Assignment No 5

November 14, 2022

[8]:

```python
# row = r
# column = c
# board = b
# rowlookup = rl
# slashcode = sc
# backslashcode = bsc
# slashcodelookup = scl
# backslashcodelookup = bscl
n = int(input())

def is_safe(r,c,sc,bsc,rl,scl,bscl):
    if (scl[sc[r][c]] or bscl[bsc[r][c]] or rl[r]):
        return False
    return True


def solve_N_Queen_until(b,c,sc,bsc,rl,scl,bscl):
    if c >= n:
        return True
    for i in range(n):
        if(is_safe(i,c,sc,bsc,rl,scl,bscl)):
           b[i][c]=1
           rl[i] = True
           scl[sc[i][c]] = True
           bscl[bsc[i][c]] = True

           if (solve_N_Queen_until(b,c+1,sc,bsc,rl,scl,bscl)):
               return True

           b[i][c] = 0
           rl[i] = False
           scl[sc[i][c]] = False
           bscl[bsc[i][c]] = False
    return False
b = [[0 for i in range(n)] for j in range(n)]
sc = [[0 for i in range(n)] for j in range(n)]
bsc = [[0 for i in range(n)] for j in range(n)]
```

```
rl = [False]*n
x = 2*n-1
scl = [False]*x
bscl = [False]*x

for rr in range(n):
    for cc in range(n):
        sc[rr][cc] = rr+cc
        bsc[rr][cc] = rr-cc+(n-1)
if (solve_N_Queen_until(b,0,sc,bsc,rl,scl,bscl)==False):
    print('solution does not exit')
else:
    for i in range(n):
        for j in range(n):
            print(b[i][j],end=" ")
        print()
```

```
8
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0
```