1. Implement (coding in any programming language) the attacks described in this chapter for the shift cipher or the Vigenere cipher.

Shift cipher:

```
# Frequency of characters
pattens_list = [8.2, 1.5, 2.8, 4.3, 12.7, 2.2, 2, 6.1, 7,
                0.2, 0.8, 4, 2.4, 6.7, 1.5, 1.9, 0.1, 6.0,
                6.3, 9.1, 2.8, 1, 2.4, .2, 2, .1]


def attack(code: str):
    # Calculation: 0.06025
    code = code.lower()
    qh = sum([(i * 0.01) ** 2 for i in pattens_list])
    text_dict = dict.fromkeys(range(26), 0)
    for i in code:
        text_dict[(ord(i) - ord('a')) % 26] += 1
    result_list = []
    for i in range(26):
        # calculate I_j
        text_res = sum([(pattens_list[j] * 0.01) * (text_dict[(j + i) % 26] / len(code))
for j in text_dict.keys()])
        result_list.append(text_res)
    abs_min = min([abs(i - qh) for i in result_list])
    abs_dis = [abs(i - qh) for i in result_list]
    for i in range(len(abs_dis)):
        if abs_dis[i] == abs_min:
            shift = i
    # get the closest value
    result = ""
    for i in code:
        result += chr((ord(i) - ord('a') - shift) % 26 + ord('a'))
    return result


if __name__ == '__main__':
    test_case =
"Likhkdgdqbwklqjfrqilghqwldowrvdbkhzurwhlwlqflskhuwkdwlvebvrfkdqjlqjwkhrughuriwkhohwwhu"
    print(attack(test_case))
```

2. Show that the shift, substitution, and Vigenère ciphers are all trivial to break using a chosen-plaintext attack. How much chosen plaintext is needed to recover the key for each of the ciphers?

> **Shift:**

> Only one pair of plaintext/ciphertext is needed for the recovery. For a plaintext $m = m_1 \cdots m_l$ and ciphertext $c = c_1 \cdots c_l$. k can be deduced by the shift from any corresponding character c_i to m_i. For example, $m =$ "abcd" and $c =$"bcde", the shift is the shift from b to a, which equals to 1. This is trivial to break.
>
> **Substitution**:
>
> This need at least the number of characters plaintext. If we only consider 26 alphabet, then 26 plaintext/ciphertext pairs should be included to get the permutation map. Therefore, the map from plaintext to ciphertext is deduced by this attack. It is trivial to break by using chosen-plaintext attack.
>
> **Vigenere cipher:**
>
> At least one pair. If the pair of plaintext/ciphertext is long enough for the key(repeated), it is similar to shift cipher. For a plaintext $m = m_1 \cdots m_l$, ciphertext $c = c_1 \cdots c_l$ and key $k = k_1 k_2 \cdots k_t k_1 k_2 \cdots k_l$, then $k_i = c_i - m_i$ for $1 \leq i \leq t$ and repeated for the remaining plaintext. If the length of the chosen pair is less than a key instance, it need more than one pair. Otherwise, one chosen pair is enough to recover the key.

3. Does composing multiple substitution ciphers one after the other improve security than using a single substitution cipher?

> No. Assume there is a plaintext m, after a substitution, the text will become s_1 until i times, then we have intermediate text $s_1, s_2 \cdots s_{i-1}$, the final ciphertext is c. Actually, all of the substitution can be considered as one substitution from m to c without s_i. In essence, the key that defines the permutation from m to c has no much difference with other permutation like just from m to s_1 in substitution cipher. Hence, it will not improve security and it still perform one substitution actually.