1. Define a graph $G = (V, E)$ as follows.
   Let $V = \{1, 2, 3, \ldots \quad 12 \quad \}$. Define $E = \{(i, j) : i, j \in V, i \neq j, i^2 + j^2 + ij + 4$ is prime $\}$.
   Create and solve (using lpsolve) an IP to find the chromatic number of G, $\chi(G)$.

First, let's find all the edges defined in the question.
I used `isprime(n)` function in MATLAB to check prime number.

The following is the code I wrote in MATLAB:

```
--------------------------------------------------------------
%% HW3 for MATH381
% @Author: Baihan Lin, @Date: Oct 2016

clear all; close all; clc;

% initialization
max = 12;
V = [1:max];
E = [];

% get edges and the corresponding graph appears
for j = 1:max
    for k = j+1:max
        if (isprime(j^2+k^2+j*k+4))
            E = [E;[j k]];
        end
    end
end

% show edges
E
--------------------------------------------------------------
```

Here is the output from MATLAB console:

```
--------------------------------------------------------------
E =
     1     2
     1     3
     1     6
     1     7
     1    11
     2     3
     2     5
     2     7
     2     9
     2    11
     3     4
     3     5
     3     6
     3     7
     3     8
     3    11
     3    12
     4     7
     4     9
     5     7
     5    10
```

```
     5     12
     6      7
     6     11
     7      8
     7      9
     7     10
     7     11
     7     12
     8     11
     9     12
    11     12
--------------------------------------------------------------
```

Then we do the IP formulation:

Now we have this graph $G = (V, E)$ with 12 vertices.
Suppose we have a set of n colors to use to color the graph (we might not use all of them).

Define $y_i \in \{0, 1\}$ $i = 1, \ldots, 12$, with $y_i = 1$ iff we use color i.
Define $x_{ik} \in \{0, 1\}$ $i = 1, \ldots, 12$, with $x_{ik} = 1$ iff vertex i will have color k

Then our IP is this:

Minimize $\sum_{i=1}^{n} y_i$ subject to

1) $\sum_{i=1}^{n} x_{ik} = 1, i = 1, \ldots, n$
2) $x_{ik} \leq y_k, i, k = 1, \ldots, n$
3) $x_{ik} + x_{jk} \leq 1$ for all $(v_i, v_j) \in E \; and \; k = 1, \ldots, n$
4) $x_{ik}, y_k \epsilon \{0, 1\}$

In order to generate necessary input for lp_solve, under the inspiration of Dr. Conroy's demo code on Knights' Domination Problem, I wrote following code in Perl (lpGenerator.pl).

Note: Here I used another way to check prime in this Perl code: for each $i^2 + j^2 + ij + 4$, I divide it by all integers smaller than itself starting from 2, and check whether there is any divisible factor.

```
--------------------------------------------------------------
#!/usr/local/bin/perl

# Baihan Lin, Oct 2016
#
# A perl script to generate an lpsolve-format IP (integer program)

$max=12;

## print out objective function
print "min:";
                    for($i=1; $i<$max+1; $i++) {
                         print "+y_",$i;
                    }
print ";\n";

## constraint 1
for($j=1; $j<$max+1; $j++) {
                    for($i=1; $i<$max+1; $i++) {
```

```
                    print "+x_",$j,"_",$i;
                }
                print "=1;\n"
}

## constraint 2
for($j=1; $j<$max+1; $j++) {
                for($i=1; $i<$max+1; $i++) {
                    print "x_",$j,"_",$i," -", "y_",$i,"<=0;\n";
                }
}

## constraint 3
for($j=1; $j<$max+1; $j++) {
                for($i=$j+1; $i<$max+1; $i++) {

                    # prime number check
                    $n=$j*$j+$i*$i+$i*$j+4;
                    $d=0; # 0 is prime, 1 is not
                    for($c=2;$c<$n;$c++) {
                           if($n%$c==0) {
                                  $d=1;
                                  break;
                           }
                    }
                    if($d==0){
                           for($k=1; $k<$max+1; $k++) {
                                  print           "x_",$j,"_",$k,"           +",
"x_",$i,"_",$k,"<=1;\n";
                           }
                    }
                }
}

## constraint 4
## specify that all variables are binary
print "bin ";
for($i=1; $i<$max+1; $i++) {
                if ($i>1) { print ","; } ## prepend a comma first
                print "y_",$i;
}
for($j=1; $j<$max+1; $j++) {
                for($i=1; $i<$max+1; $i++) {
                    print ",x_",$j,"_",$i;
                }
}
print ";\n";
--------------------------------------------------------------
```

Here is the output from this perl code I stored into lp3.txt:
Note: the output is realigned to be multiple column in Microsoft Word to avoid too many pages.

```
--------------------------------------------------------------
min:+y_1+y_2+y_3+y_4+y_5+y_6+y_7+y_8+y_9+y_10+y_11+y_12;
+x_1_1+x_1_2+x_1_3+x_1_4+x_1_5+x_1_6+x_1_7+x_1_8+x_1_9+x_1_10+x_1_11+x_1_12=1;
+x_2_1+x_2_2+x_2_3+x_2_4+x_2_5+x_2_6+x_2_7+x_2_8+x_2_9+x_2_10+x_2_11+x_2_12=1;
+x_3_1+x_3_2+x_3_3+x_3_4+x_3_5+x_3_6+x_3_7+x_3_8+x_3_9+x_3_10+x_3_11+x_3_12=1;
+x_4_1+x_4_2+x_4_3+x_4_4+x_4_5+x_4_6+x_4_7+x_4_8+x_4_9+x_4_10+x_4_11+x_4_12=1;
+x_5_1+x_5_2+x_5_3+x_5_4+x_5_5+x_5_6+x_5_7+x_5_8+x_5_9+x_5_10+x_5_11+x_5_12=1;
+x_6_1+x_6_2+x_6_3+x_6_4+x_6_5+x_6_6+x_6_7+x_6_8+x_6_9+x_6_10+x_6_11+x_6_12=1;
+x_7_1+x_7_2+x_7_3+x_7_4+x_7_5+x_7_6+x_7_7+x_7_8+x_7_9+x_7_10+x_7_11+x_7_12=1;
+x_8_1+x_8_2+x_8_3+x_8_4+x_8_5+x_8_6+x_8_7+x_8_8+x_8_9+x_8_10+x_8_11+x_8_12=1;
+x_9_1+x_9_2+x_9_3+x_9_4+x_9_5+x_9_6+x_9_7+x_9_8+x_9_9+x_9_10+x_9_11+x_9_12=1;
+x_10_1+x_10_2+x_10_3+x_10_4+x_10_5+x_10_6+x_10_7+x_10_8+x_10_9+x_10_10+x_10_11+x_10_12=1;
+x_11_1+x_11_2+x_11_3+x_11_4+x_11_5+x_11_6+x_11_7+x_11_8+x_11_9+x_11_10+x_11_11+x_11_12=1;
```

```
+x_12_1+x_12_2+x_12_3+x_12_4+x_12_5+x_12_6+x_12_7+x_12_8+x_12_9+x_12_10+x_12_11+x_12_12=1;
x_1_1 -y_1<=0;
x_1_2 -y_2<=0;
x_1_3 -y_3<=0;
x_1_4 -y_4<=0;
x_1_5 -y_5<=0;
x_1_6 -y_6<=0;
x_1_7 -y_7<=0;
x_1_8 -y_8<=0;
x_1_9 -y_9<=0;
x_1_10 -y_10<=0;
x_1_11 -y_11<=0;
x_1_12 -y_12<=0;
x_2_1 -y_1<=0;
x_2_2 -y_2<=0;
x_2_3 -y_3<=0;
x_2_4 -y_4<=0;
x_2_5 -y_5<=0;
x_2_6 -y_6<=0;
x_2_7 -y_7<=0;
x_2_8 -y_8<=0;
x_2_9 -y_9<=0;
x_2_10 -y_10<=0;
x_2_11 -y_11<=0;
x_2_12 -y_12<=0;
x_3_1 -y_1<=0;
x_3_2 -y_2<=0;
x_3_3 -y_3<=0;
x_3_4 -y_4<=0;
x_3_5 -y_5<=0;
x_3_6 -y_6<=0;
x_3_7 -y_7<=0;
x_3_8 -y_8<=0;
x_3_9 -y_9<=0;
x_3_10 -y_10<=0;
x_3_11 -y_11<=0;
x_3_12 -y_12<=0;
x_4_1 -y_1<=0;
x_4_2 -y_2<=0;
x_4_3 -y_3<=0;
x_4_4 -y_4<=0;
x_4_5 -y_5<=0;
x_4_6 -y_6<=0;
x_4_7 -y_7<=0;
x_4_8 -y_8<=0;
x_4_9 -y_9<=0;
x_4_10 -y_10<=0;
x_4_11 -y_11<=0;
x_4_12 -y_12<=0;
x_5_1 -y_1<=0;
x_5_2 -y_2<=0;
x_5_3 -y_3<=0;
x_5_4 -y_4<=0;
x_5_5 -y_5<=0;
x_5_6 -y_6<=0;
x_5_7 -y_7<=0;
x_5_8 -y_8<=0;
x_5_9 -y_9<=0;
x_5_10 -y_10<=0;
x_5_11 -y_11<=0;
x_5_12 -y_12<=0;
x_6_1 -y_1<=0;
x_6_2 -y_2<=0;
x_6_3 -y_3<=0;
x_6_4 -y_4<=0;
x_6_5 -y_5<=0;
x_6_6 -y_6<=0;
x_6_7 -y_7<=0;
x_6_8 -y_8<=0;
x_6_9 -y_9<=0;
x_6_10 -y_10<=0;
x_6_11 -y_11<=0;
x_6_12 -y_12<=0;
x_7_1 -y_1<=0;
x_7_2 -y_2<=0;
x_7_3 -y_3<=0;
x_7_4 -y_4<=0;
x_7_5 -y_5<=0;
x_7_6 -y_6<=0;
x_7_7 -y_7<=0;
x_7_8 -y_8<=0;
x_7_9 -y_9<=0;
x_7_10 -y_10<=0;
x_7_11 -y_11<=0;
x_7_12 -y_12<=0;
x_8_1 -y_1<=0;
x_8_2 -y_2<=0;
x_8_3 -y_3<=0;
x_8_4 -y_4<=0;
x_8_5 -y_5<=0;
x_8_6 -y_6<=0;
x_8_7 -y_7<=0;
x_8_8 -y_8<=0;
x_8_9 -y_9<=0;
x_8_10 -y_10<=0;
x_8_11 -y_11<=0;
x_8_12 -y_12<=0;
x_9_1 -y_1<=0;
x_9_2 -y_2<=0;
x_9_3 -y_3<=0;
x_9_4 -y_4<=0;
x_9_5 -y_5<=0;
x_9_6 -y_6<=0;
x_9_7 -y_7<=0;
x_9_8 -y_8<=0;
x_9_9 -y_9<=0;
x_9_10 -y_10<=0;
x_9_11 -y_11<=0;
x_9_12 -y_12<=0;
x_10_1 -y_1<=0;
x_10_2 -y_2<=0;
x_10_3 -y_3<=0;
x_10_4 -y_4<=0;
x_10_5 -y_5<=0;
x_10_6 -y_6<=0;
x_10_7 -y_7<=0;
x_10_8 -y_8<=0;
x_10_9 -y_9<=0;
x_10_10 -y_10<=0;
x_10_11 -y_11<=0;
x_10_12 -y_12<=0;
x_11_1 -y_1<=0;
x_11_2 -y_2<=0;
x_11_3 -y_3<=0;
x_11_4 -y_4<=0;
x_11_5 -y_5<=0;
x_11_6 -y_6<=0;
x_11_7 -y_7<=0;
x_11_8 -y_8<=0;
x_11_9 -y_9<=0;
x_11_10 -y_10<=0;
x_11_11 -y_11<=0;
x_11_12 -y_12<=0;
x_12_1 -y_1<=0;
x_12_2 -y_2<=0;
x_12_3 -y_3<=0;
x_12_4 -y_4<=0;
x_12_5 -y_5<=0;
x_12_6 -y_6<=0;
x_12_7 -y_7<=0;
x_12_8 -y_8<=0;
x_12_9 -y_9<=0;
x_12_10 -y_10<=0;
x_12_11 -y_11<=0;
x_12_12 -y_12<=0;
x_1_1 +x_2_1<=1;
x_1_2 +x_2_2<=1;
x_1_3 +x_2_3<=1;
x_1_4 +x_2_4<=1;
x_1_5 +x_2_5<=1;
x_1_6 +x_2_6<=1;
x_1_7 +x_2_7<=1;
x_1_8 +x_2_8<=1;
x_1_9 +x_2_9<=1;
x_1_10 +x_2_10<=1;
x_1_11 +x_2_11<=1;
x_1_12 +x_2_12<=1;
x_1_1 +x_3_1<=1;
x_1_2 +x_3_2<=1;
x_1_3 +x_3_3<=1;
x_1_4 +x_3_4<=1;
x_1_5 +x_3_5<=1;
x_1_6 +x_3_6<=1;
x_1_7 +x_3_7<=1;
x_1_8 +x_3_8<=1;
x_1_9 +x_3_9<=1;
x_1_10 +x_3_10<=1;
x_1_11 +x_3_11<=1;
x_1_12 +x_3_12<=1;
x_1_1 +x_6_1<=1;
x_1_2 +x_6_2<=1;
x_1_3 +x_6_3<=1;
x_1_4 +x_6_4<=1;
x_1_5 +x_6_5<=1;
x_1_6 +x_6_6<=1;
x_1_7 +x_6_7<=1;
x_1_8 +x_6_8<=1;
x_1_9 +x_6_9<=1;
x_1_10 +x_6_10<=1;
x_1_11 +x_6_11<=1;
x_1_12 +x_6_12<=1;
x_1_1 +x_7_1<=1;
x_1_2 +x_7_2<=1;
x_1_3 +x_7_3<=1;
x_1_4 +x_7_4<=1;
x_1_5 +x_7_5<=1;
x_1_6 +x_7_6<=1;
x_1_7 +x_7_7<=1;
x_1_8 +x_7_8<=1;
x_1_9 +x_7_9<=1;
x_1_10 +x_7_10<=1;
x_1_11 +x_7_11<=1;
x_1_12 +x_7_12<=1;
x_1_1 +x_11_1<=1;
x_1_2 +x_11_2<=1;
x_1_3 +x_11_3<=1;
x_1_4 +x_11_4<=1;
x_1_5 +x_11_5<=1;
x_1_6 +x_11_6<=1;
x_1_7 +x_11_7<=1;
x_1_8 +x_11_8<=1;
x_1_9 +x_11_9<=1;
x_1_10 +x_11_10<=1;
x_1_11 +x_11_11<=1;
x_1_12 +x_11_12<=1;
x_2_1 +x_3_1<=1;
x_2_2 +x_3_2<=1;
x_2_3 +x_3_3<=1;
x_2_4 +x_3_4<=1;
x_2_5 +x_3_5<=1;
x_2_6 +x_3_6<=1;
x_2_7 +x_3_7<=1;
x_2_8 +x_3_8<=1;
x_2_9 +x_3_9<=1;
x_2_10 +x_3_10<=1;
x_2_11 +x_3_11<=1;
x_2_12 +x_3_12<=1;
x_2_1 +x_5_1<=1;
x_2_2 +x_5_2<=1;
x_2_3 +x_5_3<=1;
x_2_4 +x_5_4<=1;
x_2_5 +x_5_5<=1;
x_2_6 +x_5_6<=1;
x_2_7 +x_5_7<=1;
x_2_8 +x_5_8<=1;
x_2_9 +x_5_9<=1;
x_2_10 +x_5_10<=1;
x_2_11 +x_5_11<=1;
x_2_12 +x_5_12<=1;
x_2_1 +x_7_1<=1;
x_2_2 +x_7_2<=1;
x_2_3 +x_7_3<=1;
x_2_4 +x_7_4<=1;
x_2_5 +x_7_5<=1;
x_2_6 +x_7_6<=1;
x_2_7 +x_7_7<=1;
x_2_8 +x_7_8<=1;
x_2_9 +x_7_9<=1;
x_2_10 +x_7_10<=1;
x_2_11 +x_7_11<=1;
x_2_12 +x_7_12<=1;
x_2_1 +x_9_1<=1;
x_2_2 +x_9_2<=1;
x_2_3 +x_9_3<=1;
x_2_4 +x_9_4<=1;
x_2_5 +x_9_5<=1;
x_2_6 +x_9_6<=1;
x_2_7 +x_9_7<=1;
x_2_8 +x_9_8<=1;
x_2_9 +x_9_9<=1;
x_2_10 +x_9_10<=1;
x_2_11 +x_9_11<=1;
x_2_12 +x_9_12<=1;
x_2_1 +x_11_1<=1;
x_2_2 +x_11_2<=1;
x_2_3 +x_11_3<=1;
x_2_4 +x_11_4<=1;
x_2_5 +x_11_5<=1;
x_2_6 +x_11_6<=1;
x_2_7 +x_11_7<=1;
x_2_8 +x_11_8<=1;
x_2_9 +x_11_9<=1;
x_2_10 +x_11_10<=1;
x_2_11 +x_11_11<=1;
x_2_12 +x_11_12<=1;
x_3_1 +x_4_1<=1;
x_3_2 +x_4_2<=1;
x_3_3 +x_4_3<=1;
x_3_4 +x_4_4<=1;
x_3_5 +x_4_5<=1;
x_3_6 +x_4_6<=1;
x_3_7 +x_4_7<=1;
x_3_8 +x_4_8<=1;
x_3_9 +x_4_9<=1;
x_3_10 +x_4_10<=1;
x_3_11 +x_4_11<=1;
x_3_12 +x_4_12<=1;
x_3_1 +x_5_1<=1;
x_3_2 +x_5_2<=1;
x_3_3 +x_5_3<=1;
x_3_4 +x_5_4<=1;
x_3_5 +x_5_5<=1;
x_3_6 +x_5_6<=1;
x_3_7 +x_5_7<=1;
x_3_8 +x_5_8<=1;
x_3_9 +x_5_9<=1;
x_3_10 +x_5_10<=1;
x_3_11 +x_5_11<=1;
x_3_12 +x_5_12<=1;
```

```
x_3_1 +x_6_1<=1;
x_3_2 +x_6_2<=1;
x_3_3 +x_6_3<=1;
x_3_4 +x_6_4<=1;
x_3_5 +x_6_5<=1;
x_3_6 +x_6_6<=1;
x_3_7 +x_6_7<=1;
x_3_8 +x_6_8<=1;
x_3_9 +x_6_9<=1;
x_3_10 +x_6_10<=1;
x_3_11 +x_6_11<=1;
x_3_12 +x_6_12<=1;
x_3_1 +x_7_1<=1;
x_3_2 +x_7_2<=1;
x_3_3 +x_7_3<=1;
x_3_4 +x_7_4<=1;
x_3_5 +x_7_5<=1;
x_3_6 +x_7_6<=1;
x_3_7 +x_7_7<=1;
x_3_8 +x_7_8<=1;
x_3_9 +x_7_9<=1;
x_3_10 +x_7_10<=1;
x_3_11 +x_7_11<=1;
x_3_12 +x_7_12<=1;
x_3_1 +x_8_1<=1;
x_3_2 +x_8_2<=1;
x_3_3 +x_8_3<=1;
x_3_4 +x_8_4<=1;
x_3_5 +x_8_5<=1;
x_3_6 +x_8_6<=1;
x_3_7 +x_8_7<=1;
x_3_8 +x_8_8<=1;
x_3_9 +x_8_9<=1;
x_3_10 +x_8_10<=1;
x_3_11 +x_8_11<=1;
x_3_12 +x_8_12<=1;
x_3_1 +x_11_1<=1;
x_3_2 +x_11_2<=1;
x_3_3 +x_11_3<=1;
x_3_4 +x_11_4<=1;
x_3_5 +x_11_5<=1;
x_3_6 +x_11_6<=1;
x_3_7 +x_11_7<=1;
x_3_8 +x_11_8<=1;
x_3_9 +x_11_9<=1;
x_3_10 +x_11_10<=1;
x_3_11 +x_11_11<=1;
x_3_12 +x_11_12<=1;
x_3_1 +x_12_1<=1;
x_3_2 +x_12_2<=1;
x_3_3 +x_12_3<=1;
x_3_4 +x_12_4<=1;
x_3_5 +x_12_5<=1;
x_3_6 +x_12_6<=1;
x_3_7 +x_12_7<=1;
x_3_8 +x_12_8<=1;
x_3_9 +x_12_9<=1;
x_3_10 +x_12_10<=1;
x_3_11 +x_12_11<=1;
x_3_12 +x_12_12<=1;
x_4_1 +x_7_1<=1;
x_4_2 +x_7_2<=1;
x_4_3 +x_7_3<=1;
x_4_4 +x_7_4<=1;
x_4_5 +x_7_5<=1;
x_4_6 +x_7_6<=1;
x_4_7 +x_7_7<=1;
x_4_8 +x_7_8<=1;
x_4_9 +x_7_9<=1;
x_4_10 +x_7_10<=1;
x_4_11 +x_7_11<=1;
x_4_12 +x_7_12<=1;
x_4_1 +x_9_1<=1;
x_4_2 +x_9_2<=1;
x_4_3 +x_9_3<=1;
x_4_4 +x_9_4<=1;
x_4_5 +x_9_5<=1;
x_4_6 +x_9_6<=1;
x_4_7 +x_9_7<=1;
x_4_8 +x_9_8<=1;
x_4_9 +x_9_9<=1;
x_4_10 +x_9_10<=1;
x_4_11 +x_9_11<=1;
x_4_12 +x_9_12<=1;
x_5_1 +x_7_1<=1;
x_5_2 +x_7_2<=1;
x_5_3 +x_7_3<=1;
x_5_4 +x_7_4<=1;
x_5_5 +x_7_5<=1;
x_5_6 +x_7_6<=1;
x_5_7 +x_7_7<=1;
x_5_8 +x_7_8<=1;
x_5_9 +x_7_9<=1;
x_5_10 +x_7_10<=1;
x_5_11 +x_7_11<=1;
x_5_12 +x_7_12<=1;
x_5_1 +x_10_1<=1;
x_5_2 +x_10_2<=1;
x_5_3 +x_10_3<=1;
x_5_4 +x_10_4<=1;
x_5_5 +x_10_5<=1;
x_5_6 +x_10_6<=1;
x_5_7 +x_10_7<=1;
x_5_8 +x_10_8<=1;
x_5_9 +x_10_9<=1;
x_5_10 +x_10_10<=1;
x_5_11 +x_10_11<=1;
x_5_12 +x_10_12<=1;
x_5_1 +x_12_1<=1;
x_5_2 +x_12_2<=1;
x_5_3 +x_12_3<=1;
x_5_4 +x_12_4<=1;
x_5_5 +x_12_5<=1;
x_5_6 +x_12_6<=1;
x_5_7 +x_12_7<=1;
x_5_8 +x_12_8<=1;
x_5_9 +x_12_9<=1;
x_5_10 +x_12_10<=1;
x_5_11 +x_12_11<=1;
x_5_12 +x_12_12<=1;
x_6_1 +x_7_1<=1;
x_6_2 +x_7_2<=1;
x_6_3 +x_7_3<=1;
x_6_4 +x_7_4<=1;
x_6_5 +x_7_5<=1;
x_6_6 +x_7_6<=1;
x_6_7 +x_7_7<=1;
x_6_8 +x_7_8<=1;
x_6_9 +x_7_9<=1;
x_6_10 +x_7_10<=1;
x_6_11 +x_7_11<=1;
x_6_12 +x_7_12<=1;
x_6_1 +x_11_1<=1;
x_6_2 +x_11_2<=1;
x_6_3 +x_11_3<=1;
x_6_4 +x_11_4<=1;
x_6_5 +x_11_5<=1;
x_6_6 +x_11_6<=1;
x_6_7 +x_11_7<=1;
x_6_8 +x_11_8<=1;
x_6_9 +x_11_9<=1;
x_6_10 +x_11_10<=1;
x_6_11 +x_11_11<=1;
x_6_12 +x_11_12<=1;
x_7_1 +x_8_1<=1;
x_7_2 +x_8_2<=1;
x_7_3 +x_8_3<=1;
x_7_4 +x_8_4<=1;
x_7_5 +x_8_5<=1;
x_7_6 +x_8_6<=1;
x_7_7 +x_8_7<=1;
x_7_8 +x_8_8<=1;
x_7_9 +x_8_9<=1;
x_7_10 +x_8_10<=1;
x_7_11 +x_8_11<=1;
x_7_12 +x_8_12<=1;
x_7_1 +x_9_1<=1;
x_7_2 +x_9_2<=1;
x_7_3 +x_9_3<=1;
x_7_4 +x_9_4<=1;
x_7_5 +x_9_5<=1;
x_7_6 +x_9_6<=1;
x_7_7 +x_9_7<=1;
x_7_8 +x_9_8<=1;
x_7_9 +x_9_9<=1;
x_7_10 +x_9_10<=1;
x_7_11 +x_9_11<=1;
x_7_12 +x_9_12<=1;
x_7_1 +x_10_1<=1;
x_7_2 +x_10_2<=1;
x_7_3 +x_10_3<=1;
x_7_4 +x_10_4<=1;
x_7_5 +x_10_5<=1;
x_7_6 +x_10_6<=1;
x_7_7 +x_10_7<=1;
x_7_8 +x_10_8<=1;
x_7_9 +x_10_9<=1;
x_7_10 +x_10_10<=1;
x_7_11 +x_10_11<=1;
x_7_12 +x_10_12<=1;
x_7_1 +x_11_1<=1;
x_7_2 +x_11_2<=1;
x_7_3 +x_11_3<=1;
x_7_4 +x_11_4<=1;
x_7_5 +x_11_5<=1;
x_7_6 +x_11_6<=1;
x_7_7 +x_11_7<=1;
x_7_8 +x_11_8<=1;
x_7_9 +x_11_9<=1;
x_7_10 +x_11_10<=1;
x_7_11 +x_11_11<=1;
x_7_12 +x_11_12<=1;
x_7_1 +x_12_1<=1;
x_7_2 +x_12_2<=1;
x_7_3 +x_12_3<=1;
x_7_4 +x_12_4<=1;
x_7_5 +x_12_5<=1;
x_7_6 +x_12_6<=1;
x_7_7 +x_12_7<=1;
x_7_8 +x_12_8<=1;
x_7_9 +x_12_9<=1;
x_7_10 +x_12_10<=1;
x_7_11 +x_12_11<=1;
x_7_12 +x_12_12<=1;
x_8_1 +x_11_1<=1;
x_8_2 +x_11_2<=1;
x_8_3 +x_11_3<=1;
x_8_4 +x_11_4<=1;
x_8_5 +x_11_5<=1;
x_8_6 +x_11_6<=1;
x_8_7 +x_11_7<=1;
x_8_8 +x_11_8<=1;
x_8_9 +x_11_9<=1;
x_8_10 +x_11_10<=1;
x_8_11 +x_11_11<=1;
x_8_12 +x_11_12<=1;
x_9_1 +x_12_1<=1;
x_9_2 +x_12_2<=1;
x_9_3 +x_12_3<=1;
x_9_4 +x_12_4<=1;
x_9_5 +x_12_5<=1;
x_9_6 +x_12_6<=1;
x_9_7 +x_12_7<=1;
x_9_8 +x_12_8<=1;
x_9_9 +x_12_9<=1;
x_9_10 +x_12_10<=1;
x_9_11 +x_12_11<=1;
x_9_12 +x_12_12<=1;
x_11_1 +x_12_1<=1;
x_11_2 +x_12_2<=1;
x_11_3 +x_12_3<=1;
x_11_4 +x_12_4<=1;
x_11_5 +x_12_5<=1;
x_11_6 +x_12_6<=1;
x_11_7 +x_12_7<=1;
x_11_8 +x_12_8<=1;
x_11_9 +x_12_9<=1;
x_11_10  +x_12_10<=1;
x_11_11  +x_12_11<=1;
x_11_12  +x_12_12<=1;
bin
y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8,y_9,y_10,y_11,y_12,x_1_1,x_1_2,x_1_3,x_1_4,x_1_5,x_1_6,x_1_7,
x_1_8,x_1_9,x_1_10,x_1_11,x_1_12,x_2_1,x_2_2,x_2_3,x_2_4,x_2_5,x_2_6,x_2_7,x_2_8,x_2_9,x_2_10
,x_2_11,x_2_12,x_3_1,x_3_2,x_3_3,x_3_4,x_3_5,x_3_6,x_3_7,x_3_8,x_3_9,x_3_10,x_3_11,x_3_12,x_4
_1,x_4_2,x_4_3,x_4_4,x_4_5,x_4_6,x_4_7,x_4_8,x_4_9,x_4_10,x_4_11,x_4_12,x_5_1,x_5_2,x_5_3,x_5
_4,x_5_5,x_5_6,x_5_7,x_5_8,x_5_9,x_5_10,x_5_11,x_5_12,x_6_1,x_6_2,x_6_3,x_6_4,x_6_5,x_6_6,x_6
_7,x_6_8,x_6_9,x_6_10,x_6_11,x_6_12,x_7_1,x_7_2,x_7_3,x_7_4,x_7_5,x_7_6,x_7_7,x_7_8,x_7_9,x_7
_10,x_7_11,x_7_12,x_8_1,x_8_2,x_8_3,x_8_4,x_8_5,x_8_6,x_8_7,x_8_8,x_8_9,x_8_10,x_8_11,x_8_12,
x_9_1,x_9_2,x_9_3,x_9_4,x_9_5,x_9_6,x_9_7,x_9_8,x_9_9,x_9_10,x_9_11,x_9_12,x_10_1,x_10_2,x_10
_3,x_10_4,x_10_5,x_10_6,x_10_7,x_10_8,x_10_9,x_10_10,x_10_11,x_10_12,x_11_1,x_11_2,x_11_3,x_1
1_4,x_11_5,x_11_6,x_11_7,x_11_8,x_11_9,x_11_10,x_11_11,x_11_12,x_12_1,x_12_2,x_12_3,x_12_4,x_
12_5,x_12_6,x_12_7,x_12_8,x_12_9,x_12_10,x_12_11,x_12_12;
------------------------------------------------------
```

Then I took this lp3.txt as input into lp_solve, here is the output from lp_solve in terminal:
Note: the output is realigned to be multiple column in Microsoft Word to avoid too many pages.

```
------------------------------------------------------------
Value of objective function: 5.00000000

Actual values of the variables:
y_1         1
y_2         1
y_3         1
y_4         1
y_5         1
y_6         0
y_7         0
y_8         0
y_9         0
y_10        0
y_11        0
y_12        0
x_1_1       1
x_1_2       0
x_1_3       0
x_1_4       0
x_1_5       0
x_1_6       0
x_1_7       0
x_1_8       0
x_1_9       0
x_1_10      0
x_1_11      0
x_1_12      0
x_2_1       0
x_2_2       1
x_2_3       0
x_2_4       0
x_2_5       0
x_2_6       0
x_2_7       0
x_2_8       0
x_2_9       0
x_2_10      0
x_2_11      0
x_2_12      0
x_3_1       0
x_3_2       0
x_3_3       1
x_3_4       0
x_3_5       0
x_3_6       0
x_3_7       0
x_3_8       0
x_3_9       0
x_3_10      0
x_3_11      0
x_3_12      0
x_4_1       1
x_4_2       0
x_4_3       0
x_4_4       0
x_4_5       0
x_4_6       0
x_4_7       0
x_4_8       0
x_4_9       0
x_4_10      0
x_4_11      0
x_4_12      0
x_5_1       1
x_5_2       0
x_5_3       0
x_5_4       0
x_5_5       0
x_5_6       0
x_5_7       0
x_5_8       0
x_5_9       0
x_5_10      0
x_5_11      0
x_5_12      0
x_6_1       0
x_6_2       1
x_6_3       0
x_6_4       0
x_6_5       0
x_6_6       0
x_6_7       0
x_6_8       0
x_6_9       0
x_6_10      0
x_6_11      0
x_6_12      0
x_7_1       0
x_7_2       0
x_7_3       0
x_7_4       1
x_7_5       0
x_7_6       0
x_7_7       0
x_7_8       0
x_7_9       0
x_7_10      0
x_7_11      0
x_7_12      0
x_8_1       0
x_8_2       1
x_8_3       0
x_8_4       0
x_8_5       0
x_8_6       0
x_8_7       0
x_8_8       0
x_8_9       0
x_8_10      0
x_8_11      0
x_8_12      0
x_9_1       0
x_9_2       0
x_9_3       0
x_9_4       0
x_9_5       1
x_9_6       0
x_9_7       0
x_9_8       0
x_9_9       0
x_9_10      0
x_9_11      0
x_9_12      0
x_10_1      0
x_10_2      0
x_10_3      1
x_10_4      0
x_10_5      0
x_10_6      0
x_10_7      0
x_10_8      0
x_10_9      0
x_10_10     0
x_10_11     0
x_10_12     0
x_11_1      0
x_11_2      0
x_11_3      0
x_11_4      0
x_11_5      1
x_11_6      0
x_11_7      0
x_11_8      0
x_11_9      0
x_11_10     0
x_11_11     0
x_11_12     0
x_12_1      0
x_12_2      1
x_12_3      0
x_12_4      0
x_12_5      0
x_12_6      0
x_12_7      0
x_12_8      0
x_12_9      0
x_12_10     0
x_12_11     0
x_12_12     0
------------------------------------------------------------
```

Therefore, from this solution, we know that the minimum number of colors required to color the graph when no two connected vertices are the same color is 5, in another word, the smallest number of needed colors, the chromatic number of the graph *G* is 5. $\boldsymbol{\chi(G) = 5}$**.**

For example:

- **Color 1** — 1, 4, 5
- **Color 2** — 2, 6, 8, 12
- **Color 3** — 3, 10
- **Color 4** — 7
- **Color 5** — 9, 11