

```

1 # Sriram Chalamacharla
2 # Contest 3 2019-2020
3 # Junior Division - Veitch
4
5 inp1 = input("Enter Input: ")
6 inp2 = input("Enter Second Input: ")
7 inp3 = input("Enter Third Input: ")
8 inp4 = input("Enter Fourth Input: ")
9 inp5 = input("Enter Fifth Input: ")
10
11
12 def leading_zeros(num):
13     while True:
14         if len(str(num)) < 4:
15             num = "0" + str(num)
16         if len(str(num)) == 4:
17             break
18     return num
19
20
21 def mark_used(f, arr):
22     for i in arr:
23         f[i] = "0"
24
25
26 def add_terms(e1, e2):
27     res = ""
28     if e1 and e2:
29         res += e1 + " + " + e2
30     else:
31         if e1:
32             res += e1
33         if e2:
34             res += e2
35     return res
36
37
38 def group_four(first, last):
39     exp = ""
40     res = ""
41
42     # four adjX's in top row
43     top_row = 1
44     for i in range(4):
45         top_row = top_row and int(first[i])
46
47     if top_row:
48         exp = "B"
49         mark_used(first, range(4))
50
51     # four adjX's in bottom row
52     bot_row = 1
53     for i in range(4):
54         bot_row = bot_row and int(last[i])
55
56     if bot_row:
57         exp = exp + "~B"
58         mark_used(last, range(4))
59
60     # four adjX's in columns from left to right
61     for i in range(0, 4):
62         if int(first[i]) and int(last[i]):
63             # Adj X's
64             if i + 1 < 4 and int(first[i + 1]) and int(last[i + 1]):
65                 if i == 0:
66                     exp = exp + "A"
67                 elif i == 1:
68                     exp = exp + "C"
69                 elif i == 2:

```

```

70             exp = exp + "~A"
71             mark_used(first, [i, i + 1])
72             mark_used(last, [i, i + 1])
73         # End Column AdjX's
74         elif i == 0 and int(first[3]) and int(last[3]):
75             exp = exp + "~C"
76             mark_used(first, [0, 3])
77             mark_used(last, [0, 3])
78     res = add_terms(res, exp)
79     return res
80
81
82 def group_two(first, last):
83     res = ""
84
85     # Group2 adj X's in top row
86     exp3 = ""
87     for i in range(0, 4):
88         if i + 1 < 4 and int(first[i]) and int(first[i + 1]):
89             if i == 0:
90                 exp3 = exp3 + "AB"
91             elif i == 1:
92                 exp3 = exp3 + "BC"
93             elif i == 2:
94                 exp3 = exp3 + "~AB"
95             mark_used(first, [i, i + 1])
96         # Group2 End Column AdjX's
97         elif i == 0 and int(first[0]) and int(first[3]):
98             exp3 = exp3 + "B~C"
99             mark_used(first, [0, 3])
100    res = add_terms(res, exp3)
101
102    # Group2 adj X's in bottom row
103    exp4 = ""
104    for i in range(0, 4):
105        if i + 1 < 4 and int(last[i]) and int(last[i + 1]):
106            if i == 0:
107                exp4 = exp4 + "A~B"
108            elif i == 1:
109                exp4 = exp4 + "C~B"
110            elif i == 2:
111                exp4 = exp4 + "~A~B"
112            mark_used(last, [i, i + 1])
113        # Group2 End Column AdjX's
114        elif i == 0 and int(last[0]) and int(last[3]):
115            exp4 = exp4 + "~B~C"
116            mark_used(last, [0, 3])
117    res = add_terms(res, exp4)
118
119    exp2 = ""
120    # Group2 AdjX's in columns from left to right
121    for i in range(4):
122        if int(first[i]) and int(last[i]):
123            if i == 0:
124                exp2 = exp2 + "A~C"
125            elif i == 1:
126                exp2 = exp2 + "AC"
127            elif i == 2:
128                exp2 = exp2 + "~AC"
129            elif i == 3:
130                exp2 = exp2 + "~A~C"
131            mark_used(first, [i])
132            mark_used(last, [i])
133    res = add_terms(res, exp2)
134
135    return res
136
137
138 # Single X's

```

```

139 def single_one(first, last):
140     res = ""
141     exp5 = ""
142     exp6 = ""
143     for i in range(0, 4):
144         if int(first[i]):
145             if i == 0:
146                 exp5 = "AB~C"
147             elif i == 1:
148                 exp5 = "ABC"
149             elif i == 2:
150                 exp5 = "~ABC"
151             elif i == 3:
152                 exp5 = "~AB~C"
153             mark_used(first, [i])
154             res = add_terms(res, exp5)
155
156         if int(last[i]):
157             if i == 0:
158                 exp6 = "A~B~C"
159             elif i == 1:
160                 exp6 = "A~BC"
161             elif i == 2:
162                 exp6 = "~A~BC"
163             elif i == 3:
164                 exp6 = "~A~B~C"
165             mark_used(last, [i])
166             res = add_terms(res, exp6)
167     return res
168
169
170 def group(first, last):
171     res = ""
172     res = add_terms(res, group_four(first, last))
173     res = add_terms(res, group_two(first, last))
174     res = add_terms(res, single_one(first, last))
175     return res
176
177
178 def boolean_exp(inp):
179     # decimal to hex and hex to binary
180     top, bottom = bin(int(inp[:1], 16))[2:], bin(int(inp[1:], 16))[2:]
181     # add leading zeros
182     top, bottom = leading_zeros(top), leading_zeros(bottom)
183     # convert to list
184     top_row = list(top)
185     bottom_row = list(bottom)
186
187     result = group(top_row, bottom_row)
188     #return top + " " + bottom + ", " + result
189     return result
190
191
192 print("1. " + boolean_exp(inp1))
193 print("2. " + boolean_exp(inp2))
194 print("3. " + boolean_exp(inp3))
195 print("4. " + boolean_exp(inp4))
196 print("5. " + boolean_exp(inp5))
197

```