```
/*
COVACI ANDREI GABRIEL
C.N. "Emanuil Gojdu" Oradea
prof. Maria NITA, prof. Adrian NITA
ACSL - Contest 3
Junior Division
*/

#include <iostream>
#include <fstream>
#include <cstring>

using namespace std;

ifstream in("veitch.in");
ofstream out("veitch.out");

bool v[2][4];

void complete(int row, bool a, bool b, bool c, bool d){
   if(row == 0){
      v[0][0] = a;
      v[0][1] = b;
      v[0][2] = c;
      v[0][3] = d;
   }else{
      v[1][0] = a;
      v[1][1] = b;
      v[1][2] = c;
      v[1][3] = d;
   }
}
```

```
void convertToBinary(int row, char a){
   switch(a){
      case '0':
         complete(row, false, false, false, false);
         break;
      case '1':
         complete(row, false, false, false, true);
         break;
      case '2':
         complete(row, false, false, true, false);
         break;
      case '3':
         complete(row, false, false, true, true);
         break;
      case '4':
         complete(row, false, true, false, false);
         break;
      case '5':
         complete(row, false, true, false, true);
         break;
      case '6':
         complete(row, false, true, true, false);
         break;
      case '7':
         complete(row, false, true, true, true);
         break;
      case '8':
         complete(row, true, false, false, false);
         break;
      case '9':
         complete(row, true, false, false, true);
         break;
      case 'A':
```

```
            complete(row, true, false, true, false);
            break;
        case 'B':
            complete(row, true, false, true, true);
            break;
        case 'C':
            complete(row, true, true, false, false);
            break;
        case 'D':
            complete(row, true, true, false, true);
            break;
        case 'E':
            complete(row, true, true, true, false);
            break;
        case 'F':
            complete(row, true, true, true, true);
            break;
    }
}

void showMatrix(){
    out << '\n' << "Current state of matrix : \n";
    for(int i = 0; i < 2; i++){
        for(int j = 0; j < 4; j++){
            out << v[i][j] << " ";
        }
        out << '\n';
    }
    out << '\n';
}

/*bool ifEmpty(){
    bool emty = false;
```

```
   for(int i = 0; i < 2 && !emty; i++){
      for(int j = 0; j < 4 && !emty; j++){
         emty = (v[i][j]) ? false : true;
      }
   }
   return emty;
}*/

int main()
{
   char a[2], answer[100];

   for(int o = 0; o < 5; o++){
      strcpy(answer, "");

      //read input

      in >> a;
      //out << "Hex number input : " << a << "\n";

      //convert to binary and complete the matrix

      convertToBinary(0, a[0]);
      convertToBinary(1, a[1]);

      //showMatrix();

      //check for groups of 4 Xs

      /*
      1 1 1 1
      0 0 0 0
      */
```

```
if(v[0][0] && v[0][1] && v[0][2] && v[0][3]){
   out << "B";
   strcat(answer, "B+");
   v[0][0] = false;
   v[0][1] = false;
   v[0][2] = false;
   v[0][3] = false;
   //showMatrix();
}

/*
0 0 0 0
1 1 1 1
*/
if(v[1][0] && v[1][1] && v[1][2] && v[1][3]){
   //out << "~B";
   strcat(answer, "~B+");
   v[1][0] = false;
   v[1][1] = false;
   v[1][2] = false;
   v[1][3] = false;
   //showMatrix();
}

/*
1 1 0 0
1 1 0 0
*/
if(v[0][0] && v[0][1] && v[1][0] && v[1][1]){
   //out << "A";
   strcat(answer, "A+");
   v[0][0] = false;
   v[0][1] = false;
```

```
    v[1][0] = false;
    v[1][1] = false;
    //showMatrix();
}

/*
0 1 1 0
0 1 1 0
*/
if(v[0][1] && v[0][2] && v[1][1] && v[1][2]){
    //out << "C";
    strcat(answer, "C+");
    v[0][1] = false;
    v[0][2] = false;
    v[1][1] = false;
    v[1][2] = false;
    //showMatrix();
}

/*
0 0 1 1
0 0 1 1
*/
if(v[0][2] && v[0][3] && v[1][2] && v[1][3]){
    //out << "~A";
    strcat(answer, "~A+");
    v[0][2] = false;
    v[0][3] = false;
    v[1][2] = false;
    v[1][3] = false;
    //showMatrix();
}
```

```
/*
1 0 0 1
1 0 0 1
*/
if(v[0][0] && v[0][3] && v[1][0] && v[1][3]){
   //out << "~C";
   strcat(answer, "~C+");
   v[0][0] = false;
   v[0][3] = false;
   v[1][0] = false;
   v[1][3] = false;
   //showMatrix();
}

//check for 2 adjacent Xs

//for horizontal groups
for(int i = 0; i < 2; i++){
   for(int j = 0; j < 3; j++){
      if(v[i][j] && v[i][j + 1]){
         if(j == 0){
            //out << "A";
            strcat(answer, "A");
         }else if(j == 1){
            //out << "C";
            strcat(answer, "C");
         }else if(j == 2){
            //out << "~A";
            strcat(answer, "~A");
         }

         if(i == 0){
            //out << "B";
```

```
            strcat(answer, "B");
        }else{
            //out << "~B";
            strcat(answer, "~B");
        }
        strcat(answer, "+");
        v[i][j] = false;
        v[i][j + 1] = false;
        //showMatrix();
    }
  }
}

//for vertical groups
for(int i = 0; i < 4; i++){
  if(v[0][i] && v[1][i]){
    if(i == 0){
        //out << "A~C";
        strcat(answer, "A~C");
    }else if(i == 1){
        //out << "AC";
        strcat(answer, "AC");
    }else if(i == 2){
        //out << "~AC";
        strcat(answer, "~AC");
    }else if(i == 3){
        //out << "~A~C";
        strcat(answer, "~A~C");
    }

    strcat(answer, "+");
    v[0][i] = false;
    v[1][i] = false;
```

```
        //showMatrix();
    }
}

/*
1 0 0 1
0 0 0 0
*/
if(v[0][0] && v[0][3]){
    //out << "B~C";
    strcat(answer, "B~C+");
    v[0][0] = false;
    v[0][3] = false;
    //showMatrix();
}

/*
0 0 0 0
1 0 0 1
*/
if(v[1][0] && v[1][3]){
    //out << "~B~C";
    strcat(answer, "~B~C+");
    v[1][0] = false;
    v[1][3] = false;
    //showMatrix();
}

//check for single Xs

if(v[0][0]){
    //out << "AB~C";
    strcat(answer, "AB~C+");
```

```
    v[0][0] = false;
    //showMatrix();
}
if(v[0][1]){
    //out << "ABC";
    strcat(answer, "ABC+");
    v[0][1] = false;
    //showMatrix();
}
if(v[0][2]){
    //out << "~ABC";
    strcat(answer, "~ABC+");
    v[0][2] = false;
    //showMatrix();
}
if(v[0][3]){
    //out << "~AB~C";
    strcat(answer, "~AB~C+");
    v[0][3] = false;
    //showMatrix();
}
if(v[1][0]){
    //out << "A~B~C";
    strcat(answer, "A~B~C+");
    v[1][0] = false;
    //showMatrix();
}
if(v[1][1]){
    //out << "A~BC";
    strcat(answer, "A~BC+");
    v[1][1] = false;
    //showMatrix();
}
```

```
        if(v[1][2]){
            //out << "~A~BC";
            strcat(answer, "~A~BC+");
            v[1][2] = false;
            //showMatrix();
        }
        if(v[1][3]){
            //out << "~A~B~C";
            strcat(answer, "~A~B~C+");
            v[1][3] = false;
            //showMatrix();
        }

        char a[100];
        strcpy(a, answer + strlen(answer));
        strcpy(answer + strlen(answer) - 1, a);

        //out << '\n' << answer << "\n----------\n";
        out << answer << '\n';
    }
    return 0;
}
```