#include <stdio.h>

int is\_attack(int i, int j, int board[5][5], int N) {

int k, l;

// checking for column j

for(k=1; k<=i-1; k++) {

if(board[k][j] == 1)

return 1;

}

// checking upper right diagonal

k = i-1;

l = j+1;

while (k>=1 && l<=N) {

if (board[k][l] == 1)

return 1;

k=k+1;

l=l+1;

}

// checking upper left diagonal

k = i-1;

l = j-1;

while (k>=1 && l>=1) {

if (board[k][l] == 1)

return 1;

k=k-1;

l=l-1;

}

return 0;

}

int n\_queen(int row, int n, int N, int board[5][5])

{

if (n==0)

return 1;

int j;

for (j=1; j<=N; j++)

{

if(!is\_attack(row, j, board, N))

{

board[row][j] = 1;

if (n\_queen(row+1, n-1, N, board))

return 1;

board[row][j] = 0; //backtracking

}

}

return 0;

}

int main()

{

int board[5][5];

int i, j;

for(i=0;i<=4;i++)

{

for(j=0;j<=4;j++)

board[i][j] = 0;

}

n\_queen(1, 4, 4, board);

//printing the matix

for(i=1;i<=4;i++)

{

for(j=1;j<=4;j++)

printf("%d\t",board[i][j]);

printf("\n");

}

return 0;

}