# Exercises: Private Key to Address

In this exercise you shall generate a **private key**, calculate a corresponding **public key** from it and generate a **blockchain address** for our educational blockchain network.

## Generate Private Key and Address

Write a program to **generate a private key** (256-bit number), calculate its corresponding **public key** using the **secp256k1** ECC cryptography, **compress it** in specific format (X coordinate + 0 / 1 for add / even Y coordinate) and derive a **blockchain address** from it (using RIPEMD-160), like it is described in the documentation for our educational blockchain network. Use the same format for the keys and addresses like at the sample output below.

### Sample Output

|  |  |
| --- | --- |
| Random Private Key (64 hex digits) | 7e4670ae70c98d24f3662c172dc510a085578b9ccc717e6c2f4e547edd960a34 |
| Corresponding Public Key (2 \* 64 hex digits) | c74a8458cd7a7e48f4b7ae6f4ae9f56c5c88c0f03e7c59cb4132b9d9d1600bba  **…** |
| Compressed Public Key (65 hex digits) | c74a8458cd7a7e48f4b7ae6f4ae9f56c5c88c0f03e7c59cb4132b9d9d1600bba1 |
| Blockchain Address (40 hex digits) | c3293572dbe6ebc60de4a20ed0e21446cae66b17 |

Use programming languages, libraries and tools of choice.

## Existing Private Key to Address

Write a program to **calculate a blockchain address** by given private key. Use the same format for the keys and addresses like in the previous exercise.

### Sample Input / Output

|  |
| --- |
| **Input (Private Key, hex encoded)** |
| fe549dbcccfbd11e255f6037e1e640efaca0e19966ac77a592fdf06d295952a4 |
| **Output** |
| Extracted public key: b42aed0ff6c546ac1c6601713af035746db146073f4e925dd8a18829c8db3adf1  Extracted blockchain address: dd09ae9bdb8e6dd8ecf0631f44a8566346eadf50 |

Use programming languages, libraries and tools of choice.