Crie uma função capaz de realizar multiplicação matricial da forma C=A * B. A função deve receber 6 argumentos: os ponteiros para as matrizes A, B e C, o número de linhas e colunas de A e o número de colunas de B (assuma que o número de coluna de A é igual ao número de linhas de B). O resultado da multiplicação deve ficar armazenado em C. Crie um programa para testar sua implementação, capaz de utilizar a função de multiplicação e imprimir as três matrizes. A função criada para multiplicação não deve realizar nenhum tipo de saída de dados no terminal.

Programa (Input):

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

void multiplicacao(float **a, float **b, float **c, int linhasA, int
colunasA, int colunasB){
    int i, j, k;
    int x = 0;
    for(i=0; i<linhasA; i++){
        for(j=0; j<colunasB; j++){
            for(k = 0; k < colunasA; k++){
                x = x + a[i][k]*b[k][j];
            }
        c[i][j] = x;
        x = 0;
        }
    }
}


int main() {
    int nlA, ncA, ncB, i, j;
    float **A, **B, **C;
    printf("Digite o número de linhas de A: ");
    scanf("%d", &nlA);
    printf("Digite o número de colunas de A: ");
    scanf("%d", &ncA);
    printf("Digite o número de colunas de B: ");
    scanf("%d", &ncB);
    A = malloc(nlA*sizeof(float*));
    A[0] = malloc(ncA*nlA*sizeof(float));
    for(i=1; i<nlA; i++){
        A[i] = A[i-1]+ncA;
    }
```

```
    for(i=0; i<nlA; i++){
        for(j=0; j<ncA; j++){
        printf("Digite o elemento da linha %d e da coluna %d da Matriz A
\n", i+1, j+1);
        scanf("%f", &A[i][j]);
        }
        printf("\n");
    }
    B = malloc(ncA*sizeof(float*));
    B[0] = malloc(ncB*ncA*sizeof(float));
    for(i=1; i<ncA; i++){
        B[i] = B[i-1]+ncB;
    }
    for(i=0; i<ncA; i++){
        for(j=0; j<ncB; j++){
        printf("Digite o elemento da linha %d e da coluna %d da Matriz B
\n", i+1, j+1);
        scanf("%f", &B[i][j]);
        }
        printf("\n");
    }
    C = malloc(nlA*sizeof(float*));
    C[0] = malloc(ncB*nlA*sizeof(float));
    for(i=1; i<nlA; i++){
        C[i] = C[i-1]+ncB;
    }
    multiplicacao(A, B, C, nlA, ncA, ncB);
    printf("Matriz A: \n");
    for (i=0; i<nlA; i++){
        for (j=0; j<ncA; j++){
            printf("%.f ", A[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    printf("Matriz B: \n");
    for (i=0; i<ncA; i++){
        for (j=0; j<ncB; j++){
            printf("%.f ", B[i][j]);
        }
        printf("\n");
    }
    printf("\n");
    printf("Matriz C: \n");
    for (i=0; i<nlA; i++){
        for (j=0; j<ncB; j++){
            printf("%.f ", C[i][j]);
        }
        printf("\n");
    }
    printf("\n");
```

```
    free(C[0]);
    free(C);
    free(B[0]);
    free(B);
    free(A[0]);
    free(A);
    return 0;
}
```

Output (Exemplo):

```
clang-7 -pthread -lm -o main main.c
./main
Digite o número de linhas de A: 2
Digite o número de colunas de A: 2
Digite o número de colunas de B: 1
Digite o elemento da linha 1 e da coluna 1 da Matriz A
1
Digite o elemento da linha 1 e da coluna 2 da Matriz A
2

Digite o elemento da linha 2 e da coluna 1 da Matriz A
3
Digite o elemento da linha 2 e da coluna 2 da Matriz A
4

Digite o elemento da linha 1 e da coluna 1 da Matriz B
5

Digite o elemento da linha 2 e da coluna 1 da Matriz B
6

Matriz A:
1 2
3 4

Matriz B:
5
6
```

```
Matriz C:
17
39

```