

Mathematics u11601 (C Programming) Michaelmas 2021

September 27, 2021

1 First assignment, due 12 noon, Wednesday 6/10/21

Please excuse the severity of the next two paragraphs.

Plagiarism. These assignments are meant to be quite easy, at least to begin with. 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.

The program is to print the 'divided by 7' tables from 1 to 9999, as follows:

```
1 divided by 7 is 0, remainder 1
2 divided by 7 is 0, remainder 2
3 divided by 7 is 0, remainder 3
4 divided by 7 is 0, remainder 4
5 divided by 7 is 0, remainder 5
... and so on down to
9993 divided by 7 is 1427, remainder 4
9994 divided by 7 is 1427, remainder 5
9995 divided by 7 is 1427, remainder 6
9996 divided by 7 is 1428, remainder 0
9997 divided by 7 is 1428, remainder 1
9998 divided by 7 is 1428, remainder 2
9999 divided by 7 is 1428, remainder 3
```

Edit, compile, and run your program to make sure it works. When it works correctly, submit it. Submit it using `submit-work`, which runs on `hamilton` and `synge` and probably on other maths machines.

Points to note.

- Make sure your program works, *on the maths machines*, or at least that it 'compiles.' It is bad if a program is not working properly, but the *worst* thing you can do is to submit a program which does not compile on the maths machines. Such programs will be marked very severely.

- There are 6 ‘maths machines,’ running Unix, but two versions:

hamilton.maths.tcd.ie, aturing.maths.tcd.ie, and gstokes.maths.tcd.ie,
run ‘Linux.’ The C compiler on Linux is ‘gcc’

synge.maths.tcd.ie, jbell.maths.tcd.ie, and walton.maths.tcd.ie
run ‘FreeBSD.’ The C compiler on FreeBSD is ‘clang’

To find out which version you are using, type

```
echo $OSTYPE
```

(hostname will give the name of the machine you are logged in to).

- In programming, it is very important to follow a specification *exactly*, and for this reason you will always be expected to follow the specification *exactly*. Your program should print *exactly* as shown (given that 9999 lines are printed).
- You *should* use indentation to make the program as easy to read as possible. Statements between curly braces should be indented, and so on. (This practice has been followed in examples, without mentioning it.)
- For the submit-work software: steps in using it are illustrated by a series of screenshots on the course web-page. These screenshots were prepared for the course 1266. Replace 1266 by u11601 throughout. When you start submit-work, this week at least, you will see some text including

```
u11601-progs:ass01
```

- See below how *comments* can be written in a program ...

Appropriate indentation:

```
#include <stdio.h>    // at start of line

int main ()           // at start of line
{                     // brace at start of line
    printf("hello\n");
                        // indent two places, for readability

    printf("goodbye\n"); /* indent two places, for readability
}                        // brace at start of line
    /* This is a traditional style comment, also allowed,
       multi-line.
    */
```