Bansilal Ramnath Agarwal Charitable Trust's
VISHWAKARMA INSTITUTE OF TECHNOLOGY – PUNE

ARTIFICIAL INTELLIGENCE

Division	A
Batch	2
GR-no	12311493
Roll no	54
Name	Atharva Kangralkar

Assignment 4:

Solve the N-Queen by using the CSP algorithm

Description:-

When you enter 8 as the number of queens: • The program allocates an array board [8] to store column positions of queens for each row. • It calls solve(board, 0, 8), which starts placing queens row by row. • The isSafe function ensures that each queen is placed without attacking others. • If a valid placement is found for all 8 queens, the program prints a solution

- **Code:**

```
#include <stdio.h>
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
bool isSafe(int *board, int row, int col, int n) {
	for (int i = 0; i < row; i++) {
		if (board[i] == col || abs(board[i] - col) == abs(i - row))
			return false;
	}
	return true;
}
bool solve(int *board, int row, int n) {
	if (row == n) {
		for (int i = 0; i < n; i++)
			printf("%d ", board[i]);
		printf("\n");
		return true;
	}
	for (int col = 0; col < n; col++) {
		if (isSafe(board, row, col, n)) {
			board[row] = col;
			if (solve(board, row + 1, n)) return true;
		}
	}
	return false;
}
int main() {
	int n;
	printf("Enter number of queens: ");
	scanf("%d", &n);
	int *board = (int *)malloc(n * sizeof(int));
	if (!board) {
		printf("Memory allocation failed\n");
		return 1;
	}
	if (!solve(board, 0, n))
		printf("No solution found\n");
	free(board);
	return 0;
```

}

Screenshots/Output: