

# Mathematics u11601 (C Programming) Michaelmas 2021

November 10, 2021

## Sixth assignment, due 12 noon, Wednesday 17/11/21

**Plagiarism.** The plagiarism policy is as always: you will not copy another student's assignment. If copying is detected, all students involved will lose marks, irrespective of who copied from whom.

**Read this carefully.** You should form the habit of reading specifications carefully, and following them.

**The assignment is** to write a C program, check that it works, and submit the C program. Your program is to multiply two matrices. The input will be in the form

```
k ell (int, the dimensions of the first array)
a_{00} a_{01} ... a_{ell-1,m-1} (double, the first array)
m n (dimensions of second array)
b_{00} b_{01} ... b_{m-1, n-1} (second array)
```

All your arrays should be  $10 \times 10$  double, the maximum possible size. You need to write, and use, two routines

```
void multiply ( int k, int ell, double a[10][10],
               int m, int n, double b[10][10], double c[10][10] )
which writes the product of a and b into c.
and
void print_matrix ( int m, int n, double a[10][10] )
to print the first m rows and n columns of
the array a.
```

It is assumed

- (1)  $1 \leq k, ell, m, n \leq 10$  and
- (2)  $ell == m$ , i.e., the arrays can be multiplied.
- (3) Note that only the first  $k$  rows and  $ell$  columns of  $a$  are in use, similarly for  $b$  and for the output array  $c$ .

For example:

```

% gcc 10x10.c
% cat data-10by10
3 4
1 2 3 4
4 5 6 7
7 8 9 10
4 5
3 1 5 9 3
-1 5 -9 2 -1
4 -9 2 6 4
-5 -5 2 6 5 -5
%a.out < data-10by10
first matrix
    1.000    2.000    3.000    4.000
    4.000    5.000    6.000    7.000
    7.000    8.000    9.000   10.000
second matrix
    3.000    1.000    5.000    9.000    3.000
   -1.000    5.000   -9.000    2.000   -1.000
    4.000   -9.000    2.000    6.000    4.000
   -5.000   -5.000    2.000    6.000    5.000
product
   -7.000  -36.000    1.000   55.000   33.000
   -4.000  -60.000    1.000  124.000   66.000
   -1.000  -84.000    1.000  193.000   99.000
%
```

### Points to note.

The same remarks hold as in the previous assignments, especially that interactive input/output like

Please type in the next number:

should ***not*** occur.