

Assignment 3 Specs

Wednesday, 10 November 2021

10:04 am

[Disclaimer]: This document is made out of goodwill, and while the author continues to try maintaining the most up-to-date information regarding ECE650 A3, **the author however is NOT responsible for any inaccurate information.** Please use this reference at your own risk, **You are strongly advised to refer to the actual posts and those posts answered by TAs.**

Tasks:

1. Modify the output format of Python script from Assignment 1 to match the input format of your C++ program from Assignment 2.
2. Modify your C++ program from Assignment 2 to output the input graph on standard output.
3. Write a program in C++ to generate random input for your Python script.
4. Write a driver program in C++ that uses Inter-Process Communication (IPC) to link the output of the random input generator to the input of the Python script, and the output of the Python script to the input of the C++ program from Assignment 2.

Additional Resources:

- The online book, “Operating Systems: Three Easy Pieces”, will be useful for this assignment and beyond. For this assignment, you might find Chapter 4 on Processes and Chapter 5 on Process API the most useful. The book is available online at <http://pages.cs.wisc.edu/~remzi/OSTEP/>.
- You might also want to consult “Advanced Linux Programming”, especially Chapter 3 on Processes and Chapter 5.4 on Pipes. The book is available from <https://github.com/MentorEmbedded/advancedlinuxprogramming/blob/gh-pages/alp-folder/advanced-linux-programming.pdf>

Input, Output, and Error

Input:

- Your program should take input from stdin
- As the example (Sample Run) below indicates, there are TWO kinds of input the

As the example (sample run) below indicates, there are **TWO** kinds of input the user provides:

- 1) One is via the command-line arguments, with switches such as -s and -n .
 - **This is done ONLY ONCE, when your program is first started.**
 - 2) The other type of input is the 's' command on stdin, which may be issued repeatedly, just as in Assignment 2.
 - For the 's' command, your program should output a shortest path.
- Input format errors will NOT be tested!
 - That is, the command-line arguments, whenever specified, will be formatted correctly, and
 - the s input will also be formatted correctly.
 - Possible testing scenarios (that generate errors):
 - command-line arguments may be omitted (see below for what to do in such cases):
 - Adopt Default Values
 - This two will cause error:
 - specify vertex IDs with s that do not exist, or
 - between whom a path does not exist.

Output:

- Your program should output to stdout.
- Your program should NOT generate any extraneous output:
- do not print out prompt strings such as “please enter input” and things like that.

Error:

- Errors should be output to stderr.
- Errors should always start with “Error:” followed by a brief description.

Termination:

- **ALL your processes** should terminate gracefully (and quietly) once you see EOF at stdin.

Sample Run

You should name the driver's executable ece650-a3. In the following, we assume that “\$” is the shell command-prompt

is the shell command-prompt.

```
$ cd build //make executable (line1/3)
$ cmake ../ //make executable (line2/3)
$ make install //make executable (line3/3)
$ cd ./run/bin
```

The following command runs the executable with inputs to your rgen (Piazza @588):

```
$ ./ece650-a3 -s 5 -n 4 -l 5 //run the driver program (ece650-a3); This is done
ONCE ONLY
```

#The blow graph is printed as an output from a2 c++ script as it reads an input from the python file. (Piazza @588)

```
V 8 //This is an output (print from A2): the no. of vertices (piazza: 625)
E {<1,3>,<1,4>,<1,5>,<2,4>,<5,8>,<6,3>,<6,7>} // This is an output (print from A2)
```

```
s 3 5 //This is an input (provided by user) via stdin;
//Can be issued repeatedly
//Render possible errors
```

```
3-1-5 //This is an output
```

Marking

- Does not compile/make/crashes: automatic 0
- Your program runs, awaits input and does not crash on input: + 20
- Run in default mode: + 15
- Error check arguments: + 5
- Test functionality: + 20
- Test rgen: + 20
- Replace rgen: + 20

CMake

As discussed below under “Submission Instructions”, you should:

- use a CMakeLists.txt file to build your project.
- We will build your project using the following sequence:
cd a3 && mkdir build && cd build && cmake ../ && make install
- If your code is not compiled from scratch (i.e., from the C++ sources), you get an automatic 0.

- Note that we are using make install instead of make. The install target instructs make to copy all of the binaries (both Python and C++) into directory ./run/bin. This ensures that all the executable files (including the Python program) are in the same directory.

Submission Instructions

- You should place all your files in the a3 directory in your GitLab repository. The directory should contain:
 - All your C++ and Python source-code files.
 - IMPORTANT NOTE: the executable for your random generator must be named **rgen**. The reason is that part of our testing will replace your generator with ours.
 - A CMakeLists.txt, that *builds all of the C++ executables: rgen, ece650-a2, and ece650-a3.*
 - A file user.yml that includes your name, WatIAM, and student number. See README.md for any additional information.
- The submitted files should be in the a3 directory of the master branch of your repository.
- Two file names that are important:
 - One is **your main executable**. This must be named ece650-a3. Our marking script will simply try and run this after building your project.
 - The other executable file whose name is important is **your random input generator**. The executable for this must be named **rgen**. The reason is that **for some of our tests, we will replace your rgen with ours.**

Things to be done

1. Python script

- Edit your Python script from Assignment 1 so that **its output which is the specification of the graph is formatted to match the input format of your C++ program from Assignment 2.**
 - That is, your Python script provides command-line input to the C++ program.
 - The way to do this is to simply
 - map each vertex to an index in [1, n] (where n is the number of vertices), and
 - rename your edges accordingly.
- Also, **the ONLY output that your Python script should produce to**

stdout is in response to “gg”, for which it outputs the specification of the graph (i.e., V and E).

- [???] **What's the input??? Where does the input come from?**
 - Input to Python program come from the randomly generated street specification
- [???] Which process input 'gg'???
- **[Important!!!]:** Python script do NOT actually PRINT the specs after 'gg'. Printing the Specs is the job of Cpp program from A2!
- Error output should go to stderr.
 - [???] What kind of error?

2. Cpp program from Assignment 2 (in c++)

- Edit it to print the graph that it has read before accepting an s command.
 - **This is necessary so that you know what graph has been produced and which vertices are available.**

3. Random input generator (rgen) (in c++)

- Purpose:
rgen (random input generator) should: generate random street specifications as input for your Python script.
- You MUST use /dev/urandom as the source of your random data. Also, see under “Submission Instructions” how your executable for the random generator must be named.
- **Input:**
It takes four command-line arguments. (All are optional!!!).
 - [???] Where does the four command-line arguments come from???
 - From command-line argument via driver to rgen.
 - If no command-line argument is specified, then driver pass default value
 - -s k — where k is an integer ≥ 2 .
 - The number of streets should be a random integer in $[2, k]$.
 - If this option is NOT specified, you should use a default of $k = 10$; that is, the number of streets should be a random integer in $[2, 10]$.
 - -n k — where k is an integer ≥ 1 .
 - The number of line-segments in each street should be a random integer in $[1, k]$.
 - **Default: $k = 5$.** (when this option is NOT specified)

- -l k — where k is an integer ≥ 5 .
 - Your process (regen) should wait a RANDOM NUMBER w seconds, where w is in [5, k] before generating the next (random) input.
 - Default: k = 5.
- -c k — where k is an integer ≥ 1 .
 - Your process (regen) should generate (x, y) coordinates such that every x and y value is in the range [-k, k].
 - E.g. , if k = 15, all of your coordinate values should be integers between -15 and 15.
 - Default: k = 20.
- **Output:**
 - Your program SHOULD generate a specification of streets in the format that your Python script expects (see Assignment 1).
 - You can name the streets whatever you want.
 - You should ensure that your input (to Python Script; a.k.a. the output of regen) does not have errors.
 - ◆ For example, if you generate a line-segment that overlaps with a line-segment (across all streets) generated earlier, you should regenerate that line-segment.
 - ◆ Similarly, you should NOT have any zero-length line segments.
- Also, note that your random generator could go into an infinite loop looking for a valid specification (never finding the valid specification).
 - You should disallow this from happening by limiting the number of tries.
 - That is, after A number of attempts without successfully finding valid specification, regen should exit() with an error message reported on stderr.
 - ◆ A reasonable A to adopt may be 25.
 - Whatever A you adopt, your error message should mention it. That is, your error message should be something like, “Error: failed to generate valid input for 25 simultaneous attempts”. (Print to stdout!!!)
- Before adding a new street graph specification to your Python script, your generator should issue “rm” commands to your Python script to remove ALL previous streets and replace them with the new street specification.
- After generating the input, the generator must issue the “gg” command.

- Thus, a typical interaction of the random generator is as follows:
 1. issue enough rm commands to clear any existing street database;
 2. issue add commands to add new streets satisfying the specification;
 3. issue a gg command;
 4. wait for specified number of seconds and repeat.

4. Driver (*ece650-a3*) (in c++)

- Your driver program has the overall control.
- You have at least three programs that run concurrently:
 - (1) the random generator,
 - (2) your Python script from Assignment 1, and,
 - (3) your program from Assignment 2 that takes a graph-specification as input (from Python script) and computes shortest paths.
- Your driver program should
 - fork() two processes and exec() two of those programs, and
 - then exec() the third (so it turns into a process that corresponds to the third program).
- It should set up all IPC appropriately beforehand.
- It should send:
 - normal output to stdout,
 - error output to stderr, and
 - take input from stdin.
- The ONLY input it takes
 - “s” commands, that ask for a shortest path between vertices.

Gotcha warning: note that I say “you have at least three programs that run concurrently.” You may need more than three that run concurrently. (Why?)

Sample code

- The skeleton repository for the assignment contains some sample code.
 - You should replace ALL sample code with your own code.
 - Including replacing the Python code with your solution from Assignment 1, and C++ code from your solution to Assignment 2.

FAQs (based on TAs answer on Piazza):

- Will A2 grade be released soon to ensure we have time to fix the bugs in A2?
 - Yes. very soon (this response was dated: 10-Nov)

- [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=580>
- Again, yes (response dated on 14-Nov)
- [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=613>
- Must we use stderr to output error message?
 - No
 - You can use `std::cout` or `std::cerr`
- How to add python script in CMakeList?
 - A1.py should be added like this:
 - `install(PROGRAMS ece650-a1.py DESTINATION bin)`
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=579>
- Do we have to fix the bugs in previous assignments?
 - Yes
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=580>
- How do rgen and ece650-a3 communicate?
 - Use a pipe for the communication
 - Note: Rgen only generates inputs to your code for A1
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=583>
- Which process should handle the checking for overlapping street?
 - Rgen should handle it.
 - Make sure the input to A1 (Python Script) contain no overlapping street.
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=585>
- Which program prints the specs for graph?
 - A2
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=588>
- What does it mean by "it should exit() with an error message reported on stdeer"?
 - Answer in one of the questions below
 - [Unanswered]: https://piazza.com/class/ktachtjr9z2c6?cid=584_f4
- When will the whole program (driver) stop?
 - The only true stop is when the driver receives an EOF.
 - https://piazza.com/class/ktachtjr9z2c6?cid=610_f2

- https://piazza.com/class/ktachtjr9z2c6?cid=606_f1
- https://piazza.com/class/ktachtjr9z2c6?cid=606_f1

- How does Python pass the graph specs to A2? Why does python script has to output to stdout in response to "gg"? Does it mean we have to print it?
 - [Unanswered]: https://piazza.com/class/ktachtjr9z2c6?cid=588_f1
- Is it ok to make a "remove all" command that removes all the streets instead of removing all streets individually? (https://piazza.com/class/ktachtjr9z2c6?cid=589_f3)
 - Yes
 - You can write a new function or simply use rm command to remove all streets
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=611>
- **How to use dev/urandom?**
 - Make sure you are on a Linux machine
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=590>
 - Useful reference: <https://www.cs.yale.edu/homes/aspnes/pinewiki/C%282f%29Randomization.html>
 - To understand dev/urandom: https://en.cppreference.com/w/cpp/numeric/random/random_device/random_device
- **Is a street allowed to have intersection with itself?**
 - No. Your rgen should make sure to reject it in this case.
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=592>
- **Is a street allowed to have self-loop?**
 - No. Your rgen should make sure to reject it in this case.
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=592>
- **Can two street have overlaps?**
 - No. Your rgen should make sure to reject it in this case.
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=592>
- **Is it ok for two streets to have a common point?**
 - Yes, this is valid.
 - F σ

○ L.S.

- Add "A" (0,0) (0,3)
- Add "B" (0,3) (3,3)

○ [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=592>

- How to write a make file?

- Something like this should work

create the main executable

add additional .cpp files if needed

add_executable(ece650-a2 ece650-a2.cpp)

add_executable(ece650-a3 ece650-a3.cpp)

add_executable(rgen rgen.cpp)

○ [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=593>

- **What is the max. "-s" can take in rgen?**

- No large number will be asked during testing. Handling 1000 vertices should be fine.

○ [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=596>

- **What is the maximum no. of streets number?**

- There won't be very large (-s) numbers during testing
- If it works for 100 or 200, then it should be fine

○ [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=641>

- **How to assign street name (from rgen to Python)?**

○ [Ref]: https://piazza.com/class/ktachtjr9z2c6?cid=596_f1

- **How many times do you have to generate random numbers in rgen:**

- Wait time is generated once per run time of rgen
- Coordinates are generated randomly per street
 - Each street doesn't have to have the same number of line segment (get a random number for the number of line segment for each street)
- Number of tries is determined experimentally (ensure that Rgen produce a valid specification most times)

○ [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=598>

- How to control process order?

- See this example: https://git.uwaterloo.ca/ece650-f2021/cnn/-/blob/master/using_nine_cnn

[f2021/cpp/-/blob/master/using_pipe.cpp](https://git.uwaterloo.ca/ece650-f2021/cpp/-/blob/master/using_pipe.cpp)

- In particular, these lines are useful:

```
waitpid(-1, NULL, 0);  
//When we send EOF, terminate all process  
for (pid_t k : kids) {  
    int status;  
    kill(k, SIGTERM);  
    waitpid(k, &status, 0);  
}
```
- [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=599>

- **How to build pipe between Python and C++**

- In your terminal
./rgen | python3 a1ece650.py | a2ece650
To write in c++, check the sample code given in the git.
https://git.uwaterloo.ca/ece650-f2021/cpp/-/blob/master/using_pipe.cpp
- Use dup2 and STDOUT_FILENO to connect between python and C++

- How to pass keyboard command "s 1 2" to A2 when stdin is already replaced by the pipe in A2?
 - Answer: https://piazza.com/class/ktachtjrz9z2c6?cid=604_f2
 - [Ref]: https://piazza.com/class/ktachtjrz9z2c6?cid=601_f1

- **What if a point lies on a line segment that already exists?**

- Regenerate only that point.
- [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=602>

- **Can we reorder the points (to get line segments)?**

- You shouldn't reorder the points
- [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=602>

- **How does random, say 5 (default) seconds work?**

- You need to wait for s input within 5 seconds. After 5 seconds, a new graph should be printed.
- You can change the input to 10 seconds for testing purposes.

- How to regain standard input control from A2 (When A1 and A2 are connected through a pipe)?
 - You need another pipe - output of a1 to input of a2.

- a2 executable will print the shortest path and accept inputs "s v1 v2" command.
 - See for an alternative solution also: https://piazza.com/class/ktachtjr9z2c6?cid=604_f2
- What's the suggested way to test out if my pipes are working as intended?
 - [Suggested by Kelvin Fan]: You should use the pipe console command to debug your system first. Once your three programs are working as intended for the console pipe command. You should write the driver that simulate the console command: `rgen | python3 a1ece650.py | a2ece650`
 - https://piazza.com/class/ktachtjr9z2c6?cid=604_f1
- **Are we suppose to use sleep() in rgen?**
 - Yes, and indeed it will be inside rgen
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=605>
 - [Ref]: https://piazza.com/class/ktachtjr9z2c6?cid=607_f1
- **Which process handles interval (i.e. wait for 5 seconds before regenerating a new graph specs)?**
 - Rgen
 - [Ref]: https://piazza.com/class/ktachtjr9z2c6?cid=607_f1
- **After 25 attempts , which process prints the "Error: ... after 25 attempts?" How do we print it?**
 - Rgen
 - Print errors using `std::out`
 - All ERRORS generated by rgen should be printed using `std::out`
 - Do NOT send those errors to drivers!!!
 - [Ref]: https://piazza.com/class/ktachtjr9z2c6?cid=607_f1
- Which process validates and handle input range for switches (-s, -n, -l, -c)?
 - Driver process
 - Handle it before passing the parameters (-s, -n, -l, -c) to rgen.
 - [Ref]: https://piazza.com/class/ktachtjr9z2c6?cid=607_f1
- Do we assume that we get valid range for switches (-s, -n, -l, -c)???
 - No
 - Only the input format is guaranteed to be correct

- Only the input format is guaranteed to be correct
 - You need to validate and handle potential errors (values out of bounds) regarding the range of switches before passing those parameters to rgen.
 - If there is an error in the range of switches, print it to cerr (in Driver process) and do nothing but wait for EOF to exit
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=607> f3
- What if the output rate to console is very slow (say the default wait time is 5 seconds in rgen but it takes about a minute to print to console? And a2 print one output in one update?)
 - [Suggested by classmate]: You need to flush the buffer (flush stdout at the beginning of main & for each process)
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=605> f1
- [To be verified] Errors for A3 and rgen:
 - <https://piazza.com/class/ktachtjr9z2c6?cid=607>
- What happens when we have streets whose line segments don't intersect? What do we print?
 - Print the following:
V 0
E {}
 - <https://piazza.com/class/ktachtjr9z2c6?cid=608>
- For the above case, do we print any additional error message as well?
 - No
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=608> f1
- What does 'gg' do in A1?
 - 'gg' is not a keyboard input
 - 'gg' should be a command to pass graph from a1ece650.py to a2ece650 executable
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=611>
- Does the switch order matter (-s -n -l -c)?
 - No, they are arguments. It doesn't necessarily have to be in any order
 - <https://www.geeksforgeeks.org/command-line-arguments-in-c-cpp/>
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=612>

- What should I do when python scripts fail to send output to A2?
 - Add the following: when finishing each iteration of printing
 - `sys.stdout.flush()`
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=620> f2
- **How to print error messages in rgen?**
 - Cerr
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=621>
- **What is the least no. of coordinates a street should have:**
 - 2 coordinates, i.e., 1 line segment
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=622>
- What does Python script output? Coordinates or vertices?
 - Vertices (as defined in A1; refer to A1 specs for more detail)
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=625>
- **Can you explain the rgen procedure again?**
 - rgen should do the following as mentioned in assignment3.pdf
 - generate (one) Street->validate(overlapping etc.)->output (add this one) street to a1
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=633>
- What if s is enter after the specified wait time and a new set of spec has already been generated?
 - we expect to enter the s command in the delay time we mentioned while passing it to the driver.
 - it will print the shortest path for the set of Graph which was generated just before executing the wait/sleep.
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=637>
- How do I send the arguments which is parsed by the driver to rgen via execl/execv command?
 - `execv("ece650-a1.py", argv);` where argv are arguments
 - [Ref]: <https://piazza.com/class/ktachtjr9z2c6?cid=644>
- **Can segments from the same street intersect?**
 - No

- ...

- [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=645>
- What if there is a communication issue between A1 (python script) and A2 (c++)?
 - Set the flush flag to true when printing in Python:
 - `print(... , flush=True)`
 - Otherwise the OS will queue up all the prints and not flush the buffer immediately
 - [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=647>
 - <https://stackoverflow.com/questions/230751/how-can-i-flush-the-output-of-the-print-function-unbuffer-python-output>
- I have trouble passing arguments (-s, -n, -l, -c) from driver to rgen
 - Try: `execv("ece650-a1.py", argv)`; where argv are arguments.
 - <https://piazza.com/class/ktachtjrz9z2c6?cid=649>
- How to use execv to execute python script?
 - [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=650>
- **How does the retry work? Do I increment the retry counter after failing to generate a street completely or do I increment the retry counter when one of the coordinates form self-intersection with the street itself while generating this street?**
 - Rgen logic is built on your assumption
 - [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=651>
 - [Proposed by Akib Islam]: https://piazza.com/class/ktachtjrz9z2c6?cid=651_f1
- Output (from Python to A2) taking too long?
 - Not endorsed by instructor but the student answer tallies with some of answers to similar questions.
 - [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=655>
- Can I use additional mechanism for message passing between processes in addition to pipe?
 - Yes. As long as it runs in ece machines
 - [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=658>

- **Which exit code should I use in case of error in rgen?**
 - You can use a non-zero exit code
 - [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=659>
- How do I pass the output of rgen to assignment 1 python script???
 - You need to pass the STDOUT of rgen to A1 in driver class
 - [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=660>
- When should the input graph be output? E.g. should V 8 be output as soon as it is received (by A2) or should we wait for a valid edges specification and output both at once?
 - V 8 should be output as soon as it is received and then wait for valid E input (like in assignment2.pdf)
 - [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=661>
- Is it ok that python produce zero vertex (no intersection)?
 - It's okay
 - [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=662>
- **What happens when the number of errors reached maximum number allowed in rgen?**
 - Print an error to stderr (and exit()) then wait for the driver for new set of commands/EOF
- **Does rgen use the same command-line arguments (-s -n -l -c) when generating a new set of street graph specification?**
 - Yes
 - [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=643>
- Is it ok for two streets to have a common point?
 - Yes [instructor endorsed]
 - E.g.
 - Add "A" (0,0) (0,3)
 - Add "B" (0,3) (3,3)
 - [Ref]: <https://piazza.com/class/ktachtjrz9z2c6?cid=592>

List of unanswered questions:

- All questions flagged unanswered (dated 20-Nov).

- All questions tagged unanswered (dated 20-Nov).
 - o Any posts in Piazza after post@664