3. We will prove that the dynamic programming approach used in the changedp algorithm is correct by induction. That is, we will show that

$$T[v] = \min_{V[i] \leq v} \{T[v - V[i]] + 1\}, T[0] = 0$$

is the minimum number of coins needed to make change for the amount v.

Let $V = [v_1, v_2, \ldots, v_n]$ be a vector containing distinct integer values of the coins in ascending order. Let T be an array of length $v + 1$, where $T[i]$ is the minimum number of coins needed to make the value i. Note that 0 is the minimum number of coins needed to make the amount 0. Since $T[0] = 0$, the base case holds. Now, assume that for some positive integer k,

$$T[j] = \min_{V[i] \leq j} \{T[j - V[i]] + 1\}$$

for each j where $1 \leq j \leq k$. We want to show that

$$T[k+1] = \min_{V[i] \leq k+1} \{T[k + 1 - V[i]] + 1\}.$$

Now, to determine $T[k + 1]$ we check all values $T[k + 1 - V[i]]$ where $k + 1 \leq V[i]$. These are all the possible values such that a single coin can be added to them to get the amount $k + 1$. If none of these values are possible to make (so they are set to -1), then $k + 1$ is not possible to make and it will be set to -1. Otherwise, the values $T[k + 1 - V[i]]$ that can be made using the given coin values are all the possible intermediate values from which $k + 1$ can be reached by adding a single coin. By assumption, these values are already minimal. So, by taking the minimum of these values and adding 1, we get the minimum number of coins that can be used to reach $k + 1$. This completes the proof.