

---

## Overview

---

In this challenge, you will generate addresses from a descriptor, and calculate the total balance of these addresses. The process involves parsing a given descriptor to create addresses, fetching transactions associated with these addresses from esplora, and calculating the balance based on the fetched transaction data. The final balance should be outputted in Bitcoin (BTC) and saved to a file named out.txt.

## Objective

---

The objective of this challenge is to:

- **Address Generation:** Parse a given descriptor to generate addresses.
- **Gap Handling:** Continue generating addresses until a sequence of 10 unused addresses (addresses with no transactions) is detected.
- **Transaction Fetching:** Use the Esplora API to fetch transaction data for each generated address.
- **Balance Calculation:** Calculate the total balance of all addresses based on the fetched transaction data.

The output of your script should be a file named `out.txt` that follows a specific format.

Place your solution in the appropriate directory based on your chosen language:

- [bash](#)
- [javascript](#)
- [python](#)
- [rust](#)

## Requirements

---

### Input

- You have to calculate the total balance of the addresses generated from the descriptor:

```
wpkh(tpubD6NzVbkrYhZ4XgiXtGrdW5XDAPFCL9h7we1vwNCpn8tGbBcgfVYjXyhWo4E1xkh56hjod1RhGjxbaTLV3X4FyWuejifB9jusQ46QzG87VKp/*)#adv567t2 .
```

- After running the `docker-compose`, Esplora API will be available locally on: `http://localhost:8094/regtest/api/`. Read the [Esplora API documentation](#) for more information and the specific API calls you will need. See the instructions below for running `docker-compose`.

## Output

Your script must generate an output file named `out.txt` with a single line containing the total balance of the addresses in Bitcoin (BTC).

## Execution

---

To test your solution locally:

- Uncomment the line corresponding to your language in [run.sh](#).
- Set execution permission: `chmod +x ./local.sh`.
- Execute [local.sh](#).

If your code works, you will see the test completed successfully.

## Running Esplora Locally

The `./data` folder contains the blocks with the required transactions that you need to fetch via esplora. The `docker-compose.yaml` file contains the necessary scripts to run a `bitcoind` and `esplora` node with the block files. You can locally start/stop the esplora server while writing your solution.

To start Esplora locally, run the following command:

```
/bin/bash start-esplora.sh
```



### Important

If you get the following error: `no matching manifest for ... in the manifest list entries`, uncomment the `platform: linux/amd64` line in the `docker-compose.yaml` file.

To stop Explora, run the following command:

```
/bin/bash stop-esplora.sh
```



## Evaluation Criteria

---

Your submission will be evaluated based on:

- **Autograder:** Your code must pass the autograder [test script](#).
- **Explainer Comments:** Include comments explaining each step of your code.
- **Code Quality:** Your code should be well-organized, commented, and adhere to best practices.

## Plagiarism Policy

Our plagiarism detection checker thoroughly identifies any instances of copying or cheating. Participants are required to publish their solutions in the designated repository, which is private and accessible only to the individual and the administrator. Solutions should not be shared publicly or with peers. In case of plagiarism, both parties involved will be directly disqualified to maintain fairness and integrity.

## AI Usage Disclaimer

You may use AI tools like ChatGPT to gather information and explore alternative approaches, but avoid relying solely on AI for complete solutions. Verify and validate any insights obtained and maintain a balance between AI assistance and independent problem-solving.

## Why These Restrictions?

---

These rules are designed to enhance your understanding of the technical aspects of Bitcoin. By completing this assignment, you gain practical experience with the technology that secures and maintains the trustlessness of Bitcoin. This challenge not only tests your ability to develop functional Bitcoin applications but also encourages deep engagement with the core elements of Bitcoin technology.