1. a. We use x_{YZ} to stand for the number of guards in the door between Y and Z, x_B for Entrance

the total number of guards is $x_B + x_{AB} + x_{BC} + x_{AD} + x_{BE} + x_{BF} + x_{CG} + x_{DH} + x_{EH} + x_{GJ} + x_{HI} + x_{FG}$

Due to budget, we need at most 1 guard on every door.

namely, $x_B, x_{AB}, x_{BC}, x_{AD}, x_{BE}, x_{BF}, x_{CG}, x_{DH}, x_{EH}, x_{GJ}, x_{HI}, x_{FG} \in \{0,1\}$

Each of the rooms is controlled by at least one guard.

So $x_{AD} + x_{AB} \geq 1$

$x_{AB} + x_B + x_{BC} + x_{BF} + x_{BE} \geq 1$

$x_{BC} + x_{CG} \geq 1$

$x_{AD} + x_{DH} \geq 1$

$x_{BE} + x_{EH} \geq 1$

$x_{BF} + x_{FG} \geq 1$

$x_{CG} + x_{GJ} + x_{FG} \geq 1$

$x_{DH} + x_{EH} + x_{HI} \geq 1$

$x_{HI} \geq 1$

$x_{GJ} \geq 1$

The problem is

$$\min \; x_B + x_{AB} + x_{BC} + x_{AD} + x_{BE} + x_{BF} + x_{CG} + x_{DH} + x_{EH} + x_{GJ} + x_{HI} + x_{FG}$$

s.t.

$$\begin{aligned}
&x_{AD} + x_{AB} \geq 1 \\
&x_{AB} + x_B + x_{BC} + x_{BF} + x_{BE} \geq 1 \\
&x_{BC} + x_{CG} \geq 1 \\
&x_{AD} + x_{DH} \geq 1 \\
&x_{BE} + x_{EH} \geq 1 \\
&x_{BF} + x_{FG} \geq 1 \\
&x_{CG} + x_{GJ} + x_{FG} \geq 1 \\
&x_{DH} + x_{EH} + x_{HI} \geq 1 \\
&x_{HI} \geq 1 \\
&x_{GJ} \geq 1 \\
&x_B, x_{AB}, x_{BC}, x_{AD}, x_{BE}, x_{BF}, x_{CG}, x_{DH}, x_{EH}, x_{GJ}, x_{HI}, x_{FG} \in \{0,1\}
\end{aligned}$$

b. 6 guards needed

```python
# no relax
minGuards = 12
for x_B in (0,1):
    for x_AB in (0,1):
        for x_BC in (0, 1):
            for x_AD in (0, 1):
                for x_BE in (0, 1):
                    for x_BF in (0, 1):
                        for x_CG in (0, 1):
                            for x_DH in (0, 1):
                                for x_EH in (0, 1):
                                    for x_GJ in (0, 1):
                                        for x_HI in (0, 1):
                                            for x_FG in (0, 1):
                                                if x_AD + x_AB >= 1 and x_AB + x_B + x_BC + x_BF + x_BE >= 1 and x_BC + x_CG >= 1 and \
                                                    x_AD + x_DH >= 1 and x_BE + x_EH >= 1 and x_BF + x_FG >= 1 and x_CG + x_GJ + x_FG >= 1 and\
                                                    x_DH + x_EH + x_HI >= 1 and x_HI >= 1 and x_GJ >= 1:
                                                    if minGuards > x_B + x_AB + x_BC + x_AD + x_BE + x_BF + x_CG + x_DH + x_EH + x_GJ + x_HI + x_FG:
                                                        minGuards = x_B + x_AB + x_BC + x_AD + x_BE + x_BF + x_CG + x_DH + x_EH + x_GJ + x_HI + x_FG
                                                        x_B_opt = x_B
                                                        x_AB_opt = x_AB
                                                        x_BC_opt = x_BC
                                                        x_AD_opt = x_AD
                                                        x_BE_opt = x_BE
                                                        x_BF_opt = x_BF
                                                        x_CG_opt = x_CG
                                                        x_DH_opt = x_DH
                                                        x_EH_opt = x_EH
                                                        x_GJ_opt = x_GJ
                                                        x_HI_opt = x_HI
                                                        x_FG_opt = x_FG
```

```
minGuards =  6
x_B_opt =  0
x_AB_opt =  0
x_BC_opt =  0
x_AD_opt =  1
x_BE_opt =  0
x_BF_opt =  1
x_CG_opt =  1
x_DH_opt =  0
x_EH_opt =  1
x_GJ_opt =  1
x_HI_opt =  1
x_FG_opt =  0
```

```matlab
%% LP relaxation
cvx_begin
variables x_B x_AB x_BC x_AD x_BE x_BF x_CG x_DH x_EH x_GJ x_HI x_FG
    minimize (x_B + x_AB + x_BC + x_AD + x_BE + x_BF + x_CG + x_DH + x_EH + x_GJ + x_HI + x_FG)
    subject to
        x_AD + x_AB >= 1
        x_AB + x_B + x_BC + x_BF + x_BE >= 1
        x_BC + x_CG >= 1
        x_AD + x_DH >= 1
        x_BE + x_EH >= 1
        x_BF + x_FG >= 1
        x_CG + x_GJ + x_FG >= 1
        x_DH + x_EH + x_HI >= 1
        x_HI >= 1
        x_GJ >= 1
        0 <= x_B
        0 <= x_AB
        0 <= x_BC
        0 <= x_AD
        0 <= x_BE
        0 <= x_BF
        0 <= x_CG
        0 <= x_DH
        0 <= x_EH
        0 <= x_GJ
        0 <= x_HI
        0 <= x_FG
        1 >= x_B
        1 >= x_AB
        1 >= x_BC
        1 >= x_AD
        1 >= x_BE
        1 >= x_BF
        1 >= x_CG
        1 >= x_DH
        1 >= x_EH
        1 >= x_GJ
        1 >= x_HI
        1 >= x_FG
cvx_end
```

Solution:

```
Status: Solved
Optimal value (cvx_optval): +6

x_B =

   3.9789e-10

x_AB =

   1.1973e-09

x_BC =

    0.5253

x_AD =

    1.0000

x_BE =

    0.4788

x_BF =

    0.5253

x_CG =

    0.4747

x_GJ =

    1.0000

x_DH =

   1.1978e-09

x_EH =

    0.5212

x_HI =

    1.0000

x_FG =

    0.4747
```

2.

```
cvx_begin
variables x y
maximize (17*x+12*y)
subject to
    10*x + 7*y <= 40
    x + y <= 5
    x >= 0
    y >= 0
cvx_end
x
y
```

```
Optimal value (cvx_optval): +68.3333

x =

    1.6667

y =

    3.3333
```

```
cvx_begin
variables x y
maximize (17*x+12*y)
subject to
    10*x + 7*y <= 40
    x + y <= 5
    x >= 0
    y >= 0

    x <= 1
cvx_end
x
y
```

```
Optimal value (cvx_optval): +65

x =

    1.0000

y =

    4.0000
```

```
cvx_begin
variables x y
maximize (17*x+12*y)
subject to
    10*x + 7*y <= 40
    x + y <= 5
    x >= 0
    y >= 0

    x >= 2
cvx_end
x
y
```

```
Optimal value (cvx_optval): +68.2857

x =

    2.0000

y =

    2.8571
```

```
cvx_begin
variables x y
maximize (17*x+12*y)
subject to
    10*x + 7*y <= 40
    x + y <= 5
    x >= 0
    y >= 0

    x >= 2
    y <= 2
cvx_end
x
y
```

```
Optimal value (cvx_optval): +68.2

x =

    2.6000

y =

    2.0000
```

```
cvx_begin
variables x y
maximize (17*x+12*y)
subject to
    10*x + 7*y <= 40
    x + y <= 5
    x >= 0
    y >= 0

    x >= 2
    y >= 3
cvx_end
x
y
```

```
Status: Infeasible
```

```
cvx_begin
variables x y
maximize (17*x+12*y)
subject to
    10*x + 7*y <= 40
    x + y <= 5
    x >= 0
    y >= 0

    x >= 2
    y <= 2
    x <= 2
cvx_end
x
y
```

```
Optimal value (cvx_optval): +58

x =

    2.0000

y =

    2.0000
```

```
cvx_begin
variables x y
maximize (17*x+12*y)
subject to
    10*x + 7*y <= 40
    x + y <= 5
    x >= 0
    y >= 0

    x >= 2
    y <= 2
    x >= 3
cvx_end
x
y
```

```
Optimal value (cvx_optval): +68.1429

x =

    3.0000

y =

    1.4286
```

```
cvx_begin
variables x y
maximize (17*x+12*y)
subject to
    10*x + 7*y <= 40
    x + y <= 5
    x >= 0
    y >= 0

    x >= 2
    y <= 2
    x >= 3
    y <= 1
cvx_end
x
y
```

```
Optimal value (cvx_optval): +68.1

x =

    3.3000

y =

    1.0000
```

```
cvx_begin
variables x y
maximize (17*x+12*y)
subject to
    10*x + 7*y <= 40
    x + y <= 5
    x >= 0
    y >= 0

    x >= 2
    y <= 2
    x >= 3
    y >= 2
cvx_end
x
y
```

```
Status: Infeasible
```

```
cvx_begin
variables x y
maximize (17*x+12*y)
subject to
    10*x + 7*y <= 40
    x + y <= 5
    x >= 0
    y >= 0

    x >= 2
    y <= 2
    x >= 3
    y <= 1
    x <= 3
cvx_end
x
y
```

```
Optimal value (cvx_optval): +63

x =

    3.0000

y =

    1.0000
```

```
cvx_begin
variables x y
maximize (17*x+12*y)
subject to
    10*x + 7*y <= 40
    x + y <= 5
    x >= 0
    y >= 0

    x >= 2
    y <= 2
    x >= 3
    y <= 1
    x >= 4
cvx_end
x
y
```

```
Optimal value (cvx_optval): +68

x =

    4.0000

y =

    1.2028e-09
```

actually, $y=0$

The optimal value is 68

The optimal solution is $\begin{cases} x=4 \\ y=0 \end{cases}$