#include<iostream>

using namespace std;

bool issafe(int board[][10],int i,int j,int n)

{

//check for column

for(int row=0;row<i;row++)

{

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

return false;

}

//check for left diagonal

int x=i;

int y=j;

while(x>=0 && y>=0)

{

if(board[x][y]==1)

return false;

x--;

y--;

}

//check for right diagonal

x=i;

y=j;

while(x>=0 && y<n)

{

if(board[x][y]==1)

return false;

x--;

y++;

}

//the position is now safe

return true;

}

bool nqueen(int board[][10],int i,int n)

{

//Base case

if(i==n)

{

//You have successfully place queens in n rows(0...n-1)

//print the board

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

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

cout<<"Q ";

else

cout<<"# ";

}

cout<<endl;

}

return true;

}

//recursive case

//try to place the queen on the current row and call on the remaining part

for(int j=0;j<n;j++)

{

//check if ith,jth position is safe for the queen or not

if(issafe(board,i,j,n))

{

//place the queen- assuming i,j is the right position

board[i][j]=1;

bool nextqueenrekhpaye=nqueen(board,i+1,n);

if(nextqueenrekhpaye)

return true;

//i,j is not the right position -means our assumption is wrong

board[i][j]=0;

}

}

//you have tried for all position in the row but could not able to place the queen

return false;

}

int main()

{

int n;

cin>>n;

int board[10][10]={0};

nqueen(board,0,n);

}