

```
/*
```

```
GHERGHELES MIHAI GEORGE
```

```
C.N. "Emanuil Gojdu" Oradea
```

```
prof. Maria NITA, prof. Adrian NITA
```

```
ACSL - Contest 3
```

```
Junior Division
```

```
*/
```

```
#include <fstream>
```

```
#include <cstring>
```

```
using namespace std;
```

```
ifstream in("veitch.in");
```

```
ofstream out("acsl3.out");
```

```
bool diagram[2][4];
```

```
void complete(int row, bool a, bool b, bool c, bool d){
```

```
    diagram[row][0] = a;
```

```
    diagram[row][1] = b;
```

```
    diagram[row][2] = c;
```

```
    diagram[row][3] = d;
```

```
}
```

```
void hexaToBinary(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;
    }
}

int main() {
    char cif[10], rez[100];
    strcpy(rez, "");
    for(int q = 0; q < 5; q++)
    {
        in >> cif;

        hexaToBinary(0, cif[0]);
        hexaToBinary(1, cif[1]);

        if(diagram[0][0] && diagram[0][1] && diagram[0][2] && diagram[0][3]){
            strcat(rez, "B+");
            diagram[0][0] = false;
            diagram[0][1] = false;
            diagram[0][2] = false;
            diagram[0][3] = false;
        }

        if(diagram[1][0] && diagram[1][1] && diagram[1][2] && diagram[1][3]){
            strcat(rez, "~B+");
            diagram[1][0] = false;

```

```
    diagram[1][1] = false;
    diagram[1][2] = false;
    diagram[1][3] = false;
}
```

```
if(diagram[0][0] && diagram[0][1] && diagram[1][0] && diagram[1][1]){
    strcat(rez, "A+");
    diagram[0][0] = false;
    diagram[0][1] = false;
    diagram[1][0] = false;
    diagram[1][1] = false;
}
```

```
if(diagram[0][1] && diagram[0][2] && diagram[1][1] && diagram[1][2]){
    strcat(rez, "C+");
    diagram[0][1] = false;
    diagram[0][2] = false;
    diagram[1][1] = false;
    diagram[1][2] = false;
}
```

```
if(diagram[0][2] && diagram[0][3] && diagram[1][2] && diagram[1][3]){
    strcat(rez, "~A+");
    diagram[0][2] = false;
    diagram[0][3] = false;
    diagram[1][2] = false;
    diagram[1][3] = false;
}
```

```
if(diagram[0][0] && diagram[1][0] && diagram[0][3] && diagram[1][3]){
    strcat(rez, "~C+");
    diagram[0][0] = false;
    diagram[0][3] = false;
}
```

```

    diagram[1][0] = false;
    diagram[1][3] = false;
}

for(int i = 0; i < 2; i++){
    for(int j = 0; j < 3; j++){
        if(diagram[i][j] && diagram[i][j + 1]){
            if(i == 0){
                if(j == 0){
                    strcat(rez, "AB+");
                }else if(j == 1){
                    strcat(rez, "B~C+");
                }else if(j == 2){
                    strcat(rez, "~AB+");
                }
            }
            if(i == 1){
                if(j == 0){
                    strcat(rez, "A~B+");
                }else if(j == 1){
                    strcat(rez, "~B~C+");
                }else if(j == 2){
                    strcat(rez, "~A~B+");
                }
            }
        }
        diagram[i][j] = false;
        diagram[i][j + 1] = false;
    }
}

```

```

for(int j = 0; j < 4; j++){
    if(diagram[0][j] && diagram[1][j]){
        if(j == 0){
            strcat(rez, "AC+");
        }else if(j == 1){
            strcat(rez, "A~C+");
        }else if(j == 2){
            strcat(rez, "~A~C+");
        }else if(j == 3){
            strcat(rez, "~A~C+");
        }
        diagram[0][j] = false;
        diagram[1][j] = false;
    }
}

if(diagram[0][0] && diagram[0][3]){
    strcat(rez, "B~C+");
    diagram[0][0] = false;
    diagram[0][3] = false;
}

if(diagram[1][0] && diagram[1][3]){
    strcat(rez, "~B~C+");
    diagram[1][0] = false;
    diagram[1][3] = false;
}

if(diagram[0][0]){
    strcat(rez, "AB~C+");
    diagram[0][0] = false;
}

if(diagram[0][1]){
    strcat(rez, "ABC+");
    diagram[0][1] = false;
}

```

```

    }
    if(diagram[0][2]){
strcat(rez, "~ABC+");
diagram[0][2] = false;
    }
    if(diagram[0][3]){
strcat(rez, "~AB~C+");
diagram[0][3] = false;
    }
    if(diagram[1][0]){
strcat(rez, "A~B~C+");
diagram[1][0] = false;
    }
    if(diagram[1][1]){
strcat(rez, "A~BC+");
diagram[1][1] = false;
    }
    if(diagram[1][2]){
strcat(rez, "~A~BC+");
diagram[1][2] = false;
    }
    if(diagram[1][3]){
strcat(rez, "~A~B~C+");
diagram[1][3] = false;
    }
    char b[100];
    strcpy(b, rez + strlen(rez));
    strcpy(rez + strlen(rez) - 1, b);
    out << rez << '\n';
    strcpy(rez, "");
}
return 0;
}

```