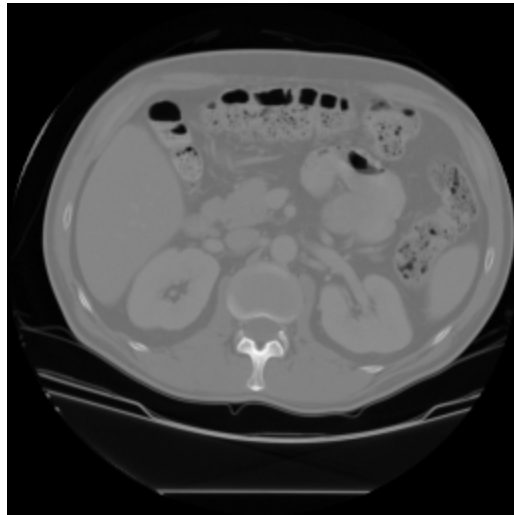
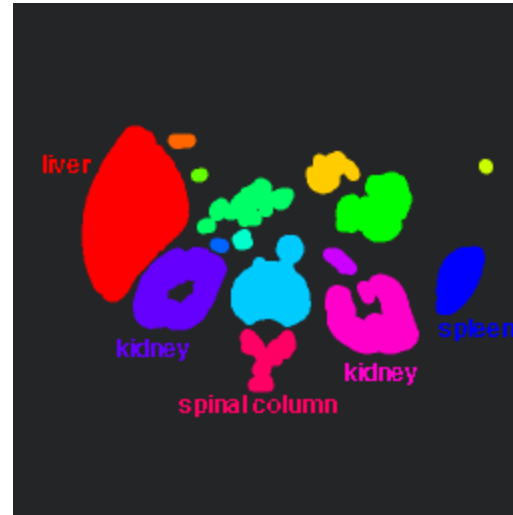


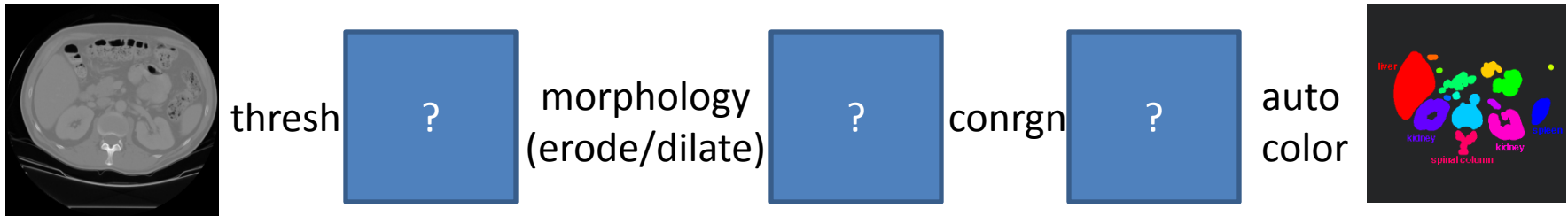
# HW1 Binary Vision in Medical Image Analysis



original CT image



colored connected components



We give you code for:

- thresh                      thresholding
- conrgn                      connected components
- autocolor                    coloring the components

You write code for:

- erode
- dilate

# Morphology Routines Command Sequence

- erode d <radius> input.pgm > output.pgm
- dilate d <radius> input.pgm > output.pgm

where the d stands for disc, b would be box, etc.

We require implementation of disc structuring elements of arbitrary size.

You can zero pad rows on the top and bottom, columns on the left and right, so the results are always full-sized images.

# Turn In (by 11:59pm October 7)

1. erode.cpp and dilate.cpp files with your code
2. a brief report showing your results on the three test images, which can be in Word or pdf

For each test image, your report should include:

- the test image with its name,
- thresholded binary image with the threshold you used,
- results of morphology with the operations you used
- the autocolored connected components. If you don't use autocolor, they will all look black.

# Evaluation: 10 pts

- Working program: 5pts
- Morphology testing : 3 pts
- Report : 1 pt
- Quality of results : 1 pt

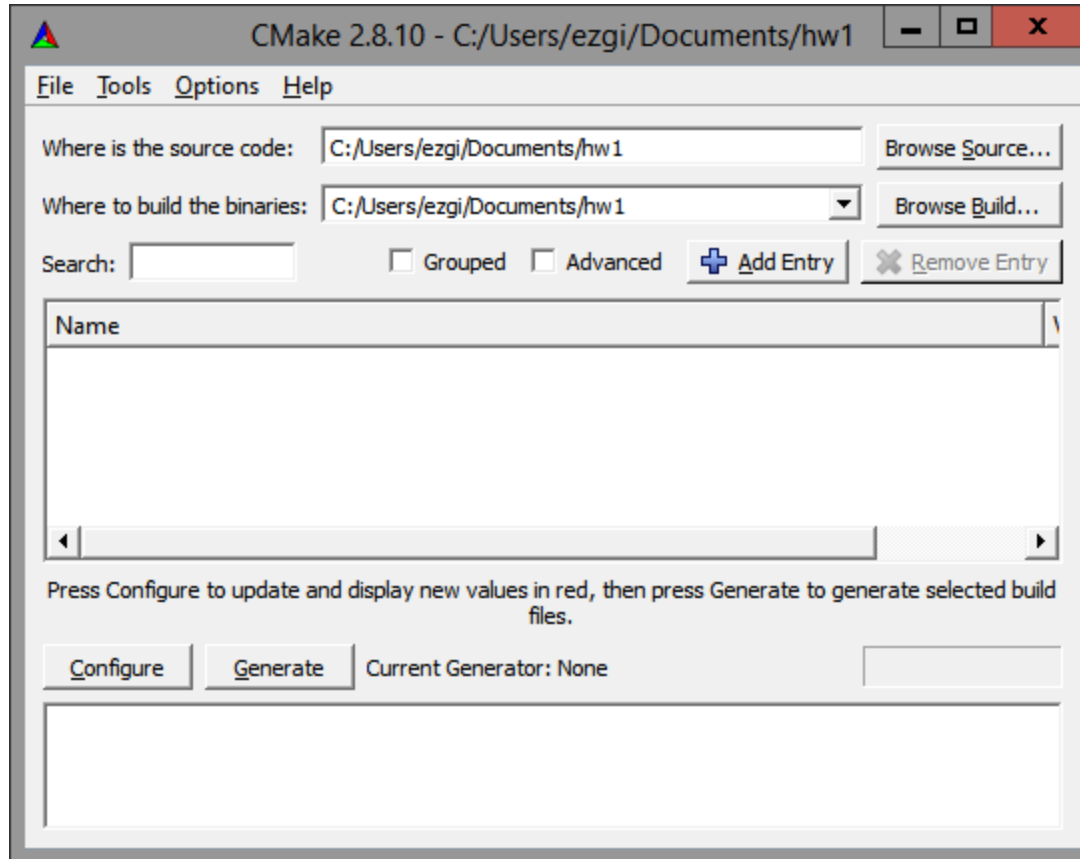
# Logistics

- Contents of hw1.zip package:
  - **images** : folder containing 3 test images
  - **utils.cpp, utils.h** : methods for reading/writing images
  - **thresh.cpp, conrgn.cpp** and **autocolor.cpp** : codes for the modules
  - **erode.cpp, dilate.cpp** : skeleton codes for you to complete
  - **CMakeLists.txt** : Build rules for CMake

# CMake

- Cross-platform make
- You provide source codes and the build rules (CMakeLists.txt), CMake configures the projects for any compiler.
- Link on the course web page:  
<http://www.CMake.org/>
- Download and install CMake on your computer.

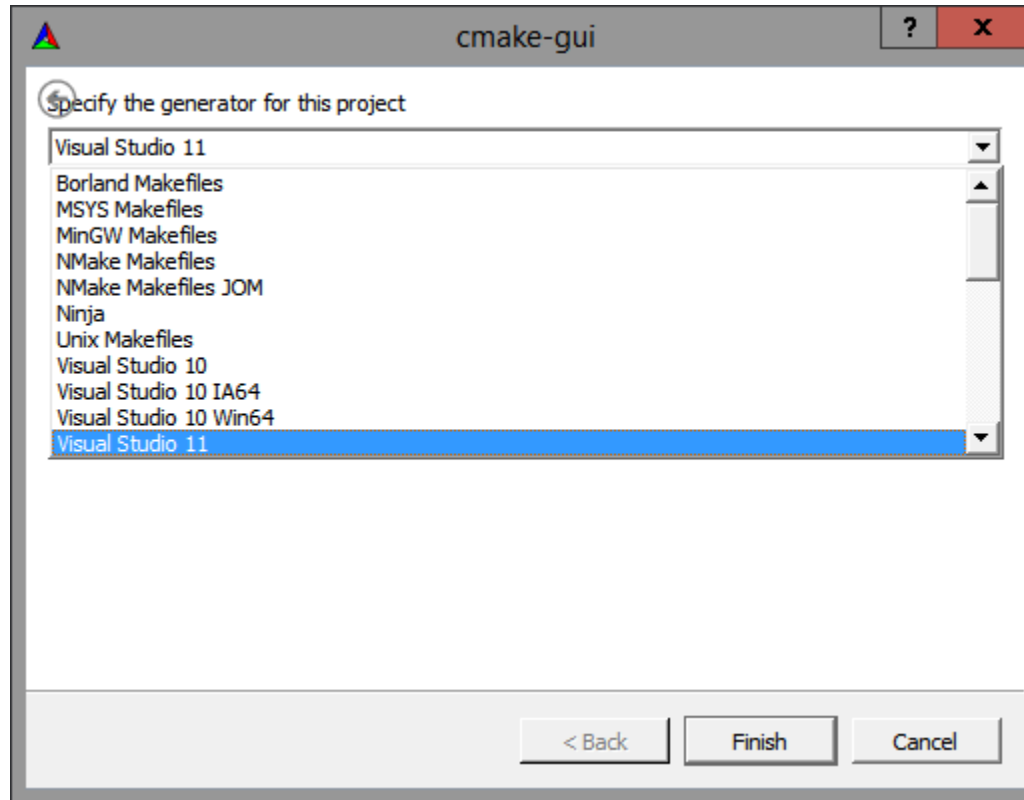
# CMake - Windows



Open CMake-gui and browse both Source and Build directories to where you unzipped the homework files. Click 'Configure'.

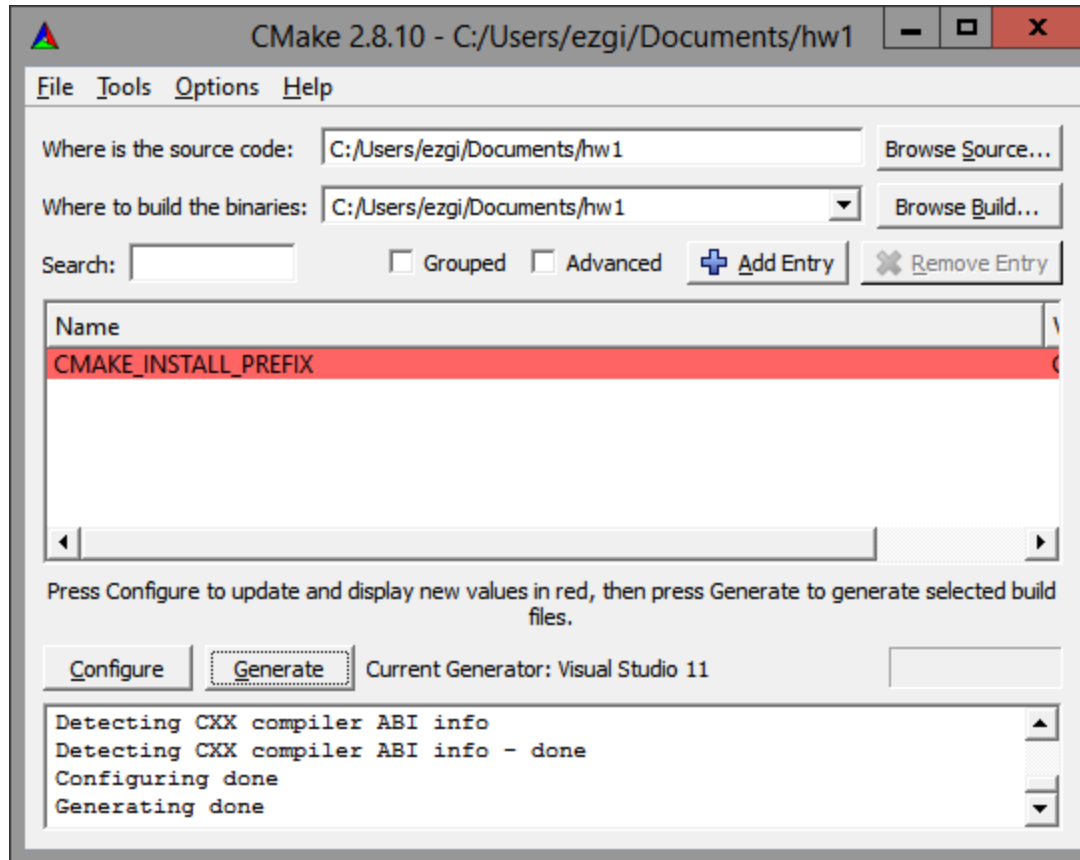


# CMake - Windows



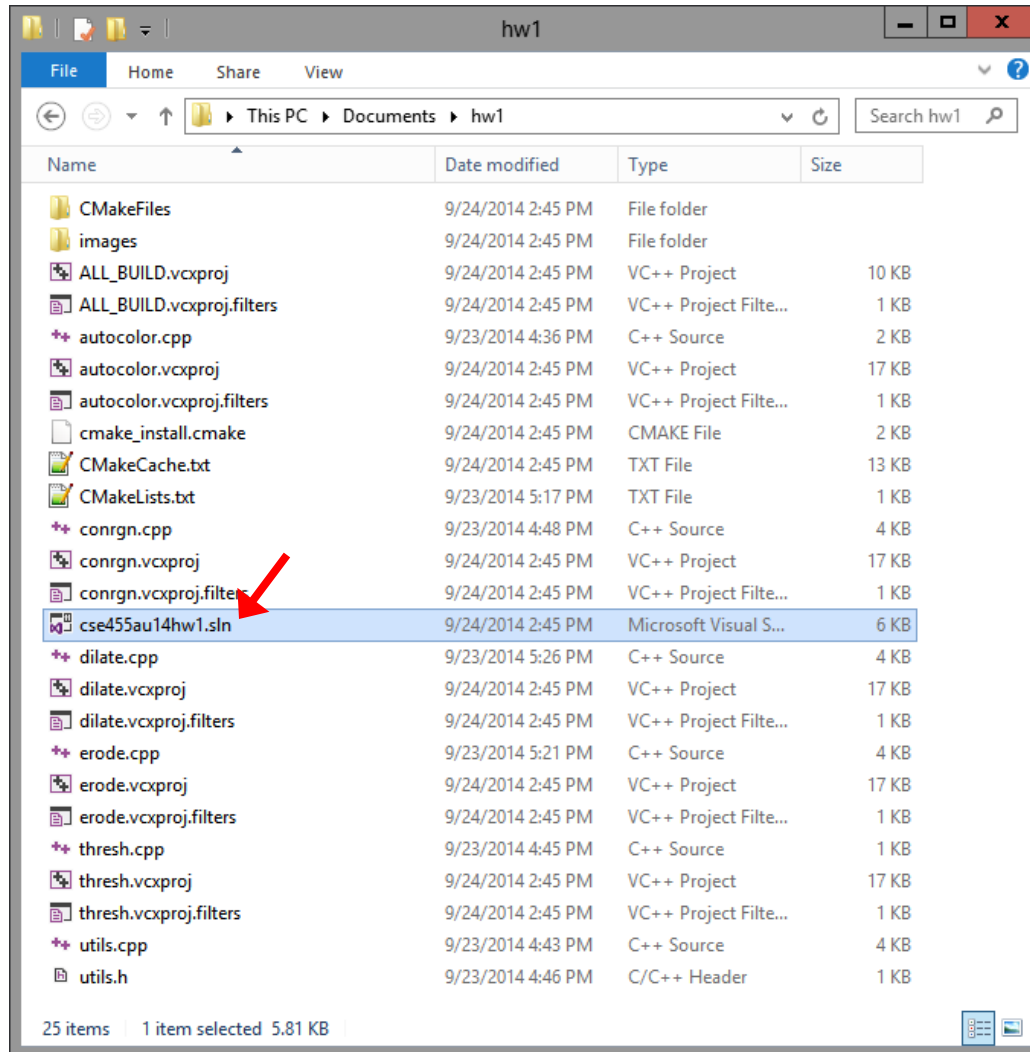
It will ask for a compiler. Pick your favorite.  
Mine is VS 11. Click 'Finish'.

# CMake - Windows



It will take a few seconds while CMake configures for your choice of compiler. Once finished, click 'Generate'.

# CMake - Windows



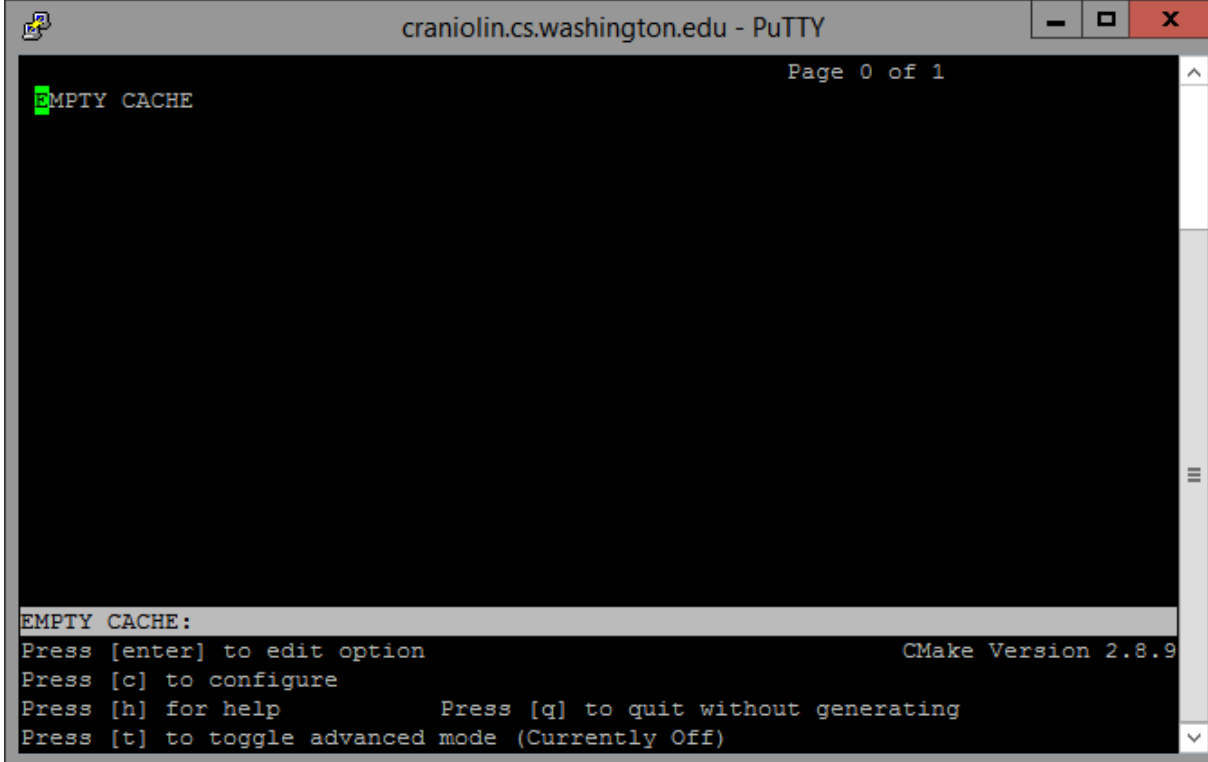
# CMake - Linux

A terminal window titled "craniolin.cs.washington.edu - PuTTY" showing a sequence of commands and their outputs. The user navigates to the "hw1" directory, lists the files, and runs "cmake .". The files listed are: autocolor.cpp, conrgn.cpp, erode.cpp, thresh.cpp, utils.h, CMakeLists.txt, dilate.cpp, images, and utils.cpp. The terminal text is as follows:

```
[ezgi@craniolin ~]$ cd hw1
[ezgi@craniolin hw1]$ ls
autocolor.cpp  conrgn.cpp  erode.cpp  thresh.cpp  utils.h
CMakeLists.txt  dilate.cpp  images     utils.cpp
[ezgi@craniolin hw1]$ cmake .
```

cd into the hw1 directory and call cmake in the same directory (since CMakeLists.txt is there).

# CMake - Linux

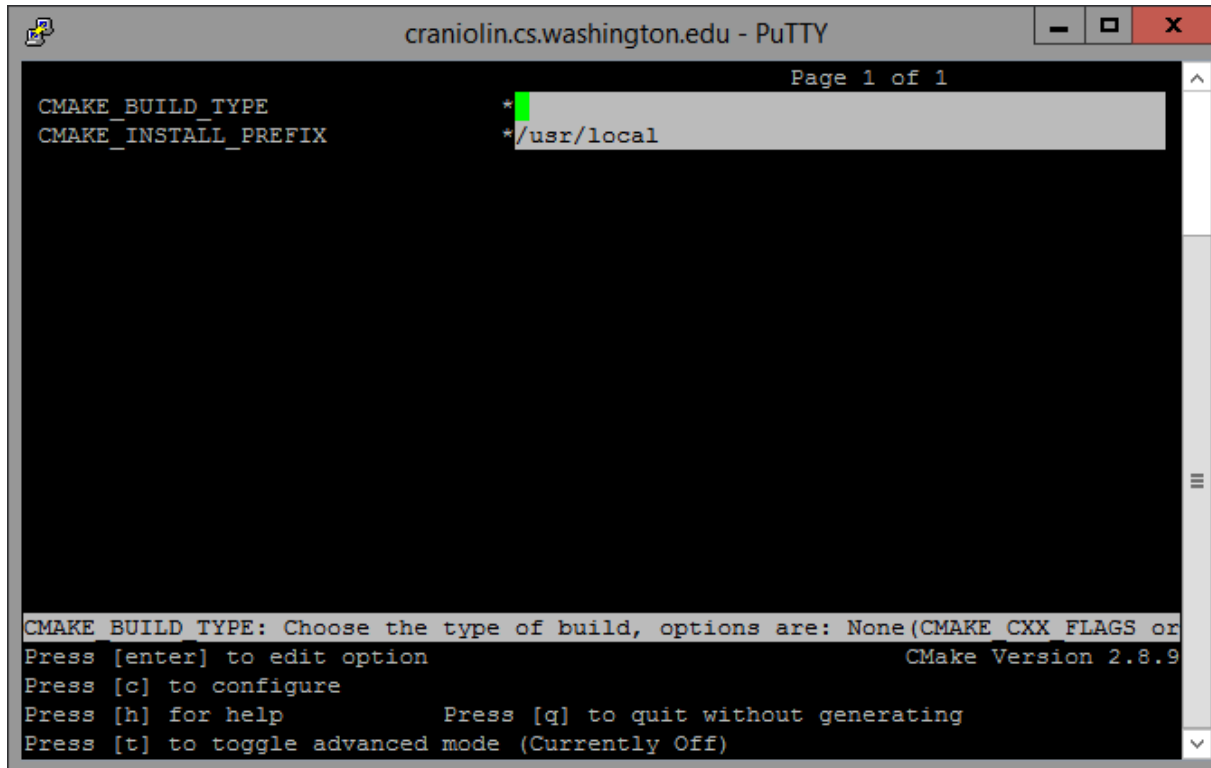


```
craniolin.cs.washington.edu - PuTTY
Page 0 of 1
EMPTY CACHE

EMPTY CACHE:
Press [enter] to edit option
Press [c] to configure
Press [h] for help
Press [t] to toggle advanced mode (Currently Off)
CMake Version 2.8.9
Press [q] to quit without generating
```

Press **c** to read CMakeLists.txt. It will fill this screen with a few options.

# CMake - Linux



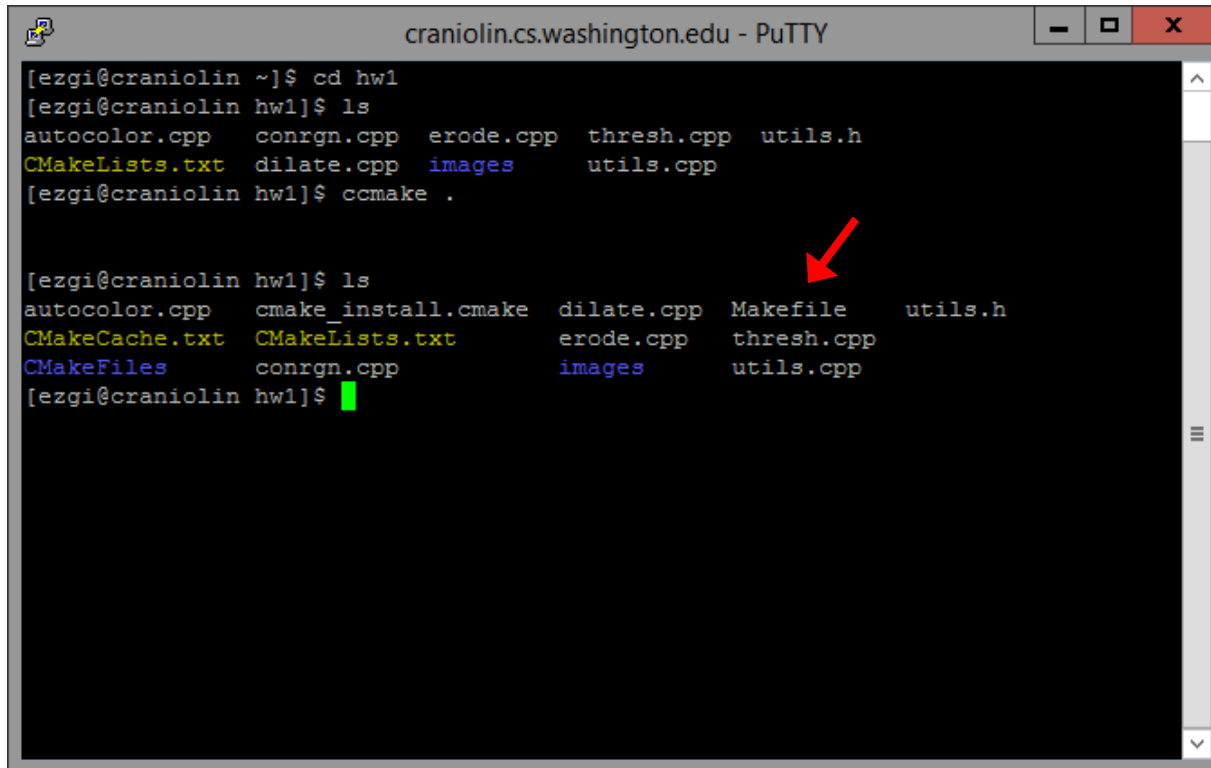
The screenshot shows a PuTTY terminal window titled "craniolin.cs.washington.edu - PuTTY". The terminal displays the CMake configuration interface. At the top, it says "Page 1 of 1". Below that, there are two options: "CMAKE\_BUILD\_TYPE" with a green cursor on the asterisk, and "CMAKE\_INSTALL\_PREFIX" with the value "/usr/local". At the bottom, there is a list of instructions: "CMAKE\_BUILD\_TYPE: Choose the type of build, options are: None(CMAKE\_CXX\_FLAGS or", "Press [enter] to edit option", "Press [c] to configure", "Press [h] for help", "Press [q] to quit without generating", and "Press [t] to toggle advanced mode (Currently Off)".

```
craniolin.cs.washington.edu - PuTTY
Page 1 of 1
CMAKE_BUILD_TYPE *
CMAKE_INSTALL_PREFIX */usr/local

CMAKE_BUILD_TYPE: Choose the type of build, options are: None(CMAKE_CXX_FLAGS or
Press [enter] to edit option
Press [c] to configure
Press [h] for help
Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

Press **c** again to configure. It will detect compilers.  
Once finished, press **g** to generate Makefile. It will exit if no error occurs.

# CMake - Linux



```
craniolin.cs.washington.edu - PuTTY
[ezgi@craniolin ~]$ cd hw1
[ezgi@craniolin hw1]$ ls
autocolor.cpp  conrgn.cpp  erode.cpp  thresh.cpp  utils.h
CMakeLists.txt  dilate.cpp  images     utils.cpp
[ezgi@craniolin hw1]$ cmake .

[ezgi@craniolin hw1]$ ls
autocolor.cpp  cmake_install.cmake  dilate.cpp  Makefile  utils.h
CMakeCache.txt  CMakeLists.txt      erode.cpp  thresh.cpp
CMakeFiles     conrgn.cpp           images     utils.cpp
[ezgi@craniolin hw1]$
```

A red arrow points to the 'Makefile' entry in the second 'ls' output.

Now you have a Makefile, and you can call 'make' to produce your binaries.

Good Luck!