

# Tutorial 11 - Compiling and Running C Locally

- You will not be using Seashell outside of CS136.
  - We will be covering how to compile and run C on your local machine.
1. Download a C compiler (and terminal on Windows)
  2. Download a text editor with syntax highlighting for C
  3. Write code in C
  4. Compile the code using the C compiler on the terminal
  5. Run the compiled code using the terminal

# Download a C compiler (Windows)

- Install MinGW with the `mingw-gcc-g++` option
- Add MinGW binaries to your `PATH` variable
- Install Git Bash to provide a Unix-like terminal for compiling and running code

# Download a C compiler (Mac)

- Open the Terminal app and enter  
`xcode-select -install`

# Download a Text Editor

- Seashell emulates a text editor and console.
- It provides syntax highlighting and other features.
- A text editor for writing C code will include several features, with syntax highlighting as the bare minimum.
- Sublime Text 3 is available for Mac, Windows, and Linux
- It requires a license but is free to use indefinitely
- Mac users may want to look into using Atom as an alternative

# Write code in C

- Implement the recursive algorithm for the Greatest Common Divisor (GCD) in C

$$\text{GCD}(a, b) = \begin{cases} a & \text{if } b = 0 \\ \text{GCD}(b, a \% b) & \text{otherwise} \end{cases}$$

- Bonus: use **command line arguments** to read in parameters

# Compile the C code using the terminal

- Using the terminal, navigate to the folder containing your C source code.
- You can use `cd [folder_name]` to "change directory" into a subfolder of the current folder. You can check the current folder with `pwd` ("present working directory").
- Compile your code using  
`gcc <source_code.c> -o <output_name>`
- This will generate a new executable binary file with the name `output_name`.

# Run the C code using the terminal

- You can run the new executable binary file by entering  
`./<output_name> [arg1] [arg2] [...]`
- `[arg1]` etc indicates optional command line arguments