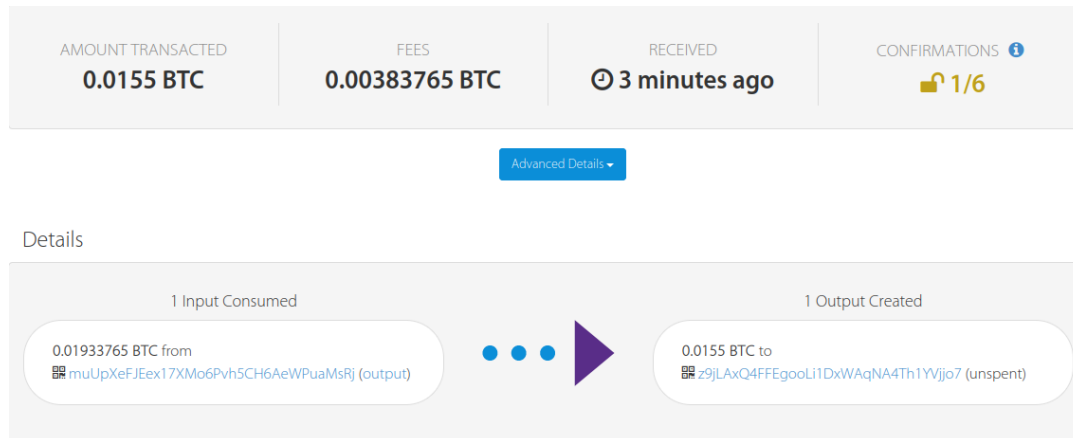


Figure 15

```

mohtab@mohtab3:~/UT/Crypto$ python3 q2_p2_2.py
address base58 = muUpXeFJEex17XMo6Pvh5CH6AeWPuaMsRj
public key = 845e6dd5157375a2a155d8e8c77d6d421e6f528c8e9fc8c795643615d118b7a6f72eba7e38e464f73dbcb598e7d5fb491bc2e13442db8bac95d2a8f89cc47bb
private key = 3badd770361141ff87bc8855c5aa9e9b270599713c3f9c6bb16c7c49dfc76d62
response status code = 201
response reason = Created
Report:
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "5caa3fd775d9c0286a793667f4a905328cb7ba199f9e6e742a8032f3ad609e91",
    "addresses": [
      "z9jLaxQ4FFEgooli1DxWAqNA4Th1YVjjo7",
      "muUpXeFJEex17XMo6Pvh5CH6AeWPuaMsRj"
    ]
  },
  "total": 1500000,
  "fees": 58880,
  "size": 231,
  "vsize": 231,
  "preference": "high",
  "relayed_by": "213.142.159.95",
  "received": "2023-05-28T17:41:18.74480761Z",
  "ver": 1,
  "double_spend": false,
  "in_sz": 1,
  "vout_sz": 1,
  "confirmations": 0,
  "inputs": [
    {
      "prev_hash": "9c795b08275d90870a9c59c5dbf3fc98151e428dce8fc8d728de1cda6e94853",
      "output_index": 0,
      "script": "8647384492280a353df754192117f3f5d245315874396facd18e4c5e18ed8f53908c934c42bf822882326405a18e9ca6ceff7b1abd7264222d6f83c6a243d5162c172d161bc7742381483845822188e4e553b628f861f04cfc81ab87a9d2dbcbae1a16f13bbe893ad6f151b5959202958bedad4810eaf88e4e81aff922b3876d18a6f75afe4198a8391d8e541a869181",
      "output_value": 1500000,
      "sequence": 4294967295,
      "addresses": [
        "z9jLaxQ4FFEgooli1DxWAqNA4Th1YVjjo7"
      ],
      "script_type": "pay-to-multi-pubkey-hash",
      "age": 2435779
    }
  ],
  "outputs": [
    {
      "value": 1500000,
      "script": "76e914992a7251f022339dd2853241ee05f2884d67c38888ac",
      "addresses": [
        "muUpXeFJEex17XMo6Pvh5CH6AeWPuaMsRj"
      ],
      "script_type": "pay-to-pubkey-hash"
    }
  ]
}

```

Figure 16 test result

The transaction.py code execution yielded the following output:

- "addresses": ["z9jLaxQ4FFEgooli1DxWAqNA4Th1YVjjo7",
"muUpXeFJEex17XMo6Pvh5CH6AeWPuaMsRj"]
- "preference": "high"
- "received": "2023-05-28T17:41:18.74480761Z"
- "double_spend": false
- "script_type": "pay-to-multi-pubkey-hash"
- Txid = 5caa3fd775d9c0286a793667f4a905328cb7ba199f9e6e742a8032f3ad609e91

Part 3-1

We anticipate that the first two numbers will be placed in the stack for spending. In this step, the first DUP2_OP operation is performed to create copies of both numbers. The subtraction operation (SUB_OP) is then applied, followed by the hashing of the resulting number. The hashed number is then concealed. The two hashed numbers are compared, and if they match, they are removed from the stack (EQUALVERIFY_OP). Numbers we choose are 1009 and 1861. Test result is shown in figure 17.

```
address_base58 = muUpXeFJEex17XMo6PvH5CH6AeWPuaMsRj
public key = 845eeddb515f375a2eaf58de8cf7dadf42febf528c89efc8c795643615d110b7a6f72e6a7e38e464f73dbce590e7d5fb491bc1e13442db8ac95d2aef89ccd7bb
private key = 3badd778361141ff87bc0855c5a9e9b270589713c3f9c6bb16c7c48dfc76d62
response status code = 201
response reason = Created
Report:
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "7246d433b0730ea74a4b8ff985a28cb2232dd9a80ddffbf14d61ca242b66f26c",
    "addresses": [
      "muUpXeFJEex17XMo6PvH5CH6AeWPuaMsRj"
    ],
    "total": 1450000,
    "fees": 50000,
    "size": 247,
    "vsize": 247,
    "preference": "high",
    "relayed_by": "213.142.159.95",
    "received": "2023-05-28T17:55:43.73412166Z",
    "ver": 1,
    "double_spend": false,
    "win_sz": 1,
    "out_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash": "5caa3fd775d9c8286a793667fa985328cb7ba199f9e6e742a8032f3ad689e91",
        "output_index": 0,
        "script": "4738448228506898d82745a092d9bb83d7383b3ac1695dd3e47eedbd9122e9ab5133e5f82283d3289fa8c771b216232bcfab89bf43742c8c2d5edc484f859d743e6223b6aba0141645eeddb515f375a2eaf58de8cf7dadf42febf528c89efc8c795643615d110b7a6f72e6a7e38e464f73dbce590e7d5fb491bc1e13442db8ac95d2aef89ccd7bb",
        "output_value": 1500000,
        "sequence": 4294967295,
        "addresses": [
          "muUpXeFJEex17XMo6PvH5CH6AeWPuaMsRj"
        ],
        "script_type": "pay-to-pubkey-hash",
        "age": 0
      }
    ],
    "outputs": [
      {
        "value": 1450000,
        "script": "6e94a91409fa2cbe5cd3b5c8bfa552fe8c1a3bf48365e3a8893a914609fa836b209f3534f5dff1cd6b6132e4094f51187",
        "addresses": null,
        "script_type": "unknown"
      }
    ]
  }
}
```

Figure 17

The transaction.py code execution yielded the following output:

- "addresses": ["muUpXeFJEex17XMo6PvH5CH6AeWPuaMsRj"]
- "preference": "high"
- "received": "2023-05-28T17:55:43.73412166Z"
- "double_spend": false
- "script_type": "pay-to-pubkey-hash"
- Txid = 7246d433b0730ea74a4b8ff985a28cb2232dd9a80ddffbf14d61ca242b66f26c

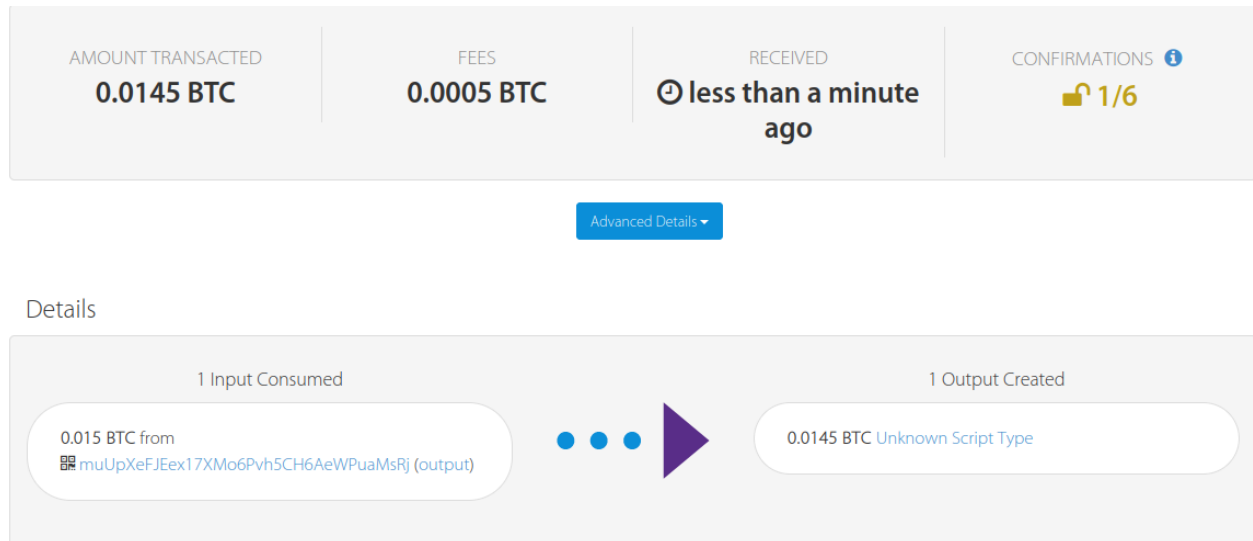


Figure 18

Part 3-2

```
mohta3b@mohta3b:~/UT/Crypto$ python3 q2_p3_2.py
address base58 = muUpXeFJEex17XMo6Pvh5CH6AeWPuaMsRj
public key = 045eeddb515f375a2eaf58de8cf7dadf42fe6f528c89efc8c795643615d110b7a6f72eba7e38e464f73dbc0590e7d5fb491bc2e13442db8bac95d2a0f89ccd7bb
private key = 3babd770361141ff87bc0855c5aa9e9b270509713c3f9c6bb16c7c40dfc76d62
response status code = 201
response reason = Created
Report:
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "fa429408784f6450bfed1167a618cd009aa7cb7da5dd23710515bfa0d1cb120d",
    "addresses": [
      "muUpXeFJEex17XMo6Pvh5CH6AeWPuaMsRj"
    ],
    "total": 1400000,
    "fees": 50000,
    "size": 91,
    "vsize": 91,
    "preference": "high",
    "relayed_by": "213.142.159.95",
    "received": "2023-05-28T18:04:20.117975185Z",
    "ver": 1,
    "double_spend": false,
    "vin_sz": 1,
    "vout_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash": "7246d433b0730ea74a4b8ff985a28cb2232dd9a80ddffbf14d61ca242b66f26c",
        "output_index": 0,
        "script": "02450702f103",
        "output_value": 1450000,
        "sequence": 4294967295,
        "script_type": "unknown",
        "age": 2435780
      }
    ],
    "outputs": [
      {
        "value": 1400000,
        "script": "76a914992a7251f922339dd2853241ee05f2884d67e38088ac",
        "addresses": [
          "muUpXeFJEex17XMo6Pvh5CH6AeWPuaMsRj"
        ],
        "script_type": "pay-to-pubkey-hash"
      }
    ]
  }
}
```

Figure 19 test result

The transaction.py code execution yielded the following output:

- "addresses": ["muUpXeFJEex17XMo6Pvh5CH6AeWPuaMsRj"]
- "preference": "high"
- "received": "2023-05-28T18:04:20.117975185Z"
- "double_spend": false
- "script_type": "pay-to-pubkey-hash"
- Txid = fa429408784f6450bfed1167a618cd009aa7cb7da5dd23710515bfa0d1cb120d

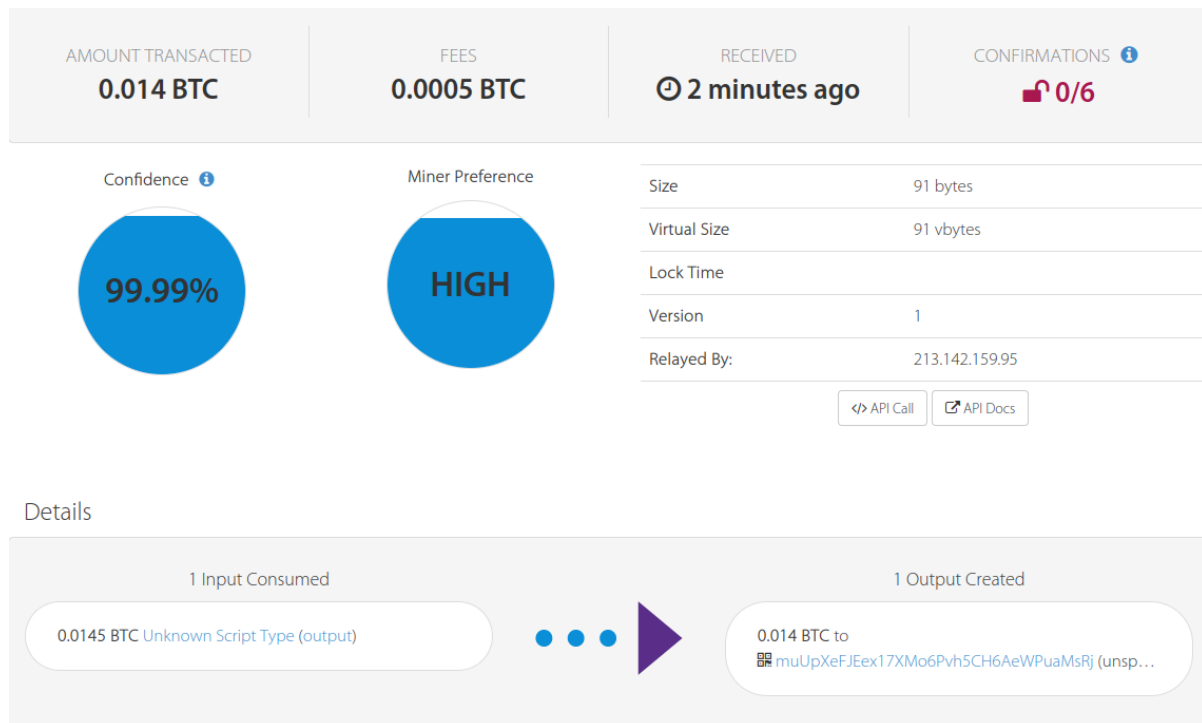


Figure 20 test result

Question 3

We utilize a specific [website](#) to identify a desired block hash for mining a block on Bitcoin's main network.

Bitcoin Block 9,511

Mined on April 02, 2009 06:53:37 • [All Blocks](#)

Unknown

Coinbase Message •

A total of 0.00 BTC (\$0.00) were sent in the block with the average transaction being 0.0000 BTC (\$0.00). Unknown earned a total reward of 50.00 BTC \$0.00. The reward consisted of a base reward of 50.00 BTC \$0.00 with an additional 0.0000 BTC (\$0.00) reward paid as fees of the 1 transactions which were included in the block.

Details



Hash	00000-4cdd9 	Depth	782,331
Capacity	00000000faac558a7a5266c3c678	Size	216
Distance	e53b53b88a619b00dd825395b8e	Version	0x1
BTC	4ca44cdd9	Merkle Root	64-fc 
Value	\$0.00	Difficulty	1.00
Value Today	\$0.00	Nonce	130,444,156
Average Value	0.0000000000 BTC	Bits	486,604,799
Median Value	50.00000000 BTC	Weight	864 WU
Input Value	0.00 BTC	Mined	50.00 BTC
Output Value	50.00 BTC	Reward	50.00000000 BTC
Transactions	1	Mined on	Apr 02, 2009, 6:53:37 AM
Witness Tx's	0	Height	9,511
Inputs	1	Confirmations	782,331
Outputs	1	Fee Range	0-0 sat/vByte
-	-	-	-

Figure 21

The structure of a Bitcoin block is as follows:

- Magic Number
- Blocksize
- Blockheader
- Transaction Count
- Transactions

The blockheader section consists of six fields:

Version - Previous Block Hash - Merkle Root Hash – Timestamp – Bits - Nonce

In this case, if we consider the value of the Bits field as 0x1f010000, we can extract the exponent as "1f" and the coefficient as "010000". By combining these values, we can derive the Target value. The Target represents a specific level of difficulty for mining a block, where the leftmost 15 bits are zero, followed by a single "1", and the remaining bits are zero. Consequently, a valid hashed block must be numerically lower than this Target value, requiring it to have 16 leading zeros.

As of now, the current Bitcoin block reward is 6.25 BTC.

In our testing process, we iterate through several nonce values until we find a nonce for which the computed hash of the block is numerically smaller than the Target value. This successful outcome indicates that the block satisfies the difficulty requirements and can be added to the blockchain.

Result of test code is shown in figure 22.

Figure 22

17