def findRow(cases):
    row = []
    if "B" in cases or "~B" in cases and "D" in cases or "~D" in cases:
        if "B" in cases and "D" in cases and "~B" not in cases and "~D" not in cases:
            row.append(1)
            return row
        elif "B" in cases and "~D" in cases and "~B" not in cases:
            row.extend((0,))
            return row
        elif "~B" in cases and "D" in cases and "~D" not in cases:
            row.append(2)
            return row
    else:
        row.append(3)
        return row

else:
    if "~B" in cases:
        row.extend((2, 3))
        return row
    elif "B" in cases:
        row.extend((0, 1))
        return row
    elif "~D" in cases:
        row.extend((0, 3))
        return row
    elif "D" in cases:
        row.extend((1, 2))
        return row
    else:
        row.extend((0, 1, 2, 3))
        return row

def finfCol(cases):
    col = []
    if "A" in cases or "~A" in cases and "C" in cases or "~C" in cases:
        if "A" in cases and "C" in cases and "~A" not in cases and "~C" not in cases:
            col.append(1)
            return col
        elif "A" in cases and "~C" in cases and "~A" not in cases:
            col.append(0)
            return col
        elif "~A" in cases and "C" in cases and "~C" not in cases:
            col.append(2)
            return col
    else:
        col.append(3)
        return col

else:
    if "~A" in cases:
        col.extend((2, 3))
        return col
    elif "A" in cases:
        col.extend((0, 1))
        return col
    elif "~C" in cases:
        col.extend((0, 3))
        return col
elif "C" in cases:
    col.extend((1, 2))
    return col
else:
    col.extend((0, 1, 2, 3))
    return col

def gridValues(input):
    a = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
    input = input.split("+")
    for i in input:
        rowOptions = findRow(i)
        colOptions = finfCol(i)
        if len(rowOptions)>len(colOptions):
            times = len(rowOptions)
        else:
            times = len(colOptions)
        for i in rowOptions:
            for j in colOptions:
                a[i][j] = 1
    return a

def hexConvert(list):
    answer = []
    num=""
    for i in list:
        for j in i:
            num = num+str(j)
        answer=answer+str(hex(int(num,2))"
        if len(answer)<4:
            if (4-len(answer))==1:
                answer = "0"+answer
            elif (4-len(answer)) ==2:
                answer = "00"+answer
            else:
                answer = "0000"+answer
        print(answer.upper())

def readFile():
    fp = open("3int_testdata.txt","r")
    for line in fp.readlines():
        line=line.strip()
        hexConvert(gridValues(line))
    readFile()