

Two-dimensional Arrays

1. Specification

Write a C program to multiply two 2-dimensional matrices.

Given a matrix A having m rows and n columns and matrix B having n rows and p columns, the product of $A \times B$ is a matrix C having m rows and p columns. Each element $C(i, j)$ is computed as follows:

$$C(i, j) = \sum_{k=1}^n (A(i, k) \times B(k, j))$$

2. Implementation

- The program to be submitted is named `mm.c`. **Use the given template `mm.c`** and fill in your code. Submit only file `mm.c`.
- You are also given a file named `mmMain.c` to test your code. Do not submit file `mmMain.c`.
- The user will input the dimensions of the matrices to be multiplied as command-line arguments to be stored in variables `nr1`, `nc1` and `nc2`.
 - `nr1`: number of rows of matrix (array) A .
 - `nc1`: number of columns of matrix A .
 - `nc2`: number of columns of matrix B .
 - The number of rows of matrix B is the same as the number of columns of matrix A and thus does not need to be specified explicitly.
- Your first task is to implement function `get_args()` to convert the command-line arguments (which are strings) to integers.
- In `main()` function in file `mmMain.c`, two arrays are allocated for your use.
- Your second task is to implement function `initMatrix()` to initialize a matrix. Element (i, j) is assigned value $i+j$. For example, `a[0][4] = 0 + 4 = 4`. Note that matrix indices start from 0 to be consistent with array indexing in C.
- Your third task is to implement function `matrix_mult()` to multiply the two matrices just allocated and initialized. Allocate an array `c` and store the result in `c`. Return the pointer to array `c` to the caller.
- Assume that all user inputs are valid. No error checking is required on inputs.

- Do not modify the function definitions.
- You may define local variables inside the functions.
- You may define and implement your own function(s) inside file `mm.c` if needed.
- **Do not use any C library functions (e.g., `atoi`) except `malloc()`, `calloc()` and `free()`.**
- **Do not use array indexing in file `mm.c`. Use only pointers to manipulate array elements. You will get a 50% point reduction if you use array indexing.** If you use pointers only, make sure that your submitted file does not contain the square brackets, even in the comments.
- To compile the program, use the following command: `cc mm.c mmMain.c -o mm`

3. Sample Inputs/Outputs

See file `mm_out.txt` for examples from running programs `mm.c` and `mmMain.c`.