2019-2020 **AMERICAN COMPUTER SCIENCE LEAGUE** **Contest #3**

Senior Division - Veitch

1

	A	A	~A	~A	
B			X	X	~D
B			X	X	D
~B			X	X	D
~B			X	X	~D
	~C	C	C	~C	

2

	A	A	~A	~A	
B					~D
B					D
~B	X	X			D
~B	X	X			~D
	~C	C	C	~C	

3

	A	A	~A	~A	
B		X	X		~D
B					D
~B	X			X	D
~B					~D
	~C	C	C	~C	

4

	A	A	~A	~A	
B	X				~D
B	X				D
~B				X	D
~B					~D
	~C	C	C	~C	

PROBLEM: The Veitch Diagram is a method to represent a Boolean expression. A 4x4 grid can represent expressions with at most 4 variables. The method places an X in the cell(s) that describes each term. The ~ symbol is used to show negation. The following rules apply when adding X's to the cells:

1. A term of one variable fills 8 cells. The term ~A fills the 8 cells in Figure #1.
2. A term with 2 variables fills 4 cells. The term A~B fills the 4 cells in Figure #2.
3. A term of 3 variables fills 2 cells. The term BC~D fills the topmost 2 cells in Figure #3.
4. A term of 4 variables fills 1 cell. The term ~A~B~CD fills the single cell in Figure #4.

Variables are eliminated from a term's simplification if the variable and its negation are included. Terms are always joined by the OR symbol (+). AND within terms will always be implied.

Each X can be used just once in the forming of groups. Once the X's are correctly placed in the cells, it's possible to represent expressions. The representation is found by using the following priority rules:

1. Group 8 adjacent X's. This representation is shown in Figure #1. When 8 adjacent X's are grouped the representation is one term. The simplification of Figure #1 is ~A. Priority order is full rows top to bottom, full columns left to right, adjacent end-rows, and finally adjacent end-columns. See Sample #1.
2. Group 4 adjacent X's. This is shown in Figure #2. When 4 adjacent X's are grouped, the representation is 2 terms. Note that C and ~C and D and ~D are eliminated from the representation. The representation of Diagram 2 is A~B. Priority order is full rows, full columns, and finally blocks of 4 in each row left to right, then top to bottom.
3. Group 4 adjacent end-row X's, then 4 adjacent end-column X's. See Sample Input #5. Note that multiple adjacent end rows or end-columns can be combined to form a group of 4 adjacent X's.
4. Group all 4 corners which gives the expression ~C~D.
5. Group 2 adjacent X's. Priority order is top row to bottom row from left to right, then left-most column to right-most column from top to bottom. The term BC~D is the representation for the topmost 2 cells in Figure #3.

6. Group 2 adjacent end-row X's for each row from top to bottom, then 2 adjacent end-column X's for each column from left to right. End row adjacent X's are shown in Figure #3. The third row in the diagram has end-row adjacent X's. The representation is ~B~CD.
7. Simplify single X's that always translate to a term of 4 variables. The single X representation for Figure #4 is ~A~B~CD.

INPUT: There will be 5 lines of input (for clarity 10 lines of sample inputs are given). Each line will contain a 4-character string. The 4 characters will each represent a hexadecimal digit. When each is converted to a 4-digit binary number with leading zeros and placed in the diagram from top row to bottom row, the 1's will represent the placement of the X's in the Veitch Diagrams. The input for Figure #3 is 6090 which represents 0110, 0000, 1001 and 0000.

OUTPUT: For each line of input, print the expression according to the rules and priorities above. Since the representation must follow the rules above, the expression must be printed in the exact order shown. Also the factors of each term must be in ABCD order. Spacing between terms and within terms will not affect the answer.

SAMPLE INPUT:	**SAMPLE OUTPUT:**
FF33	1. B+~A~B
00CC	2. A~B
6090	3. BC~D+~B~CD
8810	4. AB~C+~A~B~CD
9008	5. B~C~D+A~B~C~D
F0B8	6. B~D+~A~BD+A~B~C
9699	7. ~B~C+BCD+B~C~D
8DD8	8. A~C+ACD+~A~CD
C3C3	9. AB~D+~ABD+A~BD+~A~B~D
F111	10. B~D+~A~CD+~A~B~C~D

Senior Division - Veitch

TEST DATA

TEST INPUT:

F620
F677
910A
4F74
2019

TEST OUTPUT (extra spaces don't matter):

1. B~D + BCD + ~A~BCD
2. C + ~A~B~C + B~C~D
3. ~AB~C + A~C~D + ~A~BC~D
4. BD + ~BCD + AC~D + ~A~B~CD
5. ~A~B~C + ~ABC~D + A~B~C~D