```
def is_safe(board, row, col, N):
    for i in range(col):
        if board[row][i] == 1:
            return False

    for i, j in zip(range(row, -1, -1), range(col, -1, -1)):
        if board[i][j] == 1:
            return False

    for i, j in zip(range(row, N), range(col, -1, -1)):
        if board[i][j] == 1:
            return False

    return True

def solve_nqueens(board, col, N):
    if col >= N:
        return True

    for i in range(N):
        if is_safe(board, i, col, N):
            board[i][col] = 1

            if solve_nqueens(board, col + 1, N):
                return True

            board[i][col] = 0

    return False
```

```
def branch_and_bound_nqueens(N):
    board = [[0 for _ in range(N)] for _ in range(N)]
    if solve_nqueens(board, 0, N):
        return board
    else:
        return None

if __name__ == "__main":
    N = int(input("Enter the number of queens (N): "))
    result = branch_and_bound_nqueens(N)

    if result:
        for row in result:
            print(" ".join(map(str, row)))
    else:
        print("No solution exists for N =", N)
```

```
# Enter the number of queens (N): 4
# 0 1 0 0
# 0 0 0 1
# 1 0 0 0
# 0 0 1 0
```

```
Import random
```

```
Responses = {}

With open(“C:/Users/Siddharth/Downloads/responses.txt”, “r”) as file:
    For line in file:
        Key, response_str = line.strip().split(“: “)
        Responses[key] = response_str.split(“|”)

Def get_response(user_input):
    User_input = user_input.lower()
    For key in responses:
        If key in user_input:
            Return random.choice(responses[key])
    Return “I can’t understand this.”

Print(“Chatbot: How can I assist you?”)

While True:
    User_input = input(“You: “)
    If user_input.lower() == “exit”:  # Corrected this line
        Print(“Chatbot: Goodbye”)
        Break

    Response = get_response(user_input)
    Print(“Chatbot:”, response)
```