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

#define L 3
#define M 2
#define N 3

/* 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);
/* 行列の積の計算 */
void matrix_product(double **a, double **b, double **c, int l1, int l2, int m1, int m2, int n1, int n2);


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

        a = dmatrix(1, L, 1, M); /* 行列 a[1...L][1...M] */
        b = dmatrix(1, M, 1, N); /* 行列 b[1...M][1...N] */
        c = dmatrix(1, L, 1, N); /* 行列 c[1...L][1...N] */

        /* 行列 A の定義 */
        for(i = 1;i <= L; i++){
                for(j = 1;j <= M; j++){
                        a[i][j] = 2.0 * (i + j);
                }
        }
        /* 行列 B の定義 */
        for(i = 1;i <= M; i++){
                for(j = 1;j <= N; j++){
                        b[i][j] = 2.0 * (i + j);
                }
        }

        /* 行列の積の計算 */
        matrix_product(a, b, c, 1, L, 1, M, 1, N);
        /* 結果の表示 */
        printf("A x B の結果は次の通りです. \n");
        for(i = 1; i <= L; i++){
                for(j = 1;j <= N; j++){
                        printf("%f ",c[i][j]);
                }
                printf("\n");
        }

        /* 行列領域の解放 */
        free_dmatrix(a, 1, L, 1, M);
        free_dmatrix(b, 1, M, 1, N);
        free_dmatrix(c, 1, L, 1, N);

        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));
}

void matrix_product(double **a, double **b, double **c, int l1, int l2, int m1, int m2, int n1, int n2)
{
        int i, j, k;

        for(i = l1; i <= l2; i++){                      /* 行列の添え字 */
                for(j = n1; j <= n2; j++){
                        c[i][j] = 0.0;                  /* 変数の初期化 */
                        for(k = m1; k <= m2; k++){  /* 列の添字 */
                                c[i][j] += a[i][k] * b[k][j];
                        }
                }
        }
}
```