



CS197C

Intro to C

Nicolas Scarri - *February 1, 2017*

Contact Information

- Nicolas Scarrci
- nicolas.scarrci@umass.edu

Course Resources

- Course Website
 - <https://people.cs.umass.edu/~nscarrci/cs197c>

CS197C Course Homepage F2016

This course is intended to be a hands-on introduction to the C programming language and will teach related low-level systems concepts. Since it only runs for 6 weeks, the course is very fast paced, and a high level of programming maturity is required. CS197C covers introductory C and draws on concepts from CS121 and CS187.

All of the assignments in the course can be completed using only the information in the slides. I encourage you to ask questions in class or if preferred, by email. The point of this class is for you to learn, not for me to lecture, I am always glad to stop and explain a vexing concept.

Each lecture will last for about 30 minutes. The remainder of the time will be open for students to begin work on their weekly worksheet and their weekly project.

Office hours are by appointment only. I work in a shared space so it is *imperative* that you schedule visits ahead of time.

Overview

- Covered Topics
 - The GCC
 - Memory Management
 - Data Structures

- Expectations
 - Some Java or similar programming

Schedule

- 2/1 Welcome & Java Similarities
- 2/8 Basic GCC
- 2/15 Basic Memory
- 2/22 Advanced Memory
- 3/1 Data Structures
- 3/8 Advanced GCC

Administration

- This is a **1 credit PASS/FAIL** course.
- **5 Assignments, 5 worksheets, 1 final grade**

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			New Assignment 4:00 PM			
			New Worksheet 4:00 PM			
			Old Assignment Due 11:59 PM		Worksheet Due 11:59 PM	

Grading

**Assignment
Functionality**

5 points

50 Points Available

**Assignment
Effort**

5 points

25 Points Available

**Worksheet
Functionality**

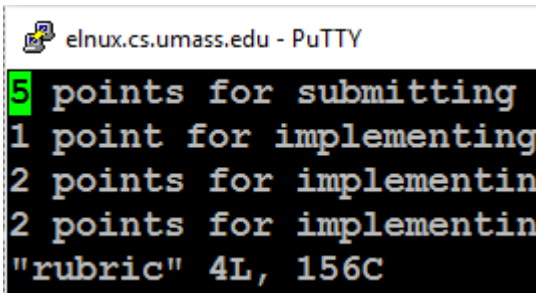
5 points

25 Points Available

65 points to pass

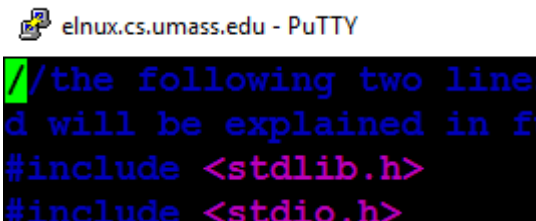
Assignment Workflow

1. `> vim rubric`



```
elnux.cs.umass.edu - PuTTY
5 points for submitting
1 point for implementing
2 points for implementin
2 points for implementin
"rubric" 4L, 156C
```

2. `> vim calc.c`



```
elnux.cs.umass.edu - PuTTY
//the following two line
d will be explained in f
#include <stdlib.h>
#include <stdio.h>
```


Assignment Workflow

3. Working ...

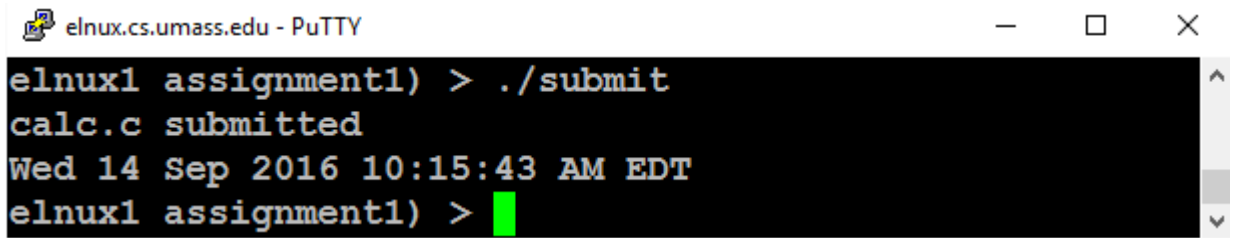
Assignment Workflow

4. `> ./test`



```
elinux.cs.umass.edu - PuTTY
2-24=-22
236*3=708
236-5=231
elinux1 solution) > 
```

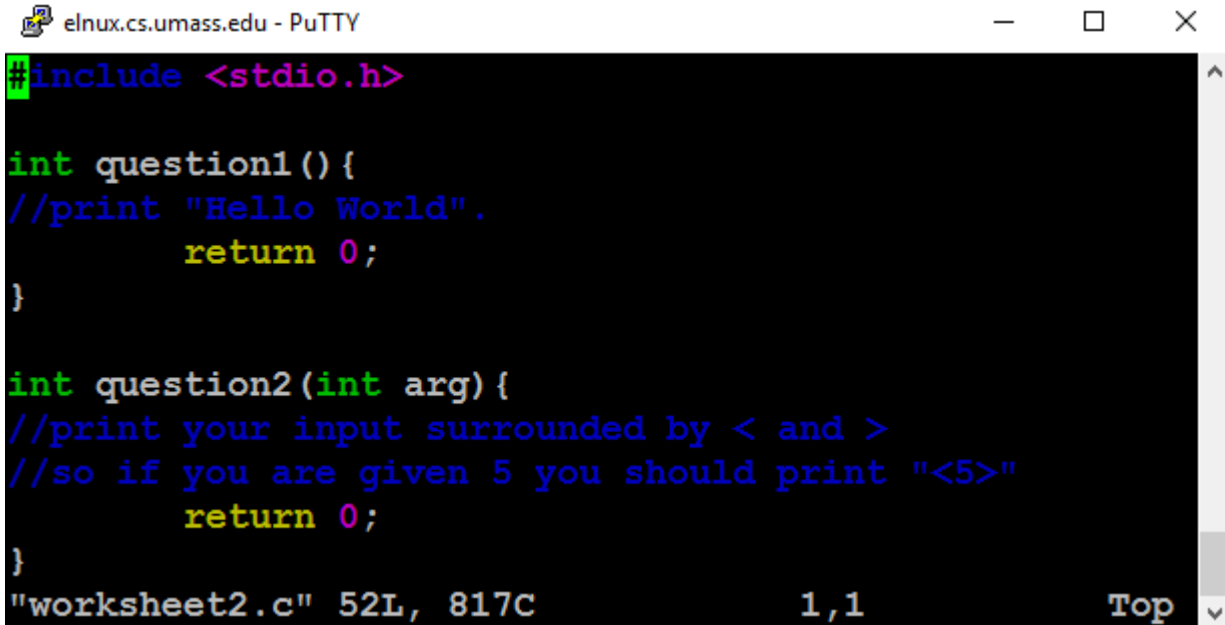
5. `> ./submit`



```
elinux.cs.umass.edu - PuTTY
elinux1 assignment1) > ./submit
calc.c submitted
Wed 14 Sep 2016 10:15:43 AM EDT
elinux1 assignment1) > 
```

Worksheet Workflow

1. > vim worksheet2.c



```
elinux.cs.umass.edu - PuTTY
#include <stdio.h>

int question1(){
//print "Hello World".
    return 0;
}

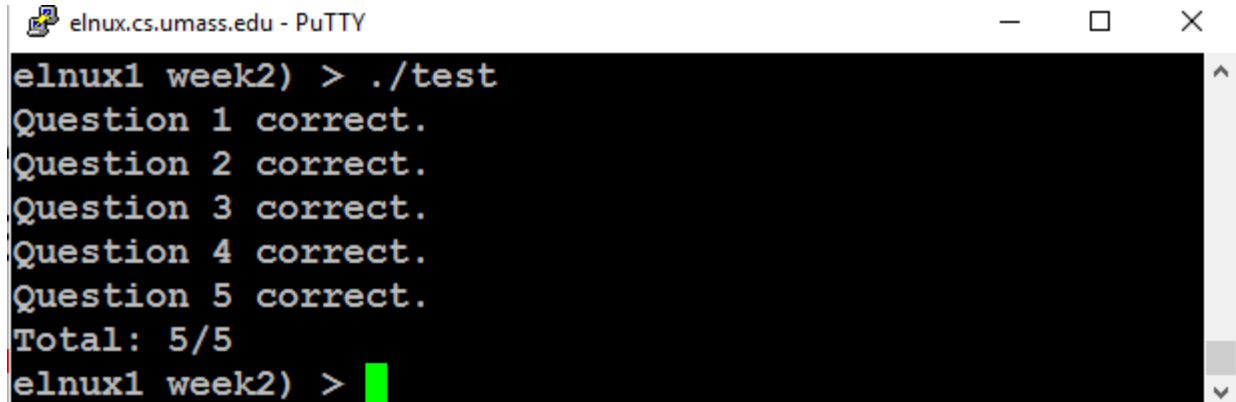
int question2(int arg){
//print your input surrounded by < and >
//so if you are given 5 you should print "<5>"
    return 0;
}
"worksheet2.c" 52L, 817C          1,1          Top
```

Worksheet Workflow

2. Working ...

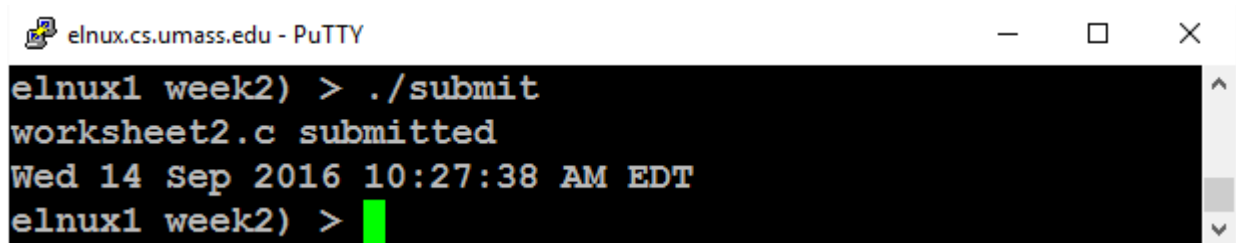
Worksheet Workflow

3. > ./test



```
elinux.cs.umass.edu - PuTTY
elinux1 week2) > ./test
Question 1 correct.
Question 2 correct.
Question 3 correct.
Question 4 correct.
Question 5 correct.
Total: 5/5
elinux1 week2) >
```

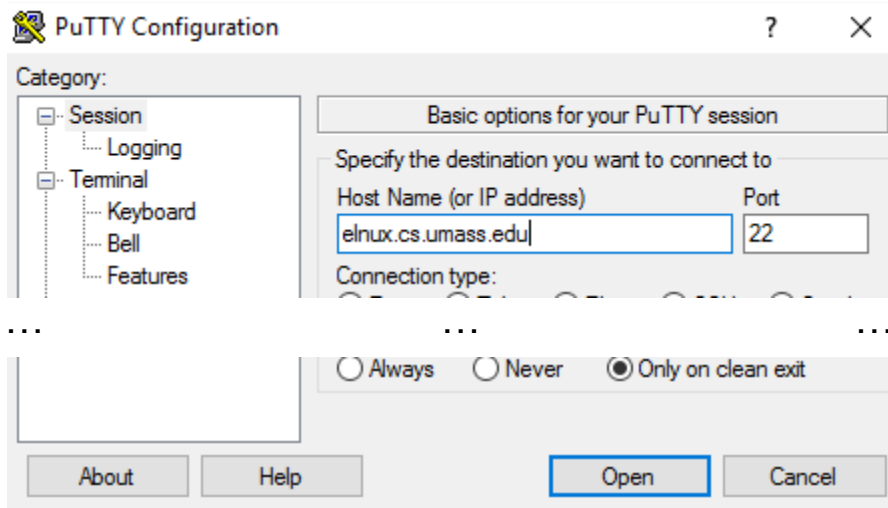
4. > ./submit



```
elinux.cs.umass.edu - PuTTY
elinux1 week2) > ./submit
worksheet2.c submitted
Wed 14 Sep 2016 10:27:38 AM EDT
elinux1 week2) >
```

Working Remotely – Windows

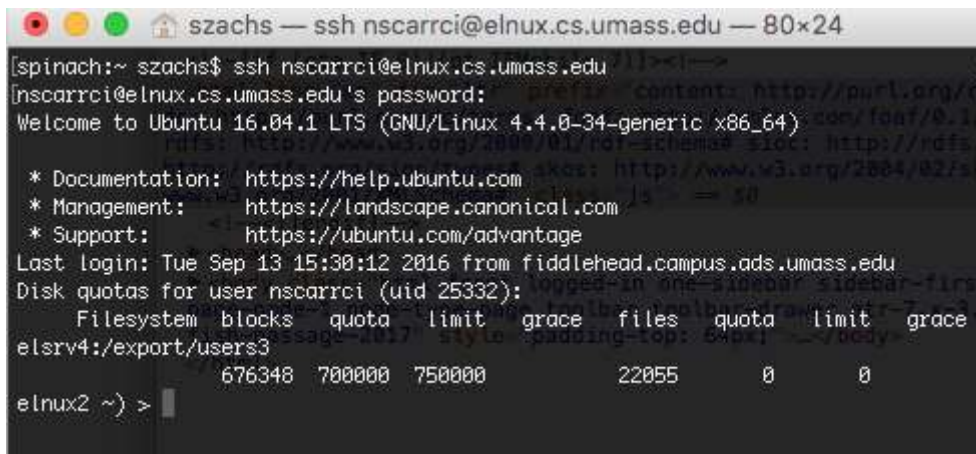
1. Download PuTTY
2. Host Name: elnux.cs.umass.edu



3. Click Open

Working Remotely – OSX & Linux

1. Open the terminal
2. **> ssh username@elinux.cs.umass.edu**



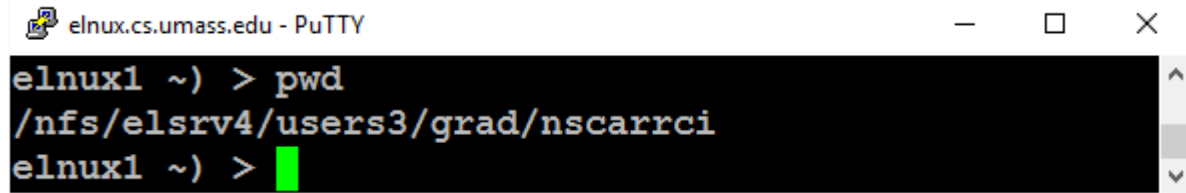
```
szachs — ssh nscarrci@elinux.cs.umass.edu — 80x24
[spinach:~ szachs$ ssh nscarrci@elinux.cs.umass.edu
[nscarrci@elinux.cs.umass.edu's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-34-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Tue Sep 13 15:30:12 2016 from fiddlehead.campus.ads.umass.edu
Disk quotas for user nscarrci (uid 25332):
   Filesystem  blocks quota  limit  grace  files  quota  limit  grace
  elsrv4:/export/users3
                676348 700000 750000
                22055      0      0

elinux2 ~) >
```

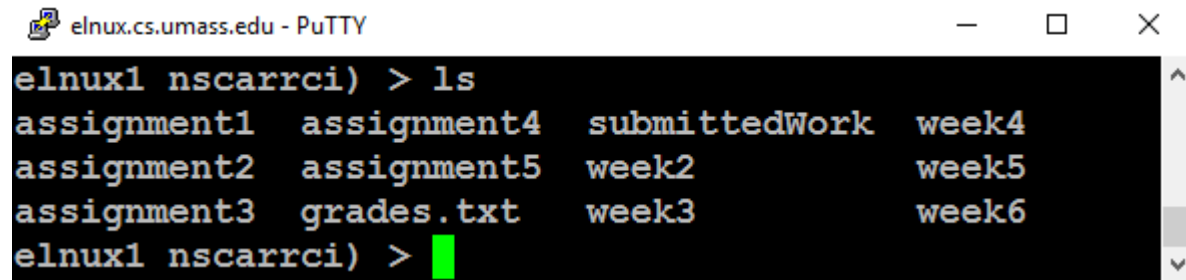
Required Unix – pwd & ls

- > pwd



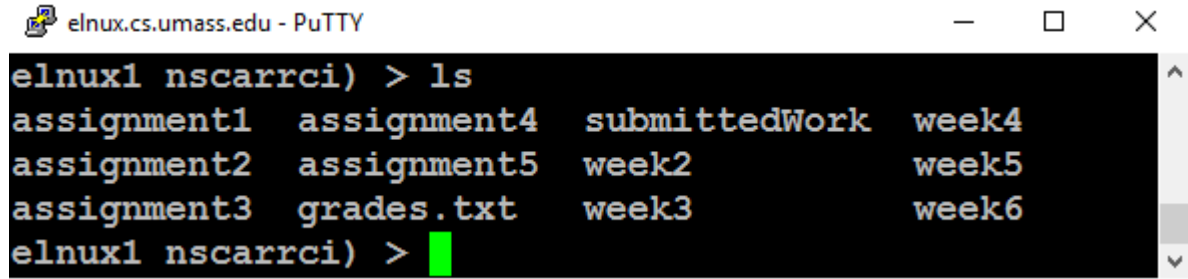
```
elinux.cs.umass.edu - PuTTY
elinux1 ~) > pwd
/nfs/el_srv4/users3/grad/nscarrci
elinux1 ~) >
```

- > ls



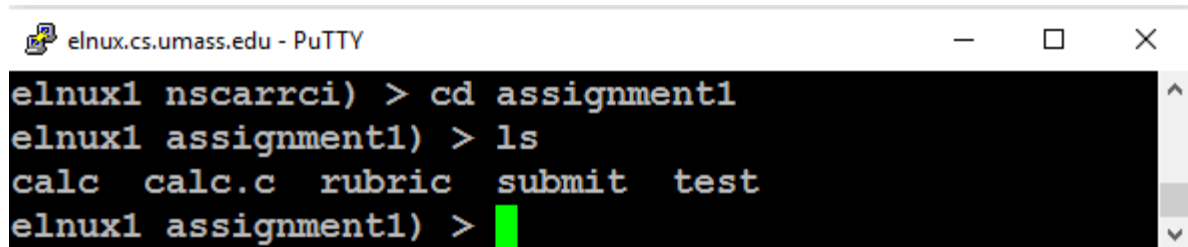
```
elinux.cs.umass.edu - PuTTY
elinux1 nscarrci) > ls
assignment1  assignment4  submittedWork  week4
assignment2  assignment5  week2          week5
assignment3  grades.txt   week3          week6
elinux1 nscarrci) >
```


Required Unix – cd



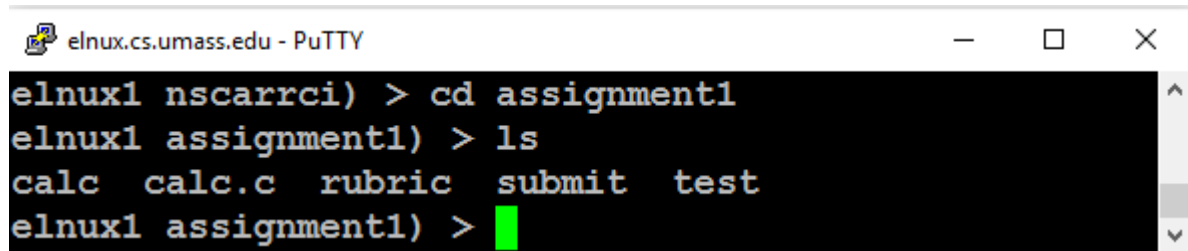
```
elnux1 nscarrci) > ls
assignment1  assignment4  submittedWork  week4
assignment2  assignment5  week2          week5
assignment3  grades.txt   week3          week6
elnux1 nscarrci) >
```

- > cd assignment1
> ls



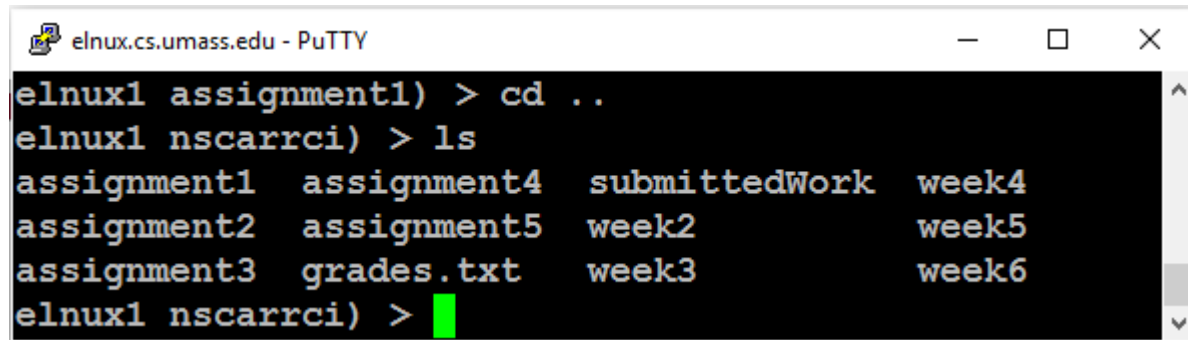
```
elnux1 nscarrci) > cd assignment1
elnux1 assignment1) > ls
calc  calc.c  rubric  submit  test
elnux1 assignment1) >
```

Required Unix – cd ..

A terminal window titled "elinux.cs.umass.edu - PuTTY" with standard window controls. The terminal shows a sequence of commands: "elinux1 nscarrci) > cd assignment1", "elinux1 assignment1) > ls", and the output "calc calc.c rubric submit test". The prompt "elinux1 assignment1) >" is followed by a green cursor.

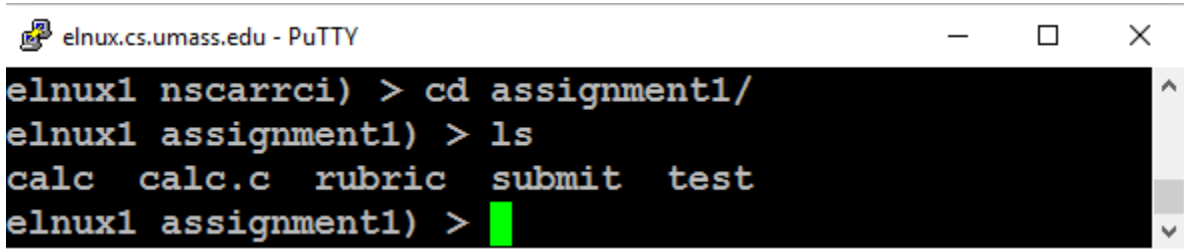
```
elinux1 nscarrci) > cd assignment1
elinux1 assignment1) > ls
calc calc.c rubric submit test
elinux1 assignment1) >
```

- > cd ..
- > ls

A terminal window titled "elinux.cs.umass.edu - PuTTY" with standard window controls. The terminal shows a sequence of commands: "elinux1 assignment1) > cd ..", "elinux1 nscarrci) > ls", and the output listing files and directories: "assignment1 assignment4 submittedWork week4", "assignment2 assignment5 week2 week5", "assignment3 grades.txt week3 week6". The prompt "elinux1 nscarrci) >" is followed by a green cursor.

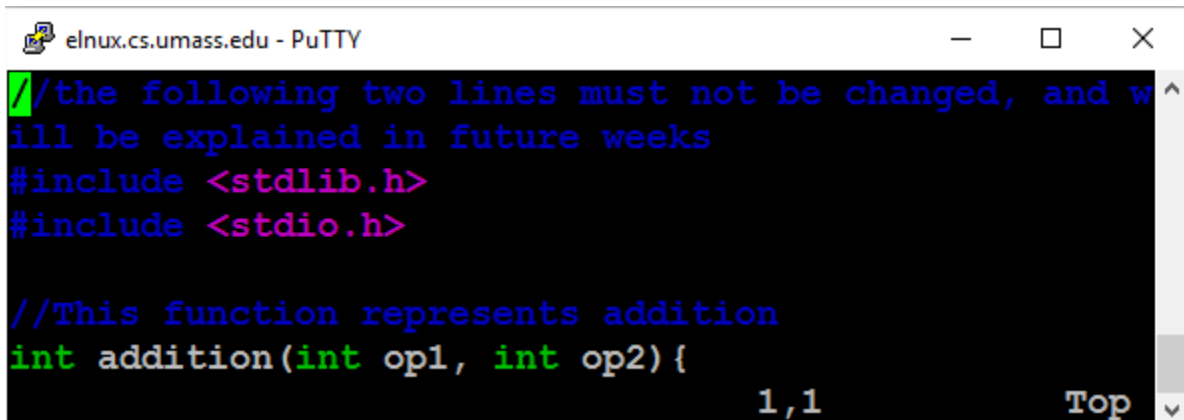
```
elinux1 assignment1) > cd ..
elinux1 nscarrci) > ls
assignment1 assignment4 submittedWork week4
assignment2 assignment5 week2 week5
assignment3 grades.txt week3 week6
elinux1 nscarrci) >
```

Required Unix – vim



```
elnuX1 nscarrci) > cd assignment1/  
elnuX1 assignment1) > ls  
calc calc.c rubric submit test  
elnuX1 assignment1) >
```

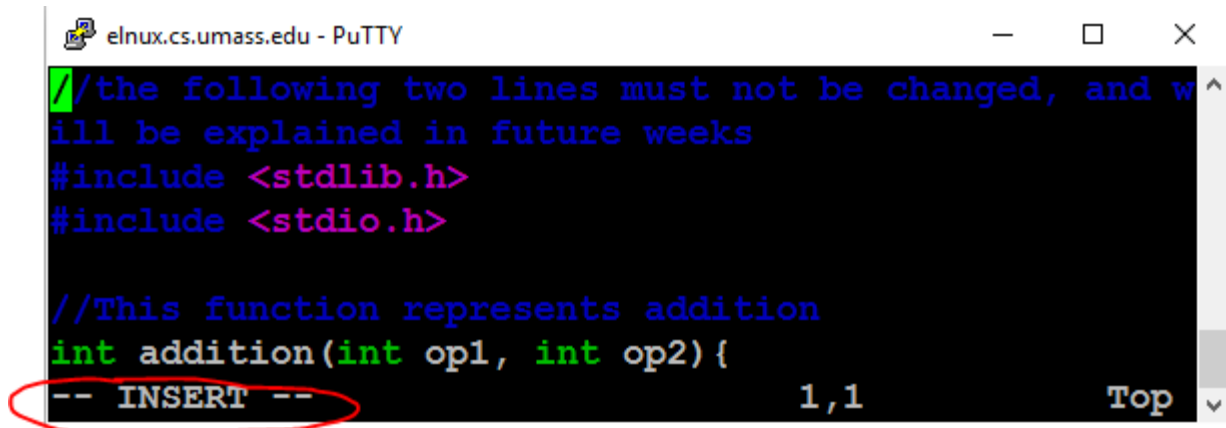
- > vim calc.c



```
//the following two lines must not be changed, and w  
ill be explained in future weeks  
#include <stdlib.h>  
#include <stdio.h>  
  
//This function represents addition  
int addition(int op1, int op2){  
1,1 Top
```

Required Unix – vim editing

- press i



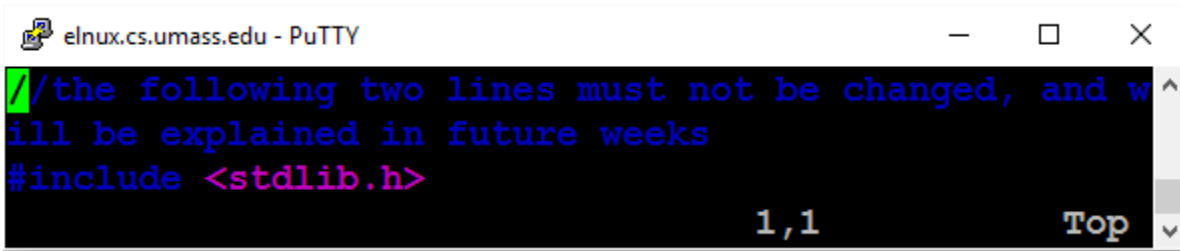
```
elinux.cs.umass.edu - PuTTY
//the following two lines must not be changed, and w
ill be explained in future weeks
#include <stdlib.h>
#include <stdio.h>

//This function represents addition
int addition(int op1, int op2){
-- INSERT --                               1,1           Top
```

- You may now type freely

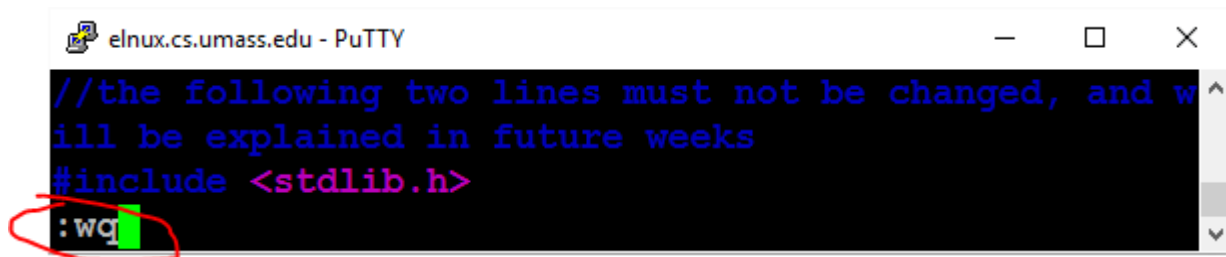
Required Unix – vim saving

- Press esc



```
elinux.cs.umass.edu - PuTTY
//the following two lines must not be changed, and will be explained in future weeks
#include <stdlib.h>
1,1 Top
```

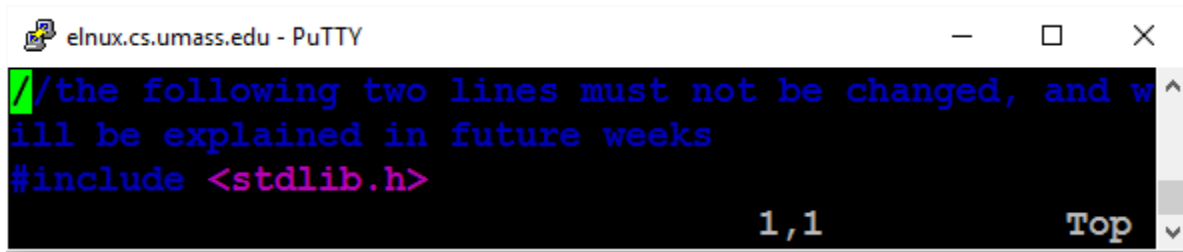
- Type :wq



```
elinux.cs.umass.edu - PuTTY
//the following two lines must not be changed, and will be explained in future weeks
#include <stdlib.h>
:wq
1,1 Top
```

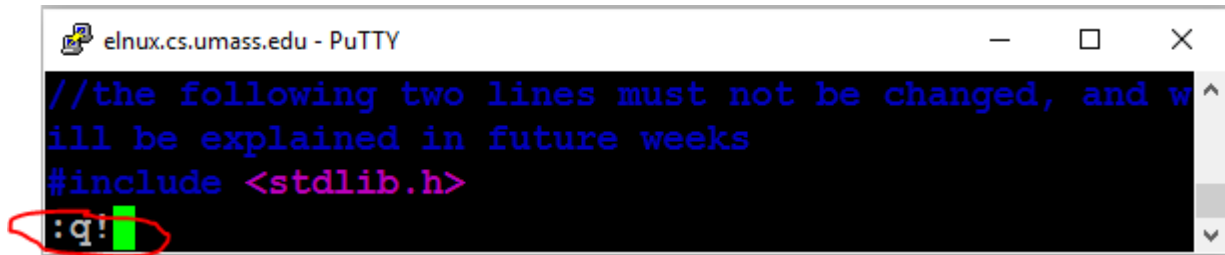
Required Unix – vim quitting

- Press esc



```
elinux.cs.umass.edu - PuTTY
//the following two lines must not be changed, and w ^
ill be explained in future weeks
#include <stdlib.h>
1,1 Top
```

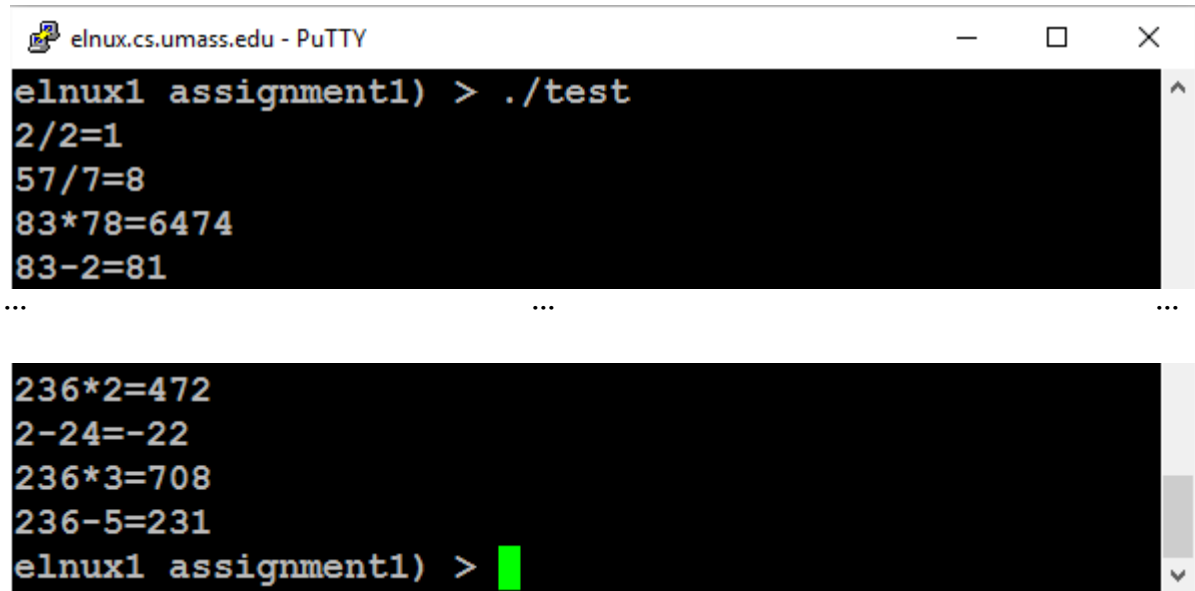
- Type :q!



```
elinux.cs.umass.edu - PuTTY
//the following two lines must not be changed, and w ^
ill be explained in future weeks
#include <stdlib.h>
:q! █
```

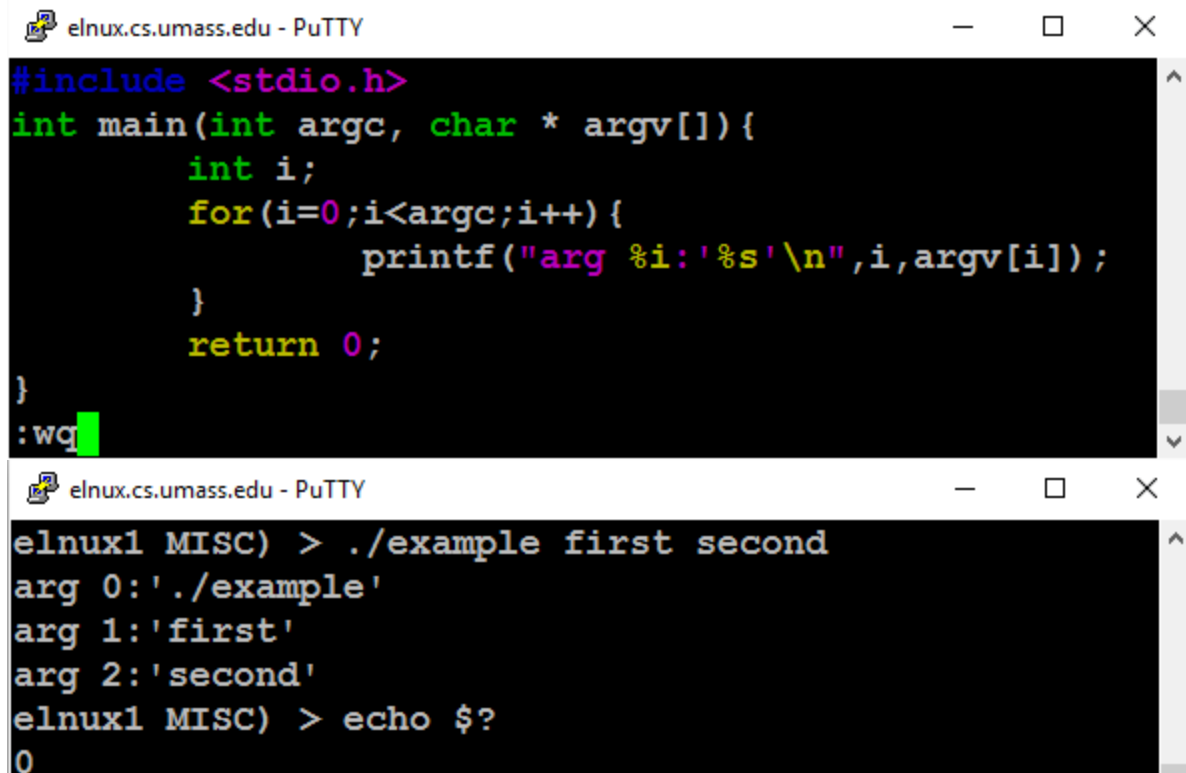
Assignment – expected results

- `> ./test`



```
elnux1 assignment1) > ./test
2/2=1
57/7=8
83*78=6474
83-2=81
...
236*2=472
2-24=-22
236*3=708
236-5=231
elnux1 assignment1) >
```

C – main function



The image shows two screenshots of a PuTTY terminal window. The top screenshot displays the source code for a C program named 'example.c'. The code includes the standard input/output header, defines the main function, and iterates through command-line arguments, printing each one. The bottom screenshot shows the program being compiled and executed. The user runs './example first second', and the output shows the program printing the arguments: './example', 'first', and 'second'. Finally, the user runs 'echo \$?' and the output is '0', indicating successful execution.

```
elinux.cs.umass.edu - PuTTY
#include <stdio.h>
int main(int argc, char * argv[]){
    int i;
    for(i=0;i<argc;i++){
        printf("arg %i: '%s'\n",i,argv[i]);
    }
    return 0;
}
:wq

elinux.cs.umass.edu - PuTTY
elinux1 MISC) > ./example first second
arg 0: './example'
arg 1: 'first'
arg 2: 'second'
elinux1 MISC) > echo $?
0
```


A photograph of a UMass Amherst campus scene, overlaid with a semi-transparent dark red filter. In the foreground, a row of tulips in shades of pink and yellow is in bloom. Behind them is a paved walkway and a grassy area. In the background, a large, modern building with horizontal bands of light and dark panels is visible, partially obscured by bare trees. The overall mood is serene and academic.

UMassAmherst
The Commonwealth's Flagship Campus