

# CS 232 Introduction to C and Unix

## HW2 (Due on January 27, 2022, 11:59pm)

The goal of this homework assignment is to test your ability to write a simple program in C in VS Code and run a C program in the Linux server. After this homework assignment, you should feel comfortable writing programs in C that use variables, operators, and control flow statements. You should also be familiar with the development process on a \*nix computer. Please read the entire homework description carefully before beginning the coding.

### **Collaboration policy:**

Since this is the first real homework assignment, I want to be up-front about the collaboration policy for this course. For this homework assignment, you are to work alone. This means that you are not to discuss this homework with anyone other than me (the instructor). You can go over general concepts (e.g., for loops, variable declarations, using VS Code, SSH to a Linux server, etc.) with other class members, but talking about how to figure out prime numbers, looking at others' code, helping them debug it, etc. is expressly prohibited. The purpose behind this policy is to make sure that every member of the class is evaluated fairly, based upon his or her own work. If you have any questions about what is or is not acceptable collaboration, please contact me; it is better to be safe than sorry. Also, if you get stuck on part of the homework, you are encouraged to come by office hours. If you cannot make office hours, just send me an email and we can find a time to meet. I'm here to help you, but I do not know how to help if you don't ask me.

### **Homework description:**

Use VS Code to open the directory that hosts your homework, and then create a new subdirectory "**hw2**". In the Brightspace, you can find a file "primes.c" with part of a program already written. You should only modify that file in the places that are marked "**TODO: fill in the code ...**". Put primes.c to your VS Code workspace under the "hw2" directory.

For this assignment, you will complete the primes.c file so that it prints out all of the prime numbers that are greater than or equal to lowerBound and less than or equal to upperBound. (The values of these variables come from the command line. We will discuss how this is done later in the course.) The prime numbers should be printed in ascending order, with one number on each line. The program should not generate any other output. The number 1 should not be considered a prime number for this homework.

As an example, for lowerBound=1 and upperBound=7, the program should output:

```
2
3
5
7
```

Note that your program should not depend upon other files; all of your work should be done by filling in the appropriate section of primes.c. Furthermore, do not include any precomputed list of primes in your code. Here is a sample run in VS Code (using Git Bash):

```
$ gcc -g -o primes primes.c -Wall
$ ./primes 2 10
2
3
5
7
```

## Compiling and Debugging in VS Code:

After you have written your program, you will need to compile it using gcc. (Look at the lecture slides for how to use gcc. If you use functions from the math library, such as log or sqrt, you will need to link with the math library by adding a "-lm" at the end of the gcc command.) If successful, gcc will produce a program (e.g., the file "primes" in Unix or "primes.exe" in Windows), which you will want to test for correctness. You can do this by running the program (from the shell) with two command-line arguments for the lower and upper bounds. For example, if your program is "primes" then you would run something like "./primes 10 20", "./primes 3 7", or "./primes -10, 0" and check if the output is actually the proper list of primes.

## Testing

When you think the program is running correctly, you can test it using the provided utility. You can find a zip file in the Brightspace, containing two files: "runSimpleTest.sh" and "primesCheckOutput". Upload these two files and your source code "primes.c" to

diamond.pfw.edu

under "hw2" subdirectory in your home directory, using the "Remote SSH" extension in VS Code. The host "diamond.pfw.edu" is a Linux server hosted for our CS department. You can use your university credential to log in. In the appendix of this homework description, it provides the detailed information about how to connect to "diamond" Linux server using VS Code "Remote SSH" extension or Putty tool. Moreover, please watch the "cs232\_hw2.mp4" video for the demo.

Next, add the execution permission to "primesCheckOutput" and "runSimpleTest.sh" by using the chmod command:

```
$ chmod +x primesCheckOutput
$ chmod +x runSimpleTests.sh
```

Now, if you run `./runSimpleTests.sh` in the terminal, that utility will compile `primes.c` to `primes`, and then run `primes` on several bounds, checking the output. You won't get to see what your program prints out, but you will get messages telling you if your program passed the tests or not. You can then run your program manually with the bounds that caused a problem in order to see what might be going wrong. If your program passes all the test cases, that is a good indication that your program is correct. Note that in the `runSimpleTest.sh` file, it uses `"gcc -g primes.c -o primes -lm -Wall -std=c99"` to compile your source code. Here, `"-std=gnu99"` is added to compile the code using C99 mode in the Linux server.

## Submission

After testing the program, push the code `primes.c` to your GitHub `"Homework_CS232"` repository under the directory `"hw2"`. When you attempt to push the code from local VS Code to GitHub, please make sure you have used VS code to open the homework `"root"` directory, instead of the `"hw2"` directory.

If you have trouble in pushing the code or any questions, please let the instructor or TA know.

## Grading rubric:

- Push `primes.c` to GitHub under `hw2` directory – 10pt
- Program `primes.c` can be compiled – 10pt
- Program `primes.c` works correctly (e.g., pass `runSimpleTests.sh` tests) – 70pt
- Program `primes.c` uses correct indentation, and the code is readable with necessary comments – 10pt

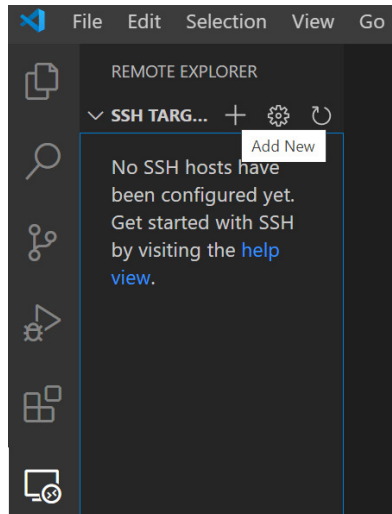
Total: 100 points.

## Appendix

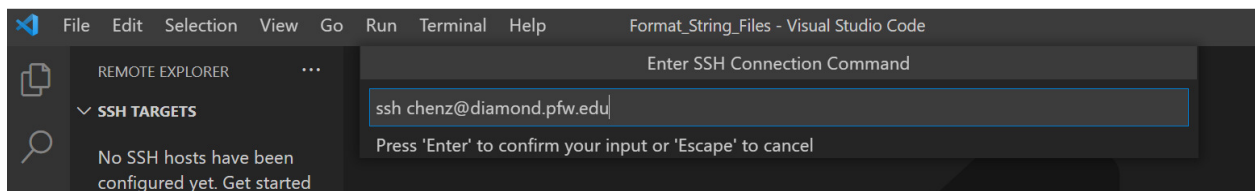
### Use VS Code `"Remote SSH"` Extension to Connect to `"diamond"` Linux Server

Here is a list of steps for using VS Code to connection to `"diamond"` Linux server:

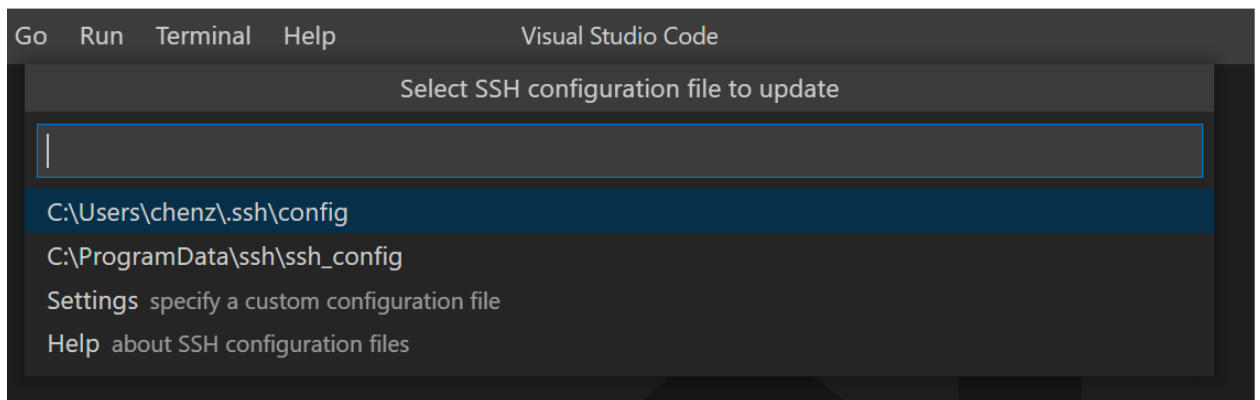
1. In VS Code, first install `"Remote - SSH"` extension.
2. In `"Remote Explorer"` tab, click `"Add New"` (`"+"` sign)



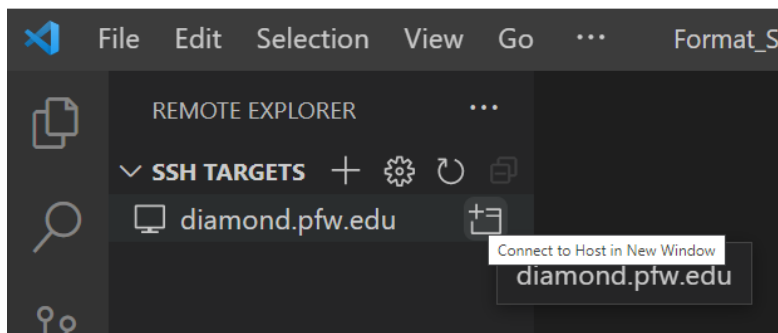
3. Then type “ssh your\_user\_name@diamond.pfw.edu” in the “Enter SSH Connection Command” box. Here “your\_user\_name” is your user name for the university account. For example,



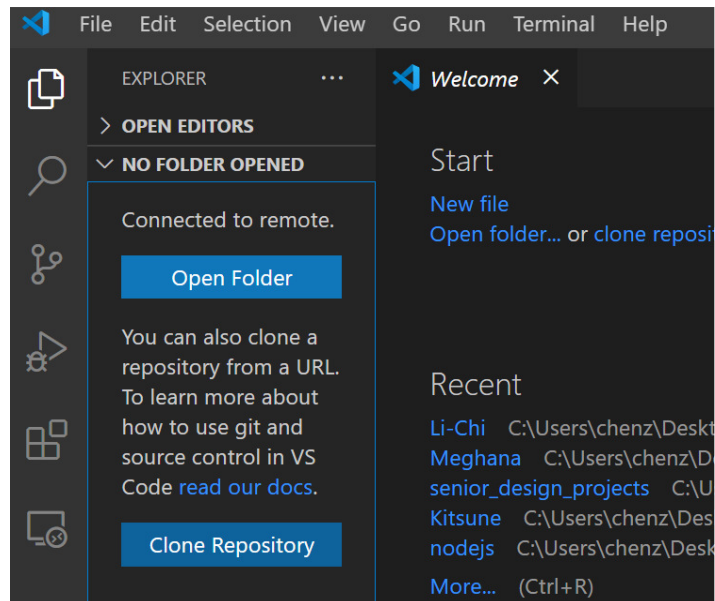
4. Select the first entry, e.g., “C:\Users\yourname\.ssh\config” file, for “Select SSH configuration file to update”



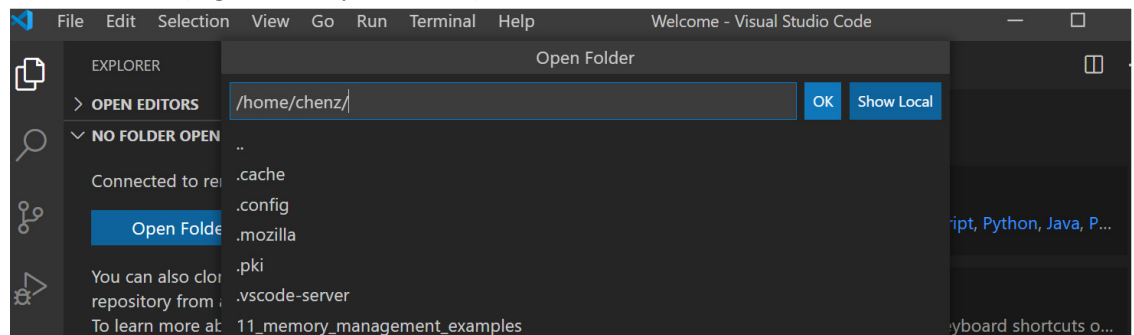
5. In “Remote Explorer” tab, click “Connect to Host in New Window”



6. Select “Linux” for “Select the platform of the remote host ...”
7. Choose “Continue” and enter your university password.
8. In “Explorer” tab, click “Open Folder”



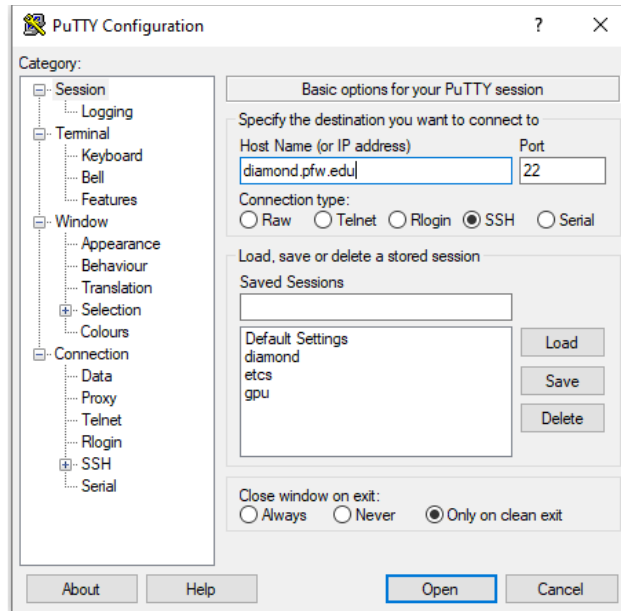
9. Use the default folder (e.g., /home/yourname/), and click “OK”.



10. Input your password one more time.
11. Then you log into “diamond” Linux server and can start use it in VS Code.
12. Next time to log in “diamond” Linux server, start from step 5.

## Use Putty to Connect to “Diamond” Linux Server

Download and install Putty from <https://www.putty.org/>. Then use host name “diamond.pfw.edu” to SSH to connect to “diamond” Linux server:



Sometimes, if you cannot use VS Code to connect to “diamond” Linux server, you need to use Putty to connect to “diamond” Linux server, and then run the following command:

```
$ rm -rf ./vscode-server/
```

After that, you should be able to use VS Code to connect to “diamond” Linux server again. The folder “vscode-server” generated by VS Code “Remote – SSH” extension can become large, so that it stops a user to use VS Code to remotely log in.

### Install OpenVPN (Off Campus)

**Off campus**, you can connect to “diamond” Linux server through OpenVPN and VS Code / Putty. That is, before using VS Code or Putty to connect to “diamond” Linux, the OpenVPN needs to be run **first**.

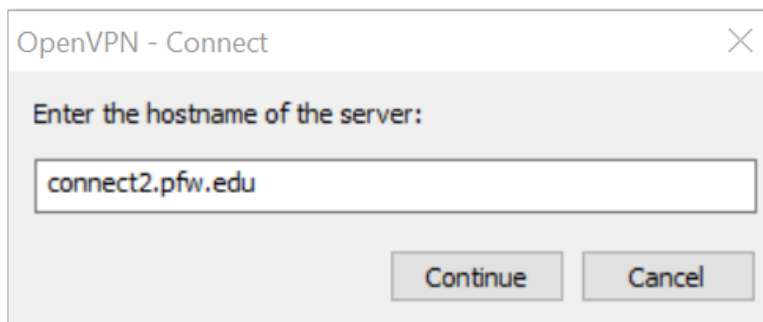
Specifically, here are instructions to install OpenVPN:

- Go to <https://etcsconnect.pfw.edu> (with Chrome), click on Advanced, proceed.
- Enter your university login information. Download your needed OpenVPN Connect for [your OS].
- Once it is installed, you will see an orange circle with an X on an antenna by the clock on the bottom right hand side.



Here are instructions to run OpenVPN:

- Right click on the orange circle for the application of OpenVPN, select “Connect...” and type “connect2.pfw.edu” in the box. Then enter your university password (or all login information) and click connect.



- You will get a notification if you have connected.