```java
//Anand Vinod
//Contest 3
//Junior Division
//For inputs and main method go to line 351

import java.lang.*;
import java.io.*;
import java.util.*;


public class AnandVC3Jr_Veitch2020 {
   static private ArrayList<Integer> used1  = new ArrayList<Integer>();
   static private ArrayList<Integer> used2  = new ArrayList<Integer>();
    public static String hexBinary(String hex){
        int numVal1 = 0;
        int numVal2 = 0;
        int finalVal = 0;
        String num1 = hex.substring(1,2);
        String num2 = hex.substring(0,1);
        int val1 = 0;
        int val2 = 0;
        if(num1.equals("A")){
            numVal1 = 10;
        }
        else if(num1.equals("B")){
            numVal1 = 11;
        }
        else if(num1.equals("C")){
            numVal1 = 12;
        }
        else if(num1.equals("D")){
            numVal1 = 13;
        }
        else if(num1.equals("E")){
            numVal1 = 14;
        }
        else if(num1.equals("F")){
            numVal1 = 15;
        }

        else{
            numVal1 = Integer.parseInt(num1);
        }

        if(num2.equals("A")){
            numVal2 = 10;
        }
        else if(num2.equals("B")){
            numVal2 = 11;
        }
        else if(num2.equals("C")){
            numVal2 = 12;
        }
        else if(num2.equals("D")){
            numVal2 = 13;
        }
        else if(num2.equals("E")){
            numVal2 = 14;
        }
        else if(num2.equals("F")){
            numVal2 = 15;
        }

        else{
            numVal2 = Integer.parseInt(num2);
        }


        return decToBinary(numVal2) + decToBinary(numVal1);
```

```
70      }
71      public static String decToBinary(int num) {
72
73          String binary ="";
74          int div = num;
75          for(int i = 0; i < 8 && div > 0; i++){
76
77              if(div%2==0)
78                  binary += 0;
79               else
80                   binary += 1;
81
82               div /= 2;
83          }
84
85          while(binary.length()<4)
86              binary +=("0");
87
88          return reverse(binary);
89      }
90
91          public static String reverse(String c){
92              String x = "";
93
94              for(int i = c.length(); i>0; i--)
95                  x+= c.substring(i-1,i);
96              return x;
97          }
98
99
100         public static String case1(String c)
101         {
102             used1.clear();
103             used2.clear();
104             String s1 = c.substring(0,4);
105             String s2 = c.substring(4,8);
106             String retVal = "";
107             String retVal2 = "";
108             String matchingCase = "";
109             if(!(s1.contains("0"))) {
110
111                 retVal += "B";
112                 used1.add(0);
113                 used1.add(1);
114                 used1.add(2);
115                 used1.add(3);
116             }
117             if(!(s2.contains("0"))) {
118
119                 retVal += "~B";
120                 used2.add(0);
121                 used2.add(1);
122                 used2.add(2);
123                 used2.add(3);
124             }
125             ArrayList<Integer> posUsed = new ArrayList<>();
126             for(int i = 0; i< s1.length(); i ++)
127             {
128                 if(s1.charAt(i) == s2.charAt(i) && s2.charAt(i) == '1') {
129                     matchingCase += i;
130                     posUsed.add(i);
131                 }
132                 if(posUsed.contains(i-1))
133                     break;
134             }
135                 if(matchingCase.length()>2)
136                 matchingCase = matchingCase.substring(0,2);
137                 if(matchingCase.length() > 1)
138                 {
```

```
                    if(matchingCase.equals("01")) {
                        retVal2 += "A";
                        used1.addAll(posUsed);
                        used2.addAll(posUsed);
                    }
                    else if(matchingCase.equals("12")) {
                        retVal2 += "C";
                        used1.addAll(posUsed);
                        used2.addAll(posUsed);
                    }
                    else if(matchingCase.equals("23")) {
                        retVal2 += "~A";
                        used1.addAll(posUsed);
                        used2.addAll(posUsed);
                    }
                }



                if(retVal.length() > 1 && retVal2.length()>1)
                    retVal += "+" + retVal2;
                else
                    retVal += retVal2;

            return retVal;
        }

        public static String case2(String c)
        {
            String s1 = c.substring(0,4);
            String s2 = c.substring(4,8);
           String a = case1(c);

            ArrayList<Integer> ad= new ArrayList<>();
            ad.add(0);
            ad.add(3);

           if(s1.charAt(0) == '1' && s1.charAt(3)=='1' && s2.charAt(0) == '1' &&
           s2.charAt(3)=='1' && a.length() > 0 && usedDoesntContain(ad) &&
           used2DoesntContain(ad))
           {
               a += "+~C";
               used1.addAll(ad);
               used2.addAll(ad);
           }
           else if(s1.charAt(0) == '1' && s1.charAt(3)=='1' && s2.charAt(0) == '1' &&
           s2.charAt(3)=='1'&& a.length() == 0 && usedDoesntContain(ad) &&
           used2DoesntContain(ad))
            {
                a += "~C";
                used1.addAll(ad);
                used2.addAll(ad);
            }

           return a;
        }

    public static String case3(String c){

       String a =  case2(c);
       String s1 = c.substring(0,4);
       String s2 = c.substring(4,8);

       HashMap<Integer, String> alp = new HashMap<>();
        alp.put(0, "AB");
        alp.put(1, "CB");
        alp.put(2, "~AB");

        HashMap<Integer, String> alp2 = new HashMap<>();
```

```
204          alp2.put(0, "A~B");
205          alp2.put(1, "C~B");
206          alp2.put(2, "~A~B");
207
208          HashMap<Integer, String> alp3 = new HashMap<>();
209          alp3.put(0, "A~C");
210          alp3.put(1, "AC");
211          alp3.put(2, "~AC");
212          alp3.put(3, "~A~C");
213
214         for(int i = 0; i < s1.length()-1; i ++)
215         {
216             ArrayList<Integer> as = new ArrayList<>();
217             as.add(i);
218             as.add(i+1);
219
220             if(s1.charAt(i) == s1.charAt(i+1)&&s1.charAt(i) == '1' &&
                usedDoesntContain(as))
221             {
222                 if(a.length() == 0)
223                     a += alp.get(i);
224                 else
225                     a += "+" + alp.get(i);
226                 used1.add(i);
227                 used1.add(i+1);
228             }
229
230         }
231
232          for(int i = 0; i < s2.length()-1; i ++)
233          {
234              ArrayList<Integer> as = new ArrayList<>();
235              as.add(i);
236              as.add(i+1);
237              if(s2.charAt(i) == s2.charAt(i+1) && s2.charAt(i) == '1' &&
                 used2DoesntContain(as))
238              {
239                  if(a.length() == 0)
240                      a += alp2.get(i);
241                  else
242                      a += "+" + alp2.get(i);
243                  used2.add(i);
244                  used2.add(i+1);
245              }
246
247          }
248
249
250          for(int i = 0; i < s2.length(); i++)
251          {
252              if(s2.charAt(i) == s1.charAt(i) && s2.charAt(i) == '1' &&
                 !(used2.contains(i) || used1.contains(i)))
253              {
254                  if(a.length() == 0)
255                      a += alp3.get(i);
256                  else
257                      a+= "+" + alp3.get(i);
258
259                  used1.add(i);
260                  used2.add(i);
261              }
262          }
263          return a;
264      }
265          public static String case4(String c){
266              String a =  case3(c);
267              String s1 = c.substring(0,4);
268              String s2 = c.substring(4,8);
269              ArrayList<Integer> ft= new ArrayList<>();
```

```
270             ft.add(0);
271             ft.add(3);
272             if(s1.charAt(0) == s1.charAt(3) && s1.charAt(0) == '1' &&
                usedDoesntContain(ft))
273             {
274                 used1.addAll(ft);
275                 if(a.length() == 0)
276                     a += "B~C";
277                 else
278                     a += "+B~C";
279             }
280
281             if(s2.charAt(0) == s2.charAt(3) && s2.charAt(0) == '1' &&
                used2DoesntContain(ft))
282             {
283                 used2.addAll(ft);
284                 if(a.length() == 0)
285                     a += "~B~C";
286                 else
287                     a += "+~B~C";
288             }
289             return a;
290         }
291
292         public static String case5(String c)
293         {
294             String s1 = c.substring(0,4);
295             String s2 = c.substring(4,8);
296             String a = case4(c);
297             HashMap<Integer, String> g1 = new HashMap<>();
298             g1.put(0 , "AB~C");
299             g1.put(1 , "ABC");
300             g1.put(2 , "~ABC");
301             g1.put(3 , "~AB~C");
302             HashMap<Integer, String> g2 = new HashMap<>();
303             g2.put(0 , "A~B~C");
304             g2.put(1 , "A~BC");
305             g2.put(2 , "~A~BC");
306             g2.put(3 , "~A~B~C");
307
308             for(int i = 0; i < s1.length(); i++)
309             {
310
311                 if(!used1.contains(i) && s1.charAt(i) == '1')
312                 {
313
314                     if(a.length()>0)
315                         a += "+" + g1.get(i);
316                     else
317                         a += g1.get(i);
318                 }
319
320                 if(!used2.contains(i) && s2.charAt(i) == '1')
321                 {
322
323                     if(a.length()>0)
324                         a += "+" + g2.get(i);
325                     else
326                         a += g2.get(i);
327                 }
328             }
329             return a;
330         }
331
332         public static boolean usedDoesntContain(ArrayList<Integer> ft)
333         {
334             for(int i = 0; i<ft.size(); i++)
335             {
336                 if(used1.contains(ft.get(i)))
```

```
                return false;
            }
            return true;
        }
    public static boolean used2DoesntContain(ArrayList<Integer> ft)
    {
        for(int i = 0; i<ft.size(); i++)
        {
            if(used2.contains(ft.get(i)))
            return false;
        }
        return true;
    }



        public static void main(String[]Args) throws IOException
        {
            File file = new File("C:\\Users\\Anand Vinod\\Desktop\\data.txt");
            Scanner scan = new Scanner(file);

            while(scan.hasNextLine())
            {
                String input = scan.nextLine();

                System.out.println(case5(hexBinary(input)));

                used1.clear();
                used2.clear();
            }
        }
}
```