Question 1:

a) Write down the code of Program $P$ in the language $L$ for which $\#(P) = 72899$.

$72900 = 2^2 \cdot 3^6 \cdot 5^2 = [2, 6, 2]$

This means that this program has three instructions, and their codes are 2, 6, and 2.

First instruction:

$2 = <0, 1> = <0, <1, 0>>$, so $a = 0$, $b = 1$, and $c = 0$.

This means the instruction is unlabeled, the instruction type is $V \leftarrow V + 1$, and the variable is $Y$, so we have:

$$Y \leftarrow Y + 1$$

Second instruction:

$6 = <0, 3> = <0, <2, 0>>$, so $a = 0$, $b = 2$, and $c = 0$.

This means the instruction is unlabeled, the instruction type is $V \leftarrow V - 1$, and the variable is $Y$, so we have:

$$Y \leftarrow Y - 1$$

Third instruction:

Same as the first one:

$$Y \leftarrow Y + 1$$
The whole program is then:

\[
Y \leftarrow Y + 1 \\
Y \leftarrow Y - 1 \\
Y \leftarrow Y + 1
\]

b) What is the number of the following program?

[D] \[
Y \leftarrow Y + 1 \\
Z5 \leftarrow Z5 + 1 \\
X \leftarrow X - 1 \\
IF X \neq 0 \text{ GOTO D}
\]

First instruction:
\[a = 4, b = 1, c = 0\]
\[<4, 1, 0>> = <4, 1> = 47\]

Second instruction:
\[a = 0, b = 1, c = 10\]
\[<0, 1, 10>> = <0, 41> = 82\]

Third instruction:
\[a = 0, b = 2, c = 1\]
\[<0, 2, 1>> = <0, 11> = 22\]

Fourth instruction:
\[a = 0, b = 6, c = 1\]
\[<0, 6, 1>> = <0, 191> = 382\]

So the number of the program is \([47, 82, 22, 382] - 1 = 2^{47} \cdot 3^{82} \cdot 5^{22} \cdot 7^{382} - 1\)

(which, by the way, is

30001575654646469639105616103774432865494129589140900946540035005249739078874
235748729500754557213249339012819321874029520223973621388872513287890734934
096360280658700359135940289889694160006417405858435874548724735689132490025
312833966965009445024648492571892758905908073433028815180588910351316261456
549122613951875734208388387950231869250446740759996451278047811941171199999
99999999999999999)
Question 2:

We have seen the trick of using the pairing function to associate numbers not only with pairs of numbers but also triplets. For example, here we create a bijection between \( z \) and triplets \((a, b, c)\):

\[ z = \langle a, \langle b, c \rangle \rangle \]

We could extend this further for quadruples, for example, a bijection between \( z \) and quadruples \((a, b, c, d)\):

\[ z = \langle a, \langle b, \langle c, d \rangle \rangle \rangle \]

Obviously, we could modify this trick for any tuple size. Does this mean that we would not need the Gödel numbers for enumerating programs in \( L \) but could use the pairing function instead? Give a reason for your answer.

If we wanted to use the pairing function for defining the number of a program based on the numbers of its instructions, we would run into a serious problem: While the pairing function is able, as seen above, to associate numbers with tuples of any size, this association is bijective (i.e., unique in both directions) if and only if we use a fixed tuple size. However, programs can have any finite number of instructions.

To illustrate this point, let us take another look at the answer to Question 1a, second instruction. There we have:

\[ 6 = \langle 0, 3 \rangle \]

Now let us assume that we wanted to the pairing function in this way to define the number of a program based on the numbers of its instructions. This would mean that the program with number 6 would consist of two instructions with numbers 0 and 3. So far, so good, but we are now facing an ambiguity, because our program may have three instructions instead of two. If we look at Question 1a again, we see:

\[ 6 = \langle 0, 3 \rangle = \langle 0, \langle 2, 0 \rangle \rangle \]

So based on our method, the program with number 6 could as well consist of three instructions, namely 0, 2, and 0. We could not be sure which one it is, which means that our method does not provide a one-to-one correspondence between the number of a program and the numbers of its instructions.

The pairing function and its nested versions provide one-to-one correspondences between numbers and tuples, as long as these tuples are known to always have the same size. For example, \( z = \langle x, y \rangle \) is a one-to-one correspondence between numbers \( z \) and pairs \((x, y)\), and \( p = \langle a, \langle b, c \rangle \rangle \) is a one-to-one correspondence between numbers \( p \) and triplets \((a, b, c)\).
If we want a one-to-one correspondence between numbers and tuples of any size, we can use Gödel numbers instead. No two distinct tuples (sequences) of any length have the same Gödel number, except if their only difference is an additional sequence of zeros at the end of one of the tuples. In our example, a program with number 6 would have the instructions 0, 0, 0, and 1, because $6 = 2^0 \cdot 3^0 \cdot 5^0 \cdot 7^1 - 1$. There is no ambiguity, even if we do not know the length of the program in advance.