```
Alexs-MacBook-Air:7 alex$ cat main.c
#include<stdio.h>
#include<math.h>

typedef struct {
    int M;
    int N;
    int LB[101];
    int YE[100];
} matrix;

void readMatrix(matrix *A);

void printVectorsOfMatrix(matrix *A);

void printNormalMatrix(matrix *A);

int isDiag(matrix *A);

int multiplyMatrix(matrix *A, matrix *B, matrix *R);


int main(void)
{
    matrix A;
    matrix B;
    matrix R;

    readMatrix(&A);
    readMatrix(&B);
    printf("\nA\n");
    printNormalMatrix(&A);
    printVectorsOfMatrix(&A);
    printf("B\n");
    printNormalMatrix(&B);
    printVectorsOfMatrix(&B);

    if (multiplyMatrix(&A, &B, &R) == 1) {
        printf("Res\n");
        printNormalMatrix(&R);
        printVectorsOfMatrix(&R);
        if (isDiag(&R)) {
            printf("^Mairix is Diagonal^\n");
        } else {
            printf("^Mairix isn't Diagonal^\n");
        }
    }
    putchar('\n');

    return 0;
}


void readMatrix(matrix *A)
{
    int no = 0;
```

```
    int tmp;

    scanf("%d%d", &A->M, &A->N);

    for (int m = 0; m < A->M; m++) {
        for (int n = 0; n < A->N; n++) {
        scanf("%d", &tmp);
        if (tmp != 0) {
            A->LB[no] = n + m * A->N;
            A->YE[no] = tmp;
            no++;
            }
        }
    }
    A->LB[no] = -1;
}

void printVectorsOfMatrix(matrix *A)
{
    int i = 0;
    printf("--------------------\nLB  YE\n");
    while (A->LB[i] != -1) {
        printf("%d\t%d\n",A->LB[i], A->YE[i]);
        i++;
    }
    printf("%d\n--------------------\n",A->LB[i]);
}

void printNormalMatrix(matrix *A)
{
    int no = 0;
    printf("--------------------\n");
    for (int m = 0; m < A->M; m++) {
        for (int n = 0; n < A->N; n++) {
        if (n + m * A->N == A->LB[no]) {
            printf("%d\t", A->YE[no]);
            no++;
        } else {
            printf("0\t");
        }
        }
        putchar('\n');
    }
    printf("--------------------\n");
}

int isDiag(matrix *A)
{
    if (A->M != A->N) {
        return 0;
    }
    for (int i = 0; A->LB[i] != -1; i++) {
        if (A->LB[i] / A->N != A->LB[i] % A->N) {
            return 0;
        }
    }
```

```
    return 1;
}

int multiplyMatrix(matrix *A, matrix *B, matrix *R)
{
    if (A->N != B->M) {
        printf("Can't multiply matrices. Number of A's rows isn't match
number of B's columns\n");
        return 0;
    }

    R->M = A->M;
    R->N = B->N;
    for (int i = 0; i < 100;i++) {
        R->YE[i] = 0;
    }

    int no = 0;
    R->LB[0] = -1;

    for (int i = 0; A->LB[i] != -1; i++) {
        for (int k = 0; B->LB[k] != -1; k++) {
            if (A->LB[i] % A->N == B->LB[k] / B->N) {
                no = 0;
                for (int l = 0; R->LB[l] != -1; l++) {
                    if (A->LB[k] % B->N + A->LB[i] / A->N * B->N == R-
>LB[l]) {
                        break;
                    }
                    no++;
                }
                if (R->LB[no] == -1) {
                    R->LB[no + 1] = -1;
                }
                R->LB[no] = B->LB[k] % B->N + A->LB[i] / A->N * B->N;
                R->YE[no] += A->YE[i] * B->YE[k];
            }
        }
    }
    for (int z = 0; R->LB[z] != -1; z++) {
        if (R->YE[z] == 0) {
            for (int z_2 = z; R->LB[z_2] != -1; z_2++) {
                R->LB[z_2] = R->LB[z_2 + 1];
                R->YE[z_2] = R->YE[z_2 + 1];
            }
        }
    }
    return 1;
}
Alexs-MacBook-Air:7 alex$ gcc -Wall -pedantic -std=c99 -g main.c
Alexs-MacBook-Air:7 alex$ ./a.out < test1.txt

A
--------------------
0   0   0   1
2   0   0   0
```

```
0   8   9   0
--------------------
--------------------
LB  YE
3   1
4   2
9   8
10  9
-1
--------------------
B
--------------------
0   5
0   0
1   1
0   0
--------------------
--------------------
LB  YE
1   5
4   1
5   1
-1
--------------------
Res
--------------------
0   0
0   10
9   9
--------------------
--------------------
LB  YE
3   10
4   9
5   9
-1
--------------------
^Mairix isn't Diagonal^


Alexs-MacBook-Air:7 alex$ ./a.out < test2.txt

A
--------------------
0   0   1
2   0   0
6   0   0
--------------------
--------------------
LB  YE
2   1
3   2
6   6
-1
--------------------
B
```

```
--------------------
0   0   0
0   4   0
1   0   0
--------------------
--------------------
LB  YE
4   4
6   1
-1
--------------------
Res
--------------------
1   0   0
0   0   0
0   0   0
--------------------
--------------------
LB  YE
0   1
-1
--------------------
^Mairix is Diagonal^
```