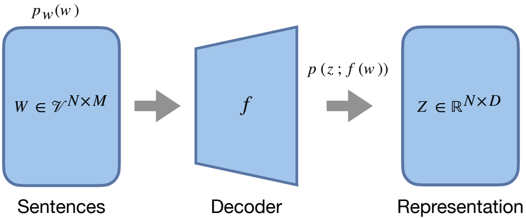
a.

```
def construct_z():
    # 1. Describe Z using a compressed code W
    def p_w(w):  # K(p_W) = length of this function
        ... # returns a probability
    w_compressed = [...]  # K(W | p_W) = ∑ − log p_W(w_n)
    w = decode_algo(w_compressed, p_w)  # K = small constant

    # 2. Decode Z from W
    def f(w):  # K(f) = length of this function
        ... # returns mean and std of a normal distribution
    def p_normal(mu, std):  # K = small constant
        ... # returns a probability
    z_mu, z_std = f(w)  # K = small constant
    z_correction = [...]  # K(Z | W, f) = ∑ − log 𝒩(z_n; z_n^μ, z_n^σ)
    z = decode_algo(z_correction, p_normal, z_mu, z_std)  # K = small constant

    return z
```

b.



c.

$$K(Z) \approx \min_{p_W, W, f} K(p_W) + K(W \mid p_W) + K(f) + K(Z \mid W, f)$$