1. Generowanie portfela blockchain (Klucz prywatny, publiczny i adres)

Opis zadania:
Stwórz funkcję w Pythonie, która generuje nowy portfel blockchain. Portfel powinien składać się z:

- Klucza prywatnego (32 bajty losowych danych),
- Klucza publicznego (wyliczanego na podstawie klucza prywatnego za pomocą algorytmu ECDSA z krzywą SECP256k1),
- Adresu blockchain (adres Bitcoin-style na podstawie klucza publicznego, wykorzystujący SHA-256 i RIPEMD-160).

Cele:

- Wygenerowanie kluczy prywatnego i publicznego,
- Wygenerowanie adresu na podstawie klucza publicznego,
- Zapewnienie bezpieczeństwa (nigdy nie przechowywać klucza prywatnego w sposób niezaszyfrowany).

---

2. Podpisywanie wiadomości (transakcji)

Opis zadania:
Użyj klucza prywatnego do podpisania wiadomości, którą będziesz traktować jako transakcję. Podpis powinien być wygenerowany za pomocą algorytmu ECDSA, używając klucza prywatnego portfela.

Cele:

- Zaimplementowanie funkcji do podpisywania wiadomości przy użyciu klucza prywatnego,
- Zwrócenie podpisu w formacie hex (w postaci ciągu znaków),
- Walidacja, że podpis jest poprawny.

---

3. Weryfikacja podpisu transakcji

Opis zadania:
Zaimplementuj funkcję, która weryfikuje podpis transakcji przy użyciu klucza publicznego. Funkcja powinna sprawdzać, czy podpis jest ważny i czy odpowiada wiadomości.

Cele:

- Weryfikacja podpisu transakcji przy użyciu klucza publicznego,
- Weryfikacja, czy podpis pasuje do wiadomości.

---

4. Implementacja transakcji (złożenie transakcji)

Opis zadania:
Stwórz mechanizm, który łączy podpisane transakcje z adresem nadawcy i odbiorcy. Transakcja powinna zawierać:

- Adres nadawcy (twój portfel),
- Adres odbiorcy,
- Kwotę,
- Podpis transakcji.

Cele:

- Zdefiniowanie struktury transakcji (nadawca, odbiorca, kwota, podpis),
- Implementacja funkcji do tworzenia transakcji.

---

5. Zapisywanie transakcji w pamięci lub pliku (Symulacja blockchaina)

Opis zadania:
Zaimplementuj prosty system, który zapisuje transakcje w formie "bloków" w strukturze danych (np. lista lub baza danych). Każda transakcja powinna zawierać:

- Identyfikator transakcji,
- Adresy nadawcy i odbiorcy,
- Wartość (ilość "kryptowaluty"),
- Podpis transakcji.

Cele:

- Symulowanie zapisów transakcji w lokalnym systemie (baza danych, plik JSON),
- Zapewnienie struktury danych do przechowywania transakcji.