```c
#include <stdio.h>
#include <stdlib.h>

#define ROW    3 /* 行の要素数 */
#define COLUMN 4 /* 列の要素数 */
/* 行列の領域確保 */
double **dmatrix(int nr1, int nr2, int nl1, int nl2);
/* 行列の領域解放 */
void free_dmatrix(double **a, int nr1, int nr2, int nl1, int nl2);
/* a[m1...m2][n1...n2] と b[m1...m2][n1...n2] の和を求める．結果は c へ */
void matrix_sum(double **a, double **b, double **c, int m1, int m2, int n1, int n2);

int main(void)
{
        double **a, **b, **c;
        int i, j;

        a = dmatrix(1, ROW, 1, COLUMN); /* 行列 a[1...ROW][1...COLUMN] */
        b = dmatrix(1, ROW, 1, COLUMN); /* 行列 b[1...ROW][1...COLUMN] */
        c = dmatrix(1, ROW, 1, COLUMN); /* 行列 c[1...ROW][1...COLUMN] */

        /* 行列の定義 */
        for(i = 1;i <= ROW; i++){
                for(j = 1;j <= COLUMN; j++){
                        a[i][j] = 2.0 * (i + j);  b[i][j] = 3.0 * (i + j);
                }
        }

        /* 行列の和の計算 */
        matrix_sum(a, b, c, 1, ROW, 1, COLUMN);

        /* 結果の表示 */
        printf("行列 A と行列 B の和は次の通りです\n");
        for(i = 1; i <= ROW; i++){
                for(j = 1;j <= COLUMN; j++){
                        printf("%f ",c[i][j]);
                }
                printf("\n");
        }

        /* 行列領域の解放 */
        free_dmatrix(a, 1, ROW, 1, COLUMN);
        free_dmatrix(b, 1, ROW, 1, COLUMN);
        free_dmatrix(c, 1, ROW, 1, COLUMN);

        return 0;
}

double **dmatrix(int nr1, int nr2, int nl1, int nl2)
{
        int i, nrow, ncol;
        double **a;

        nrow = nr2 - nr1 + 1; /* 行の数 */
        ncol = nl2 - nl1 + 1; /* 列の数 */

        /* 行の確保 */
        if((a = malloc(nrow * sizeof(double *)) )== NULL){
                printf("メモリが確保できません (行列 a)\n");
                exit(1);
        }
        a = a - nr1; /* 行をずらす */

        /* 列の確保 */
        for(i = nr1; i <= nr2; i++) a[i] = malloc(ncol * sizeof(double));
        for(i = nr1; i <= nr2; i++) a[i] = a[i] - nl1;                    /* 列をずらす */

        return (a);
}

void free_dmatrix(double **a, int nr1, int nr2, int nl1, int nl2)
{
        int i;

        /* メモリの解放 */
        for(i = nr1; i <= nr2; i++) free((void *)(a[i] + nl1));
        free((void *)(a + nr1));
}

/* 行列の和 */
/* a[m1...m2][n1...n2] と b[m1...m2][n1...n2] の和を求める．結果は c へ */
void matrix_sum(double **a, double **b, double **c, int m1, int m2, int n1, int n2)
{
        int i, j;
        for(i = m1; i <= m2; i++){
                for(j = n1; j <= n2; j++){
                        c[i][j] = a[i][j] + b[i][j];
                }
        }
}
```