CSE 16
Lab Assignment 1

Our goal in this assignment is to determine the set of all truth-value assignments that satisfy the following logical expression.

$$(*) \qquad \Big(\big((q \to r)\oplus(s \to t)\big) \to \big((u \to v)\oplus(w \to x)\big)\Big) \leftrightarrow (y \wedge z)$$

Let us begin with a simpler expression for the purposes of illustration. Consider $(p \to q)\oplus r$. Using the symbol 1 for True and 0 for False, the truth table for this expression is

p	q	r	$p \to q$	$(p \to q)\oplus r$
0	0	0	1	1
0	0	1	1	0
0	1	0	1	1
0	1	1	1	0
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	0

Each truth-value assignment to the propositional variables p, q and r can thus be represented as a bit string in the order $p\ q\ r$. Such an assignment is said to *satisfy* an expression if it makes the expression True. We call an expression *satisfiable* if there exists at least one truth-value assignment that satisfies it, i.e. if the expression is not a contradiction. The truth-value assignments that satisfy $(p \to q)\oplus r$ are therefore

0 0 0
0 1 0
1 0 1
1 1 0

We will represent truth-value assignments to the propositional variables in (*) as bit strings of length 10 given in the order

$$q\ r\ s\ t\ u\ v\ w\ x\ y\ z$$

One can check that the assignment

0 1 1 0 1 1 1 0 0 1

does not satisfy (*), and that the assignment

0 0 0 0 1 0 1 0 1 1

does satisfy (*). The expression (*) is thus a *contingency*, i.e. neither a tautology nor a contradiction. In particular (*) is satisfiable.

Let m be the number of truth-value assignments satisfying (*). We have seen that $m \geq 1$, and it so happens that $m \geq 100$. Your task is to determine all m such truth-value assignments. One way to proceed would be to construct a truth table for (*), but that table would contain $2^{10} = 1024$ rows. A daunting task, but not impossible. Another approach is to write a program that systematically produces all 1024 bit strings of length 10, and print only those that satisfy (*). Another approach might be to carefully analyze the expression (*) to understand its structure as a logical expression, and thereby avoid constructing the full table.

In this project, you will produce a text file called **`lab1.txt`** that begins with a short paragraph describing your method (algorithm) for solving the problem. This paragraph will be followed by m lines, each line containing one of the required bit strings. Each string will be presented as a space-separated string of 0's and 1's, and nothing else. Furthermore, the lines will be listed in numerical order, considering the bit strings as binary numerals in the range 0 to $2^{24} - 1 = 1023$. Submit your file to the assignment name lab1 on Gradescope before the deadline.