PYTHON CODE CHECKER BOARD

```
def solve(grid):

    l=len(grid) #length of our grid

    for y in range(l): #for every row
        for x in range(l): #for every
column
            if grid[y][x]==0: # we can
place if there is no queen in given
position
                if possible(grid,y,x): #if
empty, check if we can place a queen
                    grid[y][x]=1 #if we
can, then place it
                    solve(grid) #pass grid
for recursive solution
                    #if we end up here,
means we searched through all children
branches
                    if sum(sum(a) for a in
grid)==l: #if there are 8 queens
                        return grid #we are
successful so return
                    grid[y][x]=0 #remove
the previous placed queen
```

```
    return grid #means we searched the 
space, we can return our result
```

OUTPUT

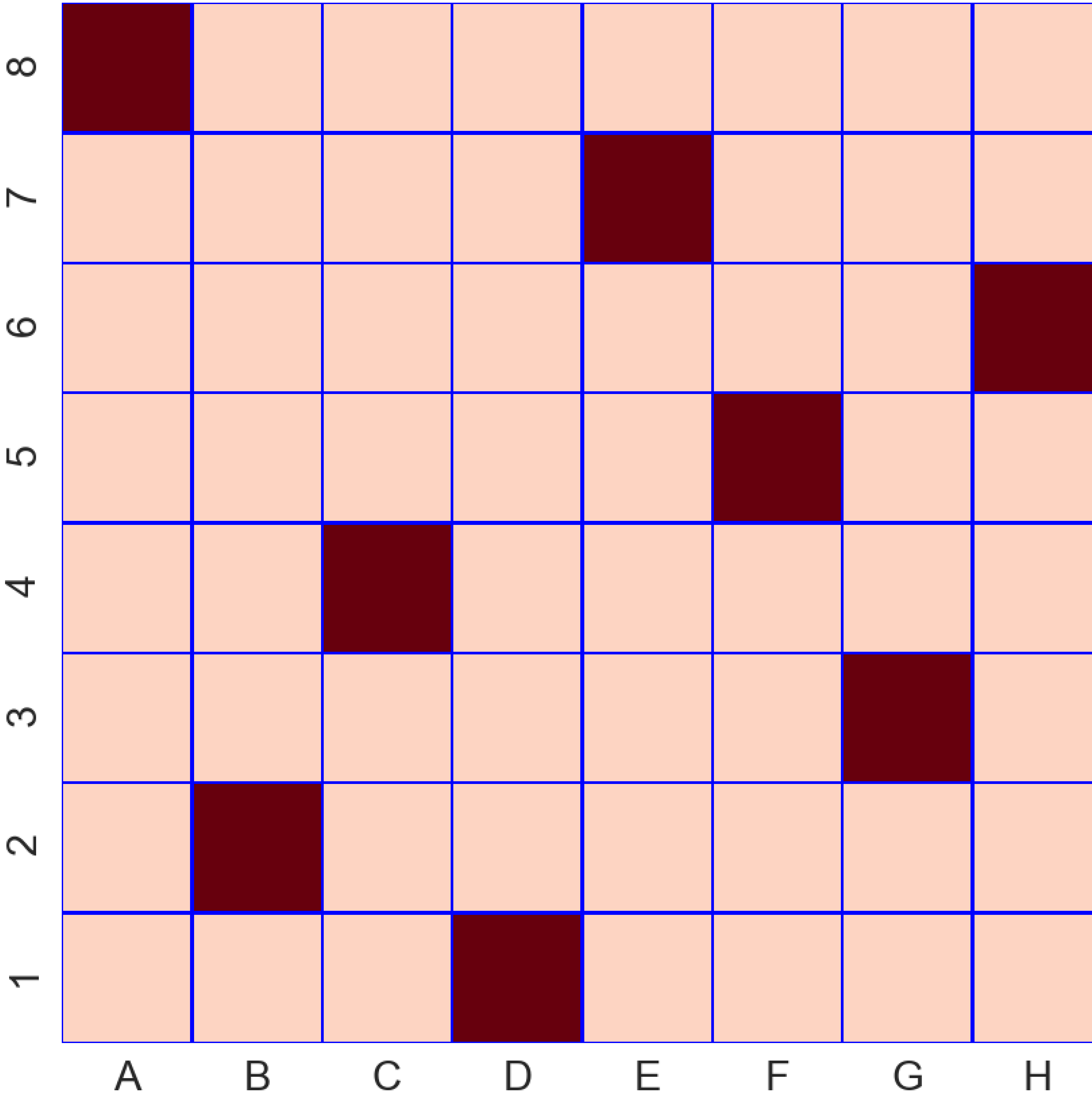
```
Solution = 
solve(copy.deepcopy(grid)) 
#get the solution
                                        print(np.matrix(Solution)) #Print the solution
```

(8×8 CHECKER BOARD)

WE CAN PLOT THIS EVEN USING CODE :

```
def plot(grid):# Plot the solution on a 
chessboard
    import seaborn as sns
    import matplotlib.pyplot as plt
    import string

    l=len(grid)
    Ly=list(range(1,l+1))[::-1]
    ly = [str(i) for i in Ly]
    Lx=list(string.ascii_uppercase)
    lx=Lx[:l]

    plt.close('all')
    sns.set(font_scale = 2)
    plt.figure(figsize=(10,10))
    ax = plt.gca() #you first need to get
the axis handle
    ax.set_aspect(1)

sns.heatmap(Solution,linewidths=.8,cbar=Fal
se,linecolor='blue',

cmap='Reds',center=0.4,xticklabels=lx,ytick
labels=ly)
```

OUTPUT AS .


8
7
6
5
4
3
2
1
A
B
C
D
E
F
G
H