Dynamic Programming

Program – 2

Aim:

To determine the maximum monetary value Ram can collect while traveling from the top-left corner to the bottom-right corner of an n×nn \times nn×n chessboard, moving only right or down.

Input:

- First line: An integer nnn – the size of the chessboard.
- Next nnn lines: nnn integers per line – the monetary values of the chessboard cells.

Code:

```
#include <stdio.h>


int max_monetary_path(int n, int board[n][n]) {
   int dp[n][n];
   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;
    scanf("%d", &n);

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

    printf("%d\n", max_monetary_path(n, board));

    return 0;
}
```

Output:

	Input	Expected	Got	
✔	3 1 2 4 2 3 4 8 7 1	19	19	✔