Aufgabe 3

Wir können die Menge B nicht trivial entscheiden da wir unendlich viele Primzahlkombinationen durchgehen müssen um für eine Zahl sicher sagen zu können das sie nicht in B liegt.
Wir definieren zunächst eine Hilfsfunktion $h'(x,t)$, Das folgende Programm zeigt, wie h' berechnet wird.

```python
    """This program shows how to compute h'"""

def is_prime(x):
    return False if x < 2 else all([x % i != 0 for i in range(2, x - 1)])


def get_next_prime(x):
    x += 1
    while not is_prime(x):
        x += 1
    return x

def h(x, t):

    p = 2
    q = 2

    for _ in range(t):

        if p - q == x:
            return x

        if p == q:
            q = 2
            p = get_next_prime(p)
        else:
            q += 1
    return 2

for curr in range(100):
    print(f" {curr}: {h(curr, 1000)}")
```

Nun können wir $h : \mathbb{N} \to \mathbb{N}$ definieren:
Wir benutzen die Cantorsche Paarungsfunktion so, dass wir ein gegbenes x in ihre beiden Komponenten zerlegen und als input für die Funktion h'(x, t) benutzen. Da h'(x, t) immer stoppt ist die Funktion berechenbar (stoppt nach t iterationen). Der Wertebereich ist passend da h'(x, t) für jedes x, falls es in b liegt, für ein genügend großes t den richtigen Wert annimmt.