**DYNAMMIC PROGRAMMING**

PLAYING WITH CHESSBOARD

AIM:

Ram is given with an n*n chessboard with each cell with a monetary value. Ram stands at the (0,0), that the position of the top left white rook. He is been given a task to reach the bottom right black rook position (n-1, n-1) constrained that he needs to reach the position by traveling the maximum monetary path under the condition that he can only travel one step right or one step down the board. Help ram to achieve it by providing an efficient DP algorithm.

CODE:

```
#include <stdio.h>

#include <stdlib.h>


#define MAX 100

int maxMonetaryPath(int n, int board[MAX][MAX]) {
   int dp[MAX][MAX];
   dp[0][0] = board[0][0];
   for (int j = 1; j < n; j++) {
      dp[0][j] = dp[0][j - 1] + board[0][j];
   }
   for (int i = 1; i < n; i++) {
      dp[i][0] = dp[i - 1][0] + board[i][0];
   }
   for (int i = 1; i < n; i++) {
      for (int j = 1; j < n; j++) {
         dp[i][j] = board[i][j] + (dp[i - 1][j] > dp[i][j - 1] ? dp[i - 1][j] : dp[i][j - 1]);
      }
```

```
    }
    return dp[n - 1][n - 1];
}

int main() {
    int n;
    int board[MAX][MAX];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &board[i][j]);
        }
    }

    // Find and print the maximum monetary path
    int result = maxMonetaryPath(n, board);
    printf("%d\n", result);

    return 0;
}
```

INPUT AND OUTPUT:

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 3<br>1 2 4<br>2 3 4<br>8 7 1 | 19 | 19 | ✔ |