Problem #1

Given a set of *n*, 1-bit numbers, the *max* of those *n* numbers is equal to 0b1 if and only if **any** of those numbers are 0b1. In contrast, the *min* of the *n* 1-bit numbers is 0b1 if and only if **all** *n* numbers are 0b1. Hence, for *n* 1-bit numbers, the *max* and *min* can be written as:

$max_1 = \bigvee_{i=1}^{n} V_i$	$min_1 = \bigwedge_{i=1}^{n} V_i$	(1)

where V_i is the i^{th} 1-bit number in the set of *n* numbers.

As an extension of the equation (1) above, consider the case of a set of *k*-bit numbers (where $k > 1$). Since any integer is composed of a series of bits, determining the minimum and maximum of the *n* numbers can be done at the bit level. The most significant bit (MSB) of the max/min is calculated using equation (1); this is because MSB of the maximum/minimum only depends on the MSBs of the set of *n* numbers. For all subsequent bits, *j* (where *j* is a bit index defined as $1 < j \leq k$), extra caution must be shown because it is not enough to only look at the j^{th} bit of the *n* numbers since a number, V_i, is only relevant to a min/max operation for bit *j* if all of its previous, more significant bits (i.e. $< j$ and greater than 0) are equivalent to the corresponding bits in the min/max number.

Example: Determine the maximum of a set of three, two-digit numbers { 0b00, 0b10, 0b01 }[1]

To determine the MSB of the maximum, you consider the most significant bit of all three words in the set, which are {0b0, 0b1, 0b0} respectively. Hence, the most significant bit (MSB) of the maximum (i.e. OR) of three numbers is 0b1. When determining the value of the second bit (i.e. least significant in this case) for the maximum, one can only consider those numbers who preceding bits are equal (same) to the previous, more significant bits in *max*. To check for equivalence, use the operator, ⇔, which is defined as:

$(A \Leftrightarrow B) \equiv (A \wedge B) \vee (\bar{A} \wedge \bar{B})$	(2)

In simpler terms, both bits must be 0b0 or both bits must be 0b1. If one did not check equivalence and used the equation (1) as is, the *max* would erroneously be: 0b11 (with the MSB coming from word 0b10 and the LSB coming from word 0b01).

When one combines the requirement of checking preceding bit equivalence with equation (1), the complete Boolean expression to find the j^{th} bit of the minimum and maximum of a set of *n* numbers is:

$max_j = \bigvee_{i=1}^{n} \left(V_{i,j} \wedge \left(\bigwedge_{k=1}^{j-1} (max_k \Leftrightarrow V_{i,k}) \right) \right)$, for $j > 1$	$min_j = \bigwedge_{i=1}^{n} \left(V_{i,j} \vee \left(\bigvee_{k=1}^{j-1} \overline{(min_k \Leftrightarrow V_{i,k})} \right) \right)$, for $j > 1$	(3)

where $V_{i,j}$ is j^{th} bit of the i^{th} number in the set of *n* numbers.

Minimax is a logical extension of equation (3). Each node in the tree has three children. Depending on whether the node is in a *max* or *min* level, it applies the corresponding equation from (3) on its children to determine the respective value of each of its *k*-bits. For this problem, *k* (i.e. bits per word) is 3 since the numbers are 3-bits in length, and size of each set of numbers, *n*, is 3 (i.e. the number of children). The following are the bit equations for each of the values in the 4 ply minimax as defined in equation (3):

[1] The case of *min* is logical extension of the *max* example and is not shown. However, its equation is provided in (**Error! Main Document Only.**).

Max – Level 0 (Root of the Tree)

$$V_1 = \bigvee_{i=1}^{3} V_{i,1}$$	(4)
$$V_2 = \bigvee_{i=1}^{3} (V_1 \Leftrightarrow V_{i,1}) \wedge V_{i,2}$$	(5)
$$V_3 = \bigvee_{i=1}^{3} (V_1 \Leftrightarrow V_{i,1}) \wedge (V_2 \Leftrightarrow V_{i,2}) \wedge V_{i,3}$$	(6)

Min – Level 1

$$V_{i,1} = \bigwedge_{j=i}^{3} V_{i,j,1}$$	(7)
$$V_{i,2} = \bigwedge_{j=1}^{3} \overline{(V_{\iota,1} \Leftrightarrow V_{\iota,J,1})} \vee V_{i,j,2}$$	(8)
$$V_{i,3} = \bigwedge_{j=1}^{3} \overline{(V_{\iota,1} \Leftrightarrow V_{\iota,J,1})} \vee \overline{(V_{\iota,2} \Leftrightarrow V_{\iota,J,2})} \vee V_{i,j,3}$$	(9)

Max – Level 2 (Root of the Tree)

$$V_{i,j,1} = \bigvee_{i=1}^{3} V_{i,j,k,1}$$	(10)
$$V_{i,j,2} = \bigvee_{i=1}^{3} (V_{i,j,1} \Leftrightarrow V_{i,j,k,1}) \wedge V_{i,j,k,2}$$	(11)
$$V_{i,j,3} = \bigvee_{i=1}^{3} (V_{i,j,1} \Leftrightarrow V_{i,j,k,1}) \wedge (V_{i,j,2} \Leftrightarrow V_{i,j,k,2}) \wedge V_{i,j,k,3}$$	(12)

Min – Level 3

$$V_{i,j,k,1} = \bigwedge_{j=i}^{3} V_{i,j,k,m,1}$$	(13)
$$V_{i,j,k,2} = \bigwedge_{j=1}^{3} \overline{(V_{\iota,J,k,1} \Leftrightarrow V_{\iota,J,k,m,1})} \vee V_{i,j,k,m,2}$$	(14)
$$V_{i,j,k,3} = \bigwedge_{j=1}^{3} \overline{(V_{\iota,J,k,1} \Leftrightarrow V_{\iota,J,k,m,1})} \vee \overline{(V_{\iota,J,k,2} \Leftrightarrow V_{\iota,J,k,m,2})} \vee V_{i,j,k,m,3}$$	(15)

$V_{1,b}$ can be modified to use only symbols $V_{i,j,k,m,b}$ by doing successive substitution using the equations above. This can be done either by hand, which has limited scalability or though automation. I chose to automate this process by writing a program to do the substitution and then convert to CNF. The program is written in Python (HW3_Q1_Sympy_Solver.py) and uses the Python SymPy library. SymPy is open source and can be downloaded from https://pypi.python.org/pypi/sympy. Included with this homework is a text file "**HW3_Q1_Results.txt**" which is the output of my SymPy program. It contains $V_{i,b}$ both in non-CNF form as well as in CNF form; these Boolean expressions

are in terms of the base proposition symbols (i.e. $V_{i,j,k,m,b}$) as required. Most of the outputs of the equations in both CNF and non-CNF formats are too long to reasonably appear in this document. Kindly refer to that as a reference.

Appendix A contains the CNF for $V_{1,1}$. This is given as an example of the tool's output. "`Not`" operations are represented with a exclamation point ("`!`"); "`Or`" operations are denoted with a plus sign ("+") while "`And`" operations are denoted with an ampersand ("&"). The canonical form for $V_{1,1}$ as shown in Appendix A is:

$$\bigwedge_{j=1}^{3}\left(\bigwedge_{r=1}^{3}\left(\bigwedge_{s=1}^{3}\left(\bigwedge_{t=1}^{3}\left(V_{1,j,1,r,i} \vee V_{1,j,1,s,i} \vee V_{1,j,1,t,i}\right)\right)\right)\right)$$

Problem #2

Question: The pigeonhole principle PHP_n^{n+1} says any function from $n+1$ pigeons into n holes must result in two pigeons in the same hole. Let $P_{i,j}$ be a variable expressing that pigeon i gets mapped to hole j. Consider the $n=3$ case.

Express the following as propositional formulas:

a. **Every i gets mapped to some j.**
b. **Some j is mapped to by i and i' where $i \neq i'$.**

The conjunction of these two statements is a propositional formula for PHP_3^4. Convert $\rightarrow PHP$ to clausal form and give a resolution refutation for this statement. Finally, trace the execution of DPLL on this formula.

Part A:

$$R_1: \bigwedge_{i=1}^{4} \left(P_{i,1} \wedge \overline{P_{i,2}} \wedge \overline{P_{i,3}}\right) \vee \left(\overline{P_{i,1}} \wedge P_{i,2} \wedge \overline{P_{i,3}}\right) \vee \left(\overline{P_{i,1}} \wedge \overline{P_{i,2}} \wedge P_{i,3}\right) \quad (16)$$

Part B:

$$R_2: \bigvee_{j=1}^{3} \left(P_{1,j} \wedge P_{2,j}\right) \vee \left(P_{1,j} \wedge P_{3,j}\right) \vee \left(P_{1,j} \wedge P_{4,j}\right) \vee \left(P_{2,j} \wedge P_{3,j}\right) \vee \left(P_{2,j} \wedge P_{4,j}\right) \vee \left(P_{3,j} \wedge P_{4,j}\right) \quad (17)$$

PHP_3^4:

$$(R_1 \wedge R_2) \quad (18)$$

$\rightarrow PHP_3^4$:

$$\rightarrow (R_1 \wedge R_2) \quad (19)$$

I wrote a Python program (HW3_Q3_Sympy_Solver.py) to convert the equation above into CNF. It uses the SymPy toolkit.

$$\bigwedge_{i=1}^{3}\left(\bigwedge_{j=1}^{3}\left(\bigwedge_{k=1}^{3}\left(\bigwedge_{m=1}^{3}\left(\overline{P_{1,i}} \vee \overline{P_{2,j}} \vee \overline{P_{3,k}} \vee \overline{P_{4,m}} \vee \bigvee_{\substack{1\le q\le 3\\ q\neq i}}\left(P_{1,q}\right) \vee \bigvee_{\substack{1\le r\le 3\\ r\neq j}}\left(P_{2,r}\right) \vee \bigvee_{\substack{1\le s\le 3\\ s\neq k}}\left(P_{3,s}\right) \vee \bigvee_{\substack{1\le t\le 3\\ t\neq m}}\left(P_{4,t}\right)\right)\right)\right)\right) \quad (20)$$

Note: The notation in (20):

$$\bigvee_{\substack{1\le q\le 3\\ q\neq i}}\left(P_{1,q}\right) \quad (21)$$

Is the disjunction $P_{1,q}$ with the condition that q cannot equal i and that the range for q is defined as $1 \leq q \leq 3$. The other three disjunctions in (20) have similar meaning but for variables r, s, and t.

Appendix A – $V_{1,b}$ in Conjunctive Normal Form

Note: The CNF equations for problem #1 were solved using my program "HW3_Q1_Sumpy_Solver.py". I included the $V_{1,1}$ in CNF since it is length is manageable as an example of the tool's output.

$V_{1,1} =$

```
(V1,1,1,1,1+V1,1,2,1,1+V1,1,3,1,1)
&(V1,1,1,1,1+V1,1,2,1,1+V1,1,3,2,1)
&(V1,1,1,1,1+V1,1,2,1,1+V1,1,3,3,1)
&(V1,1,1,1,1+V1,1,2,2,1+V1,1,3,1,1)
&(V1,1,1,1,1+V1,1,2,2,1+V1,1,3,2,1)
&(V1,1,1,1,1+V1,1,2,2,1+V1,1,3,3,1)
&(V1,1,1,1,1+V1,1,2,3,1+V1,1,3,1,1)
&(V1,1,1,1,1+V1,1,2,3,1+V1,1,3,2,1)
&(V1,1,1,1,1+V1,1,2,3,1+V1,1,3,3,1)
&(V1,1,1,2,1+V1,1,2,1,1+V1,1,3,1,1)
&(V1,1,1,2,1+V1,1,2,1,1+V1,1,3,2,1)
&(V1,1,1,2,1+V1,1,2,1,1+V1,1,3,3,1)
&(V1,1,1,2,1+V1,1,2,2,1+V1,1,3,1,1)
&(V1,1,1,2,1+V1,1,2,2,1+V1,1,3,2,1)
&(V1,1,1,2,1+V1,1,2,2,1+V1,1,3,3,1)
&(V1,1,1,2,1+V1,1,2,3,1+V1,1,3,1,1)
&(V1,1,1,2,1+V1,1,2,3,1+V1,1,3,2,1)
&(V1,1,1,2,1+V1,1,2,3,1+V1,1,3,3,1)
&(V1,1,1,3,1+V1,1,2,1,1+V1,1,3,1,1)
&(V1,1,1,3,1+V1,1,2,1,1+V1,1,3,2,1)
&(V1,1,1,3,1+V1,1,2,1,1+V1,1,3,3,1)
&(V1,1,1,3,1+V1,1,2,2,1+V1,1,3,1,1)
&(V1,1,1,3,1+V1,1,2,2,1+V1,1,3,2,1)
&(V1,1,1,3,1+V1,1,2,2,1+V1,1,3,3,1)
&(V1,1,1,3,1+V1,1,2,3,1+V1,1,3,1,1)
&(V1,1,1,3,1+V1,1,2,3,1+V1,1,3,2,1)
&(V1,1,1,3,1+V1,1,2,3,1+V1,1,3,3,1)
&(V1,2,1,1,1+V1,2,2,1,1+V1,2,3,1,1)
&(V1,2,1,1,1+V1,2,2,1,1+V1,2,3,2,1)
&(V1,2,1,1,1+V1,2,2,1,1+V1,2,3,3,1)
&(V1,2,1,1,1+V1,2,2,2,1+V1,2,3,1,1)
&(V1,2,1,1,1+V1,2,2,2,1+V1,2,3,2,1)
&(V1,2,1,1,1+V1,2,2,2,1+V1,2,3,3,1)
&(V1,2,1,1,1+V1,2,2,3,1+V1,2,3,1,1)
&(V1,2,1,1,1+V1,2,2,3,1+V1,2,3,2,1)
&(V1,2,1,1,1+V1,2,2,3,1+V1,2,3,3,1)
&(V1,2,1,2,1+V1,2,2,1,1+V1,2,3,1,1)
&(V1,2,1,2,1+V1,2,2,1,1+V1,2,3,2,1)
&(V1,2,1,2,1+V1,2,2,1,1+V1,2,3,3,1)
&(V1,2,1,2,1+V1,2,2,2,1+V1,2,3,1,1)
&(V1,2,1,2,1+V1,2,2,2,1+V1,2,3,2,1)
&(V1,2,1,2,1+V1,2,2,2,1+V1,2,3,3,1)
&(V1,2,1,2,1+V1,2,2,3,1+V1,2,3,1,1)
&(V1,2,1,2,1+V1,2,2,3,1+V1,2,3,2,1)
&(V1,2,1,2,1+V1,2,2,3,1+V1,2,3,3,1)
&(V1,2,1,3,1+V1,2,2,1,1+V1,2,3,1,1)
&(V1,2,1,3,1+V1,2,2,1,1+V1,2,3,2,1)
&(V1,2,1,3,1+V1,2,2,1,1+V1,2,3,3,1)
&(V1,2,1,3,1+V1,2,2,2,1+V1,2,3,1,1)
```

```
&(V1,2,1,3,1+V1,2,2,2,1+V1,2,3,2,1)
&(V1,2,1,3,1+V1,2,2,2,1+V1,2,3,3,1)
&(V1,2,1,3,1+V1,2,2,3,1+V1,2,3,1,1)
&(V1,2,1,3,1+V1,2,2,3,1+V1,2,3,2,1)
&(V1,2,1,3,1+V1,2,2,3,1+V1,2,3,3,1)
&(V1,3,1,1,1+V1,3,2,1,1+V1,3,3,1,1)
&(V1,3,1,1,1+V1,3,2,1,1+V1,3,3,2,1)
&(V1,3,1,1,1+V1,3,2,1,1+V1,3,3,3,1)
&(V1,3,1,1,1+V1,3,2,2,1+V1,3,3,1,1)
&(V1,3,1,1,1+V1,3,2,2,1+V1,3,3,2,1)
&(V1,3,1,1,1+V1,3,2,2,1+V1,3,3,3,1)
&(V1,3,1,1,1+V1,3,2,3,1+V1,3,3,1,1)
&(V1,3,1,1,1+V1,3,2,3,1+V1,3,3,2,1)
&(V1,3,1,1,1+V1,3,2,3,1+V1,3,3,3,1)
&(V1,3,1,2,1+V1,3,2,1,1+V1,3,3,1,1)
&(V1,3,1,2,1+V1,3,2,1,1+V1,3,3,2,1)
&(V1,3,1,2,1+V1,3,2,1,1+V1,3,3,3,1)
&(V1,3,1,2,1+V1,3,2,2,1+V1,3,3,1,1)
&(V1,3,1,2,1+V1,3,2,2,1+V1,3,3,2,1)
&(V1,3,1,2,1+V1,3,2,2,1+V1,3,3,3,1)
&(V1,3,1,2,1+V1,3,2,3,1+V1,3,3,1,1)
&(V1,3,1,2,1+V1,3,2,3,1+V1,3,3,2,1)
&(V1,3,1,2,1+V1,3,2,3,1+V1,3,3,3,1)
&(V1,3,1,3,1+V1,3,2,1,1+V1,3,3,1,1)
&(V1,3,1,3,1+V1,3,2,1,1+V1,3,3,2,1)
&(V1,3,1,3,1+V1,3,2,1,1+V1,3,3,3,1)
&(V1,3,1,3,1+V1,3,2,2,1+V1,3,3,1,1)
&(V1,3,1,3,1+V1,3,2,2,1+V1,3,3,2,1)
&(V1,3,1,3,1+V1,3,2,2,1+V1,3,3,3,1)
&(V1,3,1,3,1+V1,3,2,3,1+V1,3,3,1,1)
&(V1,3,1,3,1+V1,3,2,3,1+V1,3,3,2,1)
&(V1,3,1,3,1+V1,3,2,3,1+V1,3,3,3,1)
```

Appendix B – $\rightarrow PHP_3^4$ in Conjunctive Normal Form

$\rightarrow PHP_3^4 =$

```
(!P11+!P21+!P31+!P41+P12+P13+P22+P23+P32+P33+P42+P43)
&(!P11+!P21+!P31+!P42+P12+P13+P22+P23+P32+P33+P41+P43)
&(!P11+!P21+!P31+!P43+P12+P13+P22+P23+P32+P33+P41+P42)
&(!P11+!P21+!P32+!P41+P12+P13+P22+P23+P31+P33+P42+P43)
&(!P11+!P21+!P32+!P42+P12+P13+P22+P23+P31+P33+P41+P43)
&(!P11+!P21+!P32+!P43+P12+P13+P22+P23+P31+P33+P41+P42)
&(!P11+!P21+!P33+!P41+P12+P13+P22+P23+P31+P32+P42+P43)
&(!P11+!P21+!P33+!P42+P12+P13+P22+P23+P31+P32+P41+P43)
&(!P11+!P21+!P33+!P43+P12+P13+P22+P23+P31+P32+P41+P42)
&(!P11+!P22+!P31+!P41+P12+P13+P21+P23+P32+P33+P42+P43)
&(!P11+!P22+!P31+!P42+P12+P13+P21+P23+P32+P33+P41+P43)
&(!P11+!P22+!P31+!P43+P12+P13+P21+P23+P32+P33+P41+P42)
&(!P11+!P22+!P32+!P41+P12+P13+P21+P23+P31+P33+P42+P43)
&(!P11+!P22+!P32+!P42+P12+P13+P21+P23+P31+P33+P41+P43)
&(!P11+!P22+!P32+!P43+P12+P13+P21+P23+P31+P33+P41+P42)
&(!P11+!P22+!P33+!P41+P12+P13+P21+P23+P31+P32+P42+P43)
&(!P11+!P22+!P33+!P42+P12+P13+P21+P23+P31+P32+P41+P43)
&(!P11+!P22+!P33+!P43+P12+P13+P21+P23+P31+P32+P41+P42)
&(!P11+!P23+!P31+!P41+P12+P13+P21+P22+P32+P33+P42+P43)
&(!P11+!P23+!P31+!P42+P12+P13+P21+P22+P32+P33+P41+P43)
&(!P11+!P23+!P31+!P43+P12+P13+P21+P22+P32+P33+P41+P42)
&(!P11+!P23+!P32+!P41+P12+P13+P21+P22+P31+P33+P42+P43)
&(!P11+!P23+!P32+!P42+P12+P13+P21+P22+P31+P33+P41+P43)
&(!P11+!P23+!P32+!P43+P12+P13+P21+P22+P31+P33+P41+P42)
&(!P11+!P23+!P33+!P41+P12+P13+P21+P22+P31+P32+P42+P43)
&(!P11+!P23+!P33+!P42+P12+P13+P21+P22+P31+P32+P41+P43)
&(!P11+!P23+!P33+!P43+P12+P13+P21+P22+P31+P32+P41+P42)
&(!P12+!P21+!P31+!P41+P11+P13+P22+P23+P32+P33+P42+P43)
&(!P12+!P21+!P31+!P42+P11+P13+P22+P23+P32+P33+P41+P43)
&(!P12+!P21+!P31+!P43+P11+P13+P22+P23+P32+P33+P41+P42)
&(!P12+!P21+!P32+!P41+P11+P13+P22+P23+P31+P33+P42+P43)
&(!P12+!P21+!P32+!P42+P11+P13+P22+P23+P31+P33+P41+P43)
&(!P12+!P21+!P32+!P43+P11+P13+P22+P23+P31+P33+P41+P42)
&(!P12+!P21+!P33+!P41+P11+P13+P22+P23+P31+P32+P42+P43)
&(!P12+!P21+!P33+!P42+P11+P13+P22+P23+P31+P32+P41+P43)
&(!P12+!P21+!P33+!P43+P11+P13+P22+P23+P31+P32+P41+P42)
&(!P12+!P22+!P31+!P41+P11+P13+P21+P23+P32+P33+P42+P43)
&(!P12+!P22+!P31+!P42+P11+P13+P21+P23+P32+P33+P41+P43)
&(!P12+!P22+!P31+!P43+P11+P13+P21+P23+P32+P33+P41+P42)
&(!P12+!P22+!P32+!P41+P11+P13+P21+P23+P31+P33+P42+P43)
&(!P12+!P22+!P32+!P42+P11+P13+P21+P23+P31+P33+P41+P43)
&(!P12+!P22+!P32+!P43+P11+P13+P21+P23+P31+P33+P41+P42)
&(!P12+!P22+!P33+!P41+P11+P13+P21+P23+P31+P32+P42+P43)
&(!P12+!P22+!P33+!P42+P11+P13+P21+P23+P31+P32+P41+P43)
&(!P12+!P22+!P33+!P43+P11+P13+P21+P23+P31+P32+P41+P42)
&(!P12+!P23+!P31+!P41+P11+P13+P21+P22+P32+P33+P42+P43)
&(!P12+!P23+!P31+!P42+P11+P13+P21+P22+P32+P33+P41+P43)
&(!P12+!P23+!P31+!P43+P11+P13+P21+P22+P32+P33+P41+P42)
&(!P12+!P23+!P32+!P41+P11+P13+P21+P22+P31+P33+P42+P43)
&(!P12+!P23+!P32+!P42+P11+P13+P21+P22+P31+P33+P41+P43)
&(!P12+!P23+!P32+!P43+P11+P13+P21+P22+P31+P33+P41+P42)
```

```
&(!P12+!P23+!P33+!P41+P11+P13+P21+P22+P31+P32+P42+P43)
&(!P12+!P23+!P33+!P42+P11+P13+P21+P22+P31+P32+P41+P43)
&(!P12+!P23+!P33+!P43+P11+P13+P21+P22+P31+P32+P41+P42)
&(!P13+!P21+!P31+!P41+P11+P12+P22+P23+P32+P33+P42+P43)
&(!P13+!P21+!P31+!P42+P11+P12+P22+P23+P32+P33+P41+P43)
&(!P13+!P21+!P31+!P43+P11+P12+P22+P23+P32+P33+P41+P42)
&(!P13+!P21+!P32+!P41+P11+P12+P22+P23+P31+P33+P42+P43)
&(!P13+!P21+!P32+!P42+P11+P12+P22+P23+P31+P33+P41+P43)
&(!P13+!P21+!P32+!P43+P11+P12+P22+P23+P31+P33+P41+P42)
&(!P13+!P21+!P33+!P41+P11+P12+P22+P23+P31+P32+P42+P43)
&(!P13+!P21+!P33+!P42+P11+P12+P22+P23+P31+P32+P41+P43)
&(!P13+!P21+!P33+!P43+P11+P12+P22+P23+P31+P32+P41+P42)
&(!P13+!P22+!P31+!P41+P11+P12+P21+P23+P32+P33+P42+P43)
&(!P13+!P22+!P31+!P42+P11+P12+P21+P23+P32+P33+P41+P43)
&(!P13+!P22+!P31+!P43+P11+P12+P21+P23+P32+P33+P41+P42)
&(!P13+!P22+!P32+!P41+P11+P12+P21+P23+P31+P33+P42+P43)
&(!P13+!P22+!P32+!P42+P11+P12+P21+P23+P31+P33+P41+P43)
&(!P13+!P22+!P32+!P43+P11+P12+P21+P23+P31+P33+P41+P42)
&(!P13+!P22+!P33+!P41+P11+P12+P21+P23+P31+P32+P42+P43)
&(!P13+!P22+!P33+!P42+P11+P12+P21+P23+P31+P32+P41+P43)
&(!P13+!P22+!P33+!P43+P11+P12+P21+P23+P31+P32+P41+P42)
&(!P13+!P23+!P31+!P41+P11+P12+P21+P22+P32+P33+P42+P43)
&(!P13+!P23+!P31+!P42+P11+P12+P21+P22+P32+P33+P41+P43)
&(!P13+!P23+!P31+!P43+P11+P12+P21+P22+P32+P33+P41+P42)
&(!P13+!P23+!P32+!P41+P11+P12+P21+P22+P31+P33+P42+P43)
&(!P13+!P23+!P32+!P42+P11+P12+P21+P22+P31+P33+P41+P43)
&(!P13+!P23+!P32+!P43+P11+P12+P21+P22+P31+P33+P41+P42)
&(!P13+!P23+!P33+!P41+P11+P12+P21+P22+P31+P32+P42+P43)
&(!P13+!P23+!P33+!P42+P11+P12+P21+P22+P31+P32+P41+P43)
&(!P13+!P23+!P33+!P43+P11+P12+P21+P22+P31+P32+P41+P42)
```