20F-COMSCI97-1 Midterm

NEVIN LIANG

TOTAL POINTS

65 / 100

QUESTION 1

32 pts

1.1 1a **6 / 12**

✓ **- 6 pts** Click here to replace this description.

- incorrect adhoc shuf implementation
- inefficient because of loops
- no final output shuf
- refer solutions

1.2 1b **8 / 10**

✓ **- 2 pts** Click here to replace this description.

merely invokes myspell; doesn't verify output

1.3 1c **0 / 10**

✓ **- 10 pts** Click here to replace this description.

QUESTION 2

2 2 **14 / 14**

✓ **- 0 pts** Correct

QUESTION 3

3 18 pts

3.1 3a **6 / 6**

✓ **- 0 pts** Correct

3.2 3b **0 / 6**

✓ **- 6 pts** significantly incomplete / incorrect; refer solutions

3.3 3c **2 / 6**

✓ **- 4 pts** major issues; refer solutions

QUESTION 4

4 4 **9 / 9**

✓ **- 0 pts** Correct

QUESTION 5

5 12 pts

5.1 5a **4 / 8**

+ **0 pts** Blank

+ **2 pts** Wrote a very simple explanation or most of the detail is incorrect or missing a lot of stuff

✓ **+ 4 pts Explanation is missing a good amount of detail or contains wrong details**

+ **6 pts** Explanation is missing some detail or contains some wrong details

+ **8 pts** Full explanation

5.2 5b **4 / 4**

✓ **- 0 pts** Correct

QUESTION 6

6 6 **12 / 15**

+ **15 pts** Complete solution

✓ **+ 12 pts missing minor points / details**

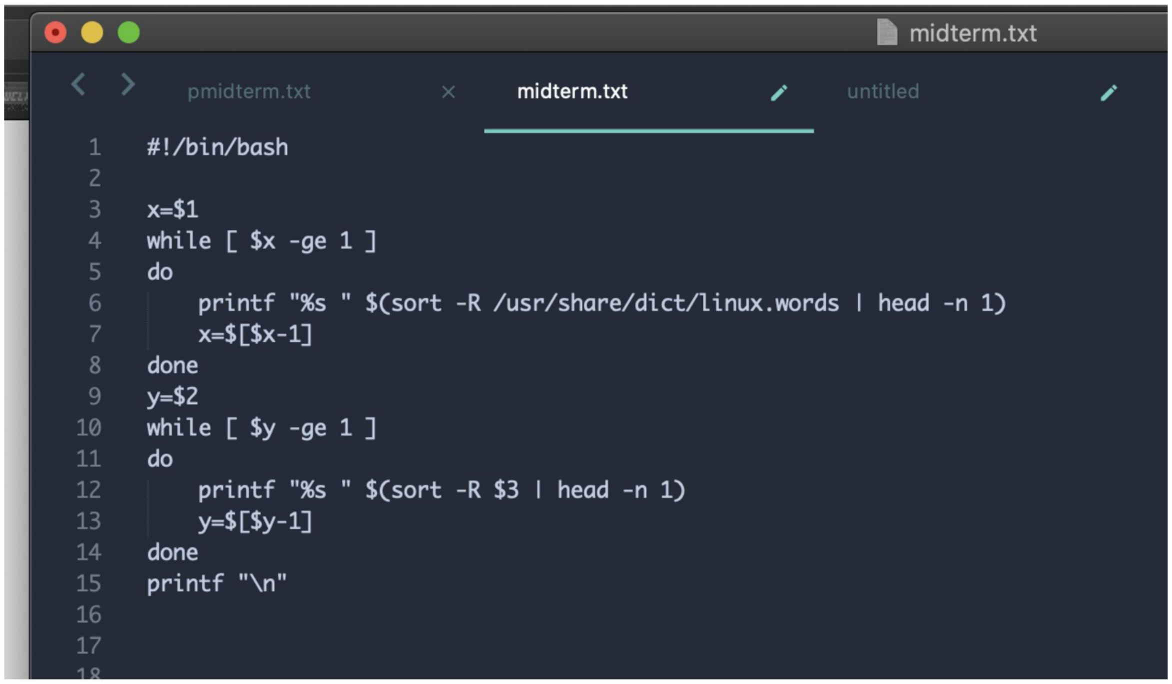
+ **9 pts** missing major points / details

+ **6 pts** only minor overlap with expected points / details; refer solutions

+ **3 pts** barely touches expected points / details; refer solutions

+ **0 pts** significantly incomplete / incorrect; refer solutons

(Put your answer to 1a here.)



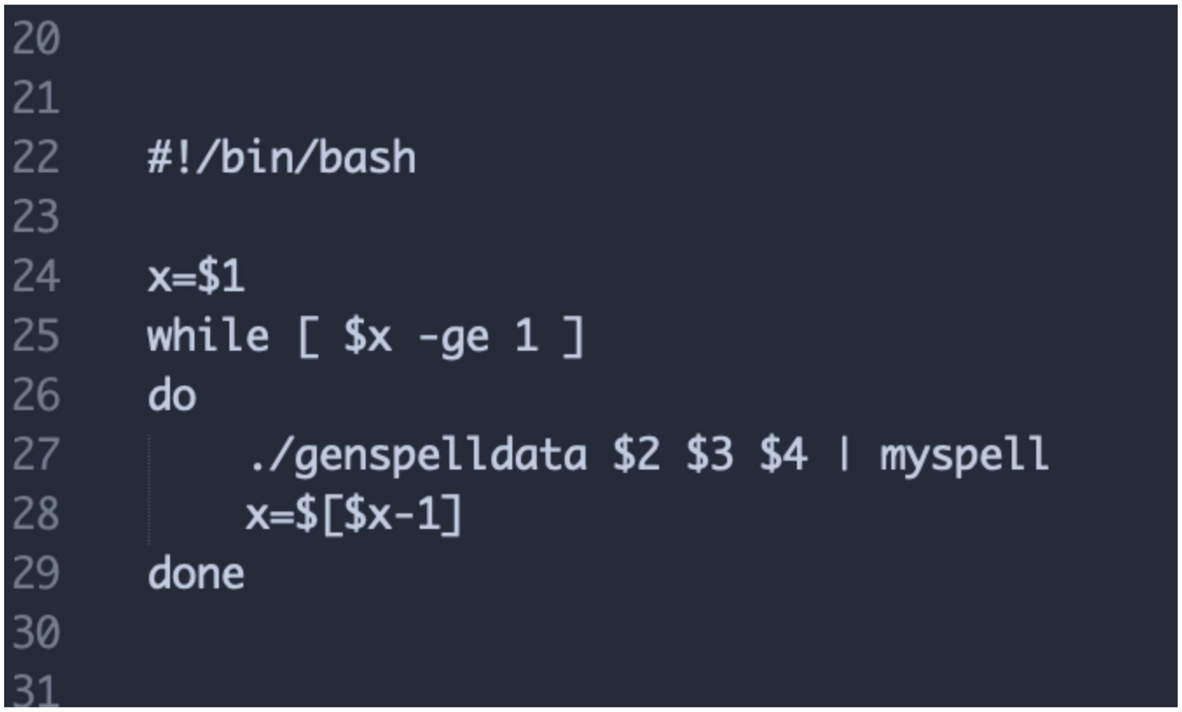
```
#!/bin/bash

x=$1
while [ $x -ge 1 ]
do
    printf "%s " $(sort -R /usr/share/dict/linux.words | head -n 1)
    x=$[$x-1]
done
y=$2
while [ $y -ge 1 ]
do
    printf "%s " $(sort -R $3 | head -n 1)
    y=$[$y-1]
done
printf "\n"
```

1.1 1a 6 / 12

✓ **- 6 pts Click here to replace this description.**

- incorrect adhoc shuf implementation
- inefficient because of loops
- no final output shuf
- refer solutions

1b (10 minutes). Assuming the shell scripts ‘genspelldata’ and ‘myspell’ exist in the current directory, write a shell script ‘testmyspell’ that takes three unsigned decimal number operands N G B F and tests ‘myspell’ with N different randomly chosen test cases, each containing G good words and B bad words (the latter taken from the file F). For example, ‘testmyspell 1000 5 3 bad.words’ should test ‘myspell’ 1000 times, each with a randomly-chosen test case generated by ‘genspelldata 5 3 bad.words’.



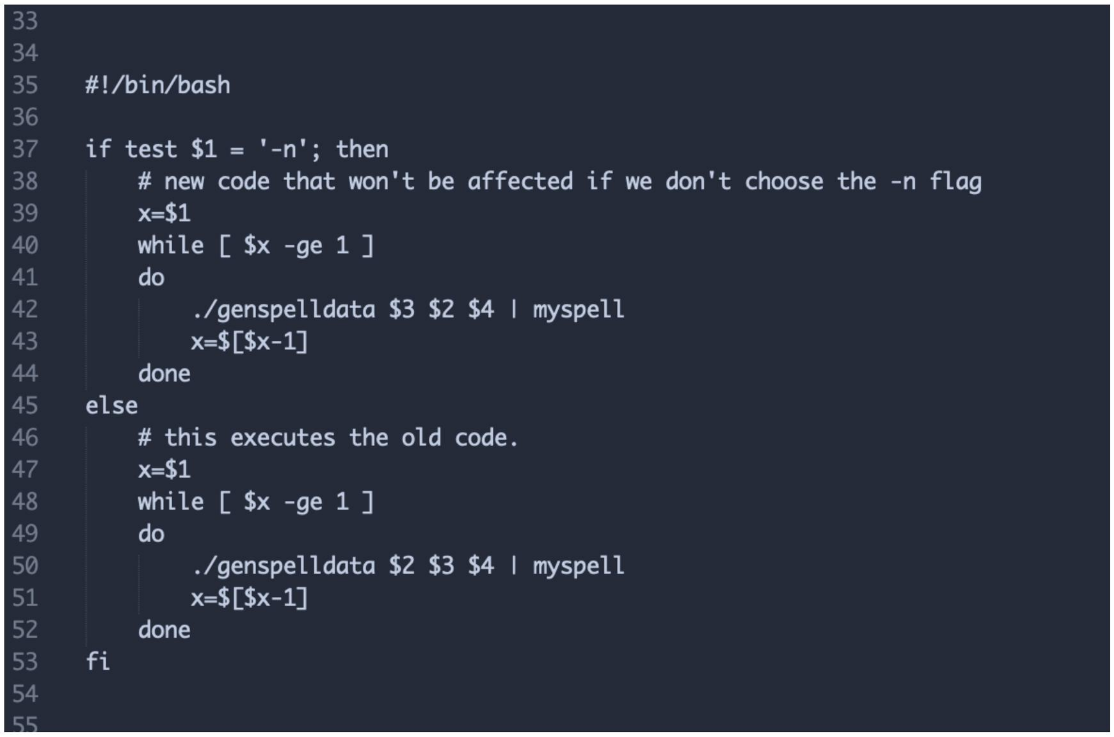
```
#!/bin/bash

x=$1
while [ $x -ge 1 ]
do
    ./genspelldata $2 $3 $4 | myspell
    x=$[$x-1]
done
```

1.2 1b 8 / 10

✓ - 2 pts **Click here to replace this description.**

merely invokes myspell; doesn't verify output

1c (10 minutes). Suppose you are maintaining ‘genspelldata’ and are worried that the changes that you make might break it. You want to have some regression tests for ‘genspelldata’, so that you can test it after changing it, and make sure that it outputs exactly the same thing after the change as before the change. Describe and implement extensions to the behavior of ‘genspelldata’ to make it regression-testable in this sense, while still keeping the functionality that the script already has. Your extensions should add an option or options to ‘genspelldata’ to make it regression-testable. (Hint: see the full documentation for ‘shuf’.)

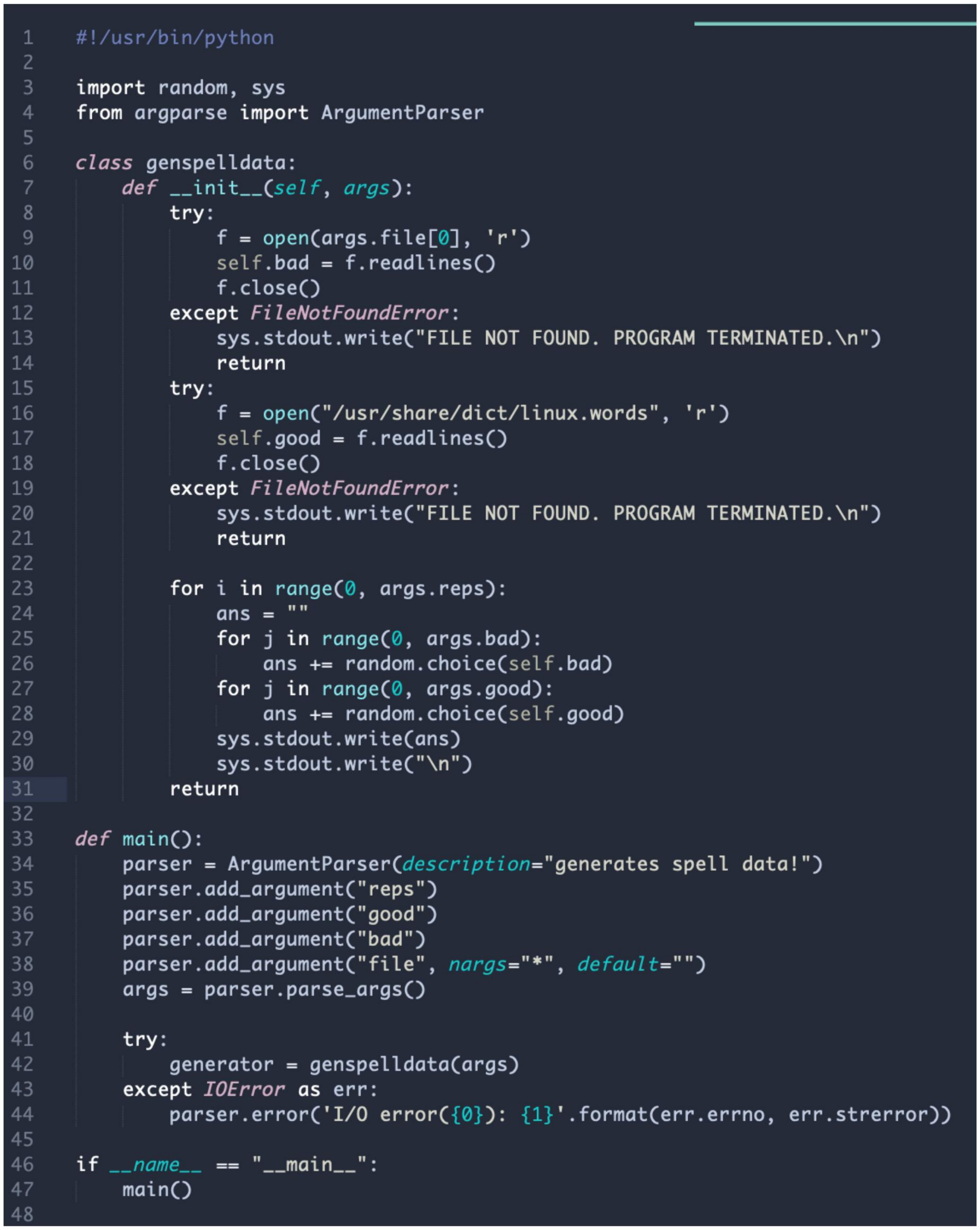


```
#!/bin/bash

if test $1 = '-n'; then
    # new code that won't be affected if we don't choose the -n flag
    x=$1
    while [ $x -ge 1 ]
    do
        ./genspelldata $3 $2 $4 | myspell
        x=$[$x-1]
    done
else
    # this executes the old code.
    x=$1
    while [ $x -ge 1 ]
    do
        ./genspelldata $2 $3 $4 | myspell
        x=$[$x-1]
    done
fi
```

1.3 1c **0 / 10**

✓ **- 10 pts** Click here to replace this description.

2 (14 minutes). Reimplement the ‘genspelldata’ command in Python, using only the argparse, random, string, and sys modules. (See (1a) for the specification of ‘genspelldata’.) Or, if it’s not practical to implement ‘genspelldata’ that way, explain why not, implement it as best you can in Python some other way, and explain any shortcomings in your approach. Either way, make your solution as simple and clear as you can.



```python
#!/usr/bin/python

import random, sys
from argparse import ArgumentParser

class genspelldata:
    def __init__(self, args):
        try:
            f = open(args.file[0], 'r')
            self.bad = f.readlines()
            f.close()
        except FileNotFoundError:
            sys.stdout.write("FILE NOT FOUND. PROGRAM TERMINATED.\n")
            return
        try:
            f = open("/usr/share/dict/linux.words", 'r')
            self.good = f.readlines()
            f.close()
        except FileNotFoundError:
            sys.stdout.write("FILE NOT FOUND. PROGRAM TERMINATED.\n")
            return

        for i in range(0, args.reps):
            ans = ""
            for j in range(0, args.bad):
                ans += random.choice(self.bad)
            for j in range(0, args.good):
                ans += random.choice(self.good)
            sys.stdout.write(ans)
            sys.stdout.write("\n")
        return

def main():
    parser = ArgumentParser(description="generates spell data!")
    parser.add_argument("reps")
    parser.add_argument("good")
    parser.add_argument("bad")
    parser.add_argument("file", nargs="*", default="")
    args = parser.parse_args()

    try:
        generator = genspelldata(args)
    except IOError as err:
        parser.error('I/O error({0}): {1}'.format(err.errno, err.strerror))

if __name__ == "__main__":
    main()
```

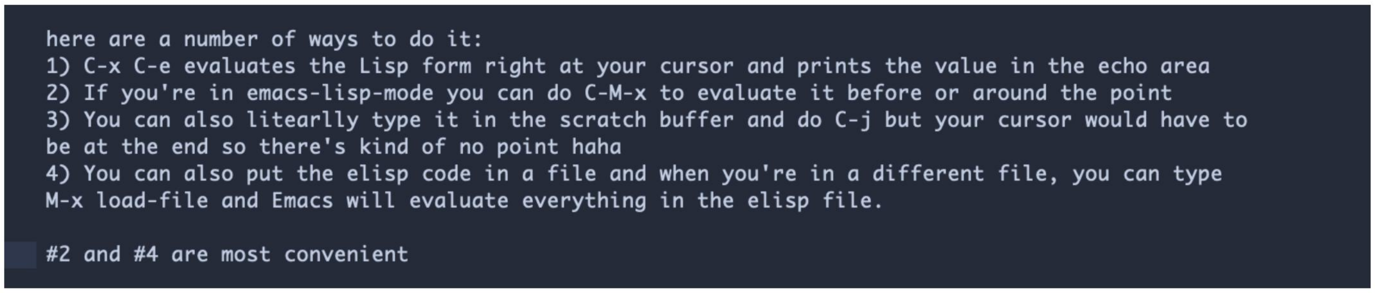
2 2 **14 / 14**

✓ **- 0 pts** Correct

3. Consider the following Elisp code:

```
  (defun delete-horizontal-space (&optional backward-only)
    "Delete all spaces and tabs around point.
  If BACKWARD-ONLY is non-nil, delete them only before point."
    (interactive "*P")
    (let ((orig-pos (point)))
      (delete-region
       (if backward-only
           orig-pos
         (progn
           (skip-chars-forward " \t")
           (constrain-to-field nil orig-pos t)))
       (progn
         (skip-chars-backward " \t")
         (constrain-to-field nil orig-pos)))))
```

3a (6 minutes). Give two ways to call this function from an Emacs session in which you want to delete all spaces and tabs around the cursor. Prefer convenience.

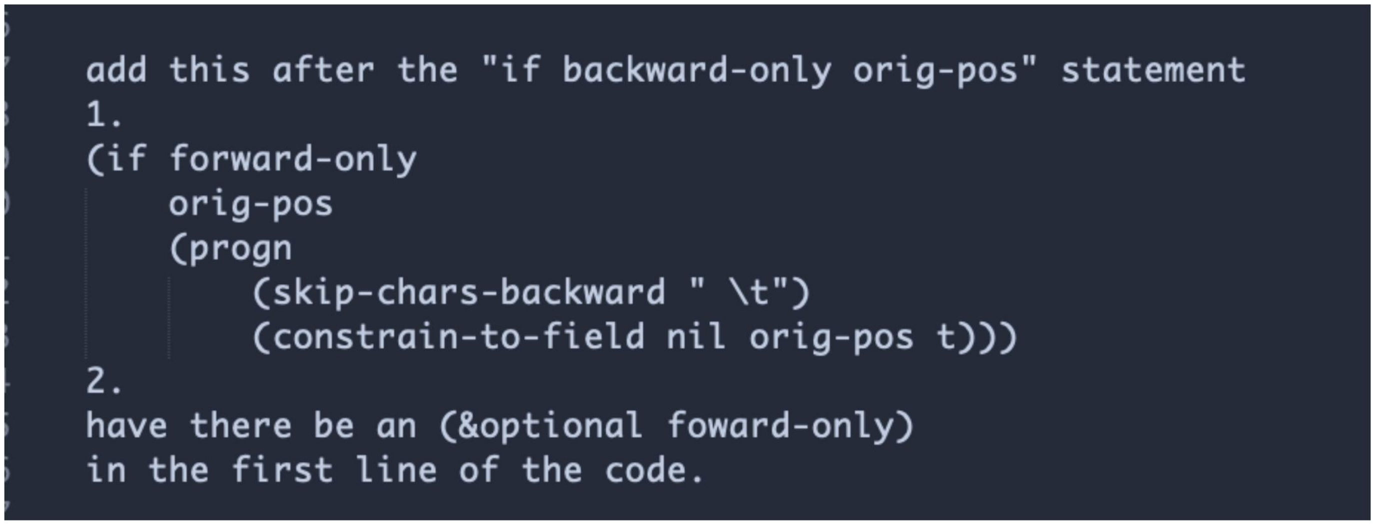

here are a number of ways to do it:
1) C-x C-e evaluates the Lisp form right at your cursor and prints the value in the echo area
2) If you're in emacs-lisp-mode you can do C-M-x to evaluate it before or around the point
3) You can also litearlly type it in the scratch buffer and do C-j but your cursor would have to be at the end so there's kind of no point haha
4) You can also put the elisp code in a file and when you're in a different file, you can type M-x load-file and Emacs will evaluate everything in the elisp file.

#2 and #4 are most convenient

3.1 3a 6 / 6

✓ - 0 pts Correct

3b (6 minutes). Extend the function so that the caller can optionally
delete spaces and tabs only *after* the current position.

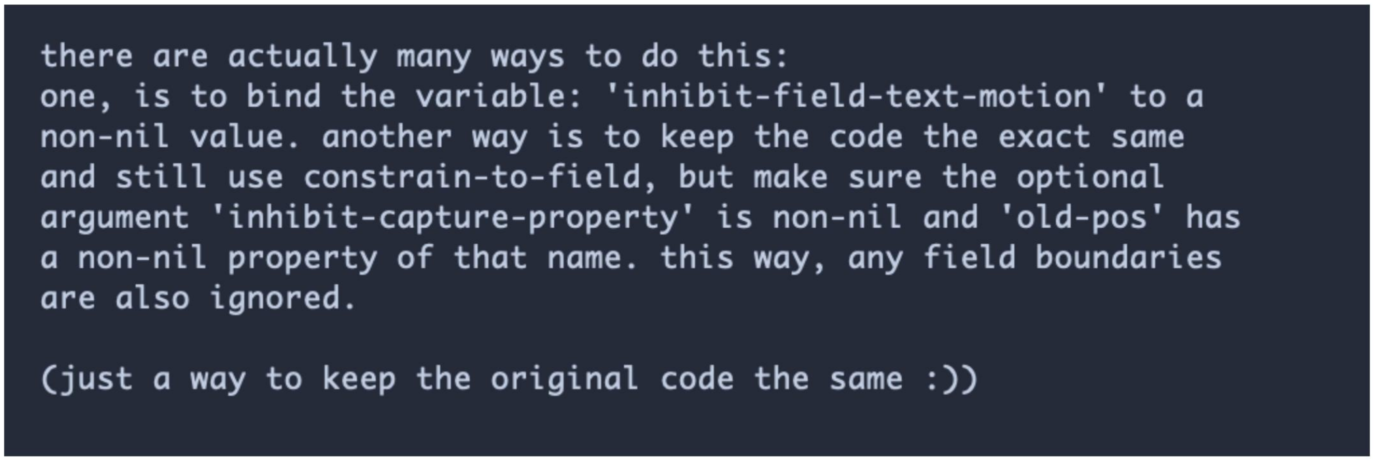


```
add this after the "if backward-only orig-pos" statement
1.
(if forward-only
    orig-pos
    (progn
        (skip-chars-backward " \t")
        (constrain-to-field nil orig-pos t)))
2.
have there be an (&optional foward-only)
in the first line of the code.
```

3.2 3b **0 / 6**

✓ **- 6 pts** significantly incomplete / incorrect; refer solutions

3c (6 minutes). Modify the original (non-extended) function so that it does not worry about fields, i.e., it does not call ‘constrain-to-field’ and always behaves as if the entire buffer were one field.



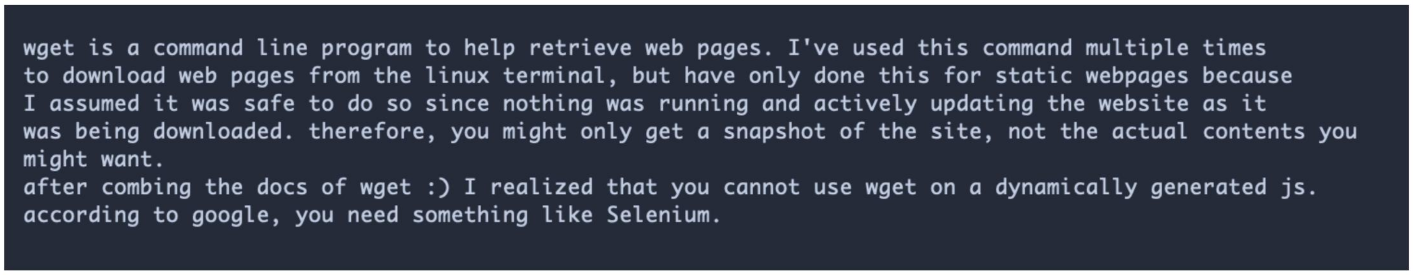
there are actually many ways to do this:
one, is to bind the variable: 'inhibit-field-text-motion' to a non-nil value. another way is to keep the code the exact same and still use constrain-to-field, but make sure the optional argument 'inhibit-capture-property' is non-nil and 'old-pos' has a non-nil property of that name. this way, any field boundaries are also ignored.

(just a way to keep the original code the same :))

3.3 3c **2 / 6**

✓ **- 4 pts** major issues; refer solutions

4 (9 minutes). Explain how the ‘wget’ shell command fits into the client-server computing model as exemplified by React. For example, what are the pros and cons of using ‘wget’ to retrieve data from a system built by React?



wget is a command line program to help retrieve web pages. I've used this command multiple times
to download web pages from the linux terminal, but have only done this for static webpages because
I assumed it was safe to do so since nothing was running and actively updating the website as it
was being downloaded. therefore, you might only get a snapshot of the site, not the actual contents you
might want.
after combing the docs of wget :) I realized that you cannot use wget on a dynamically generated js.
according to google, you need something like Selenium.

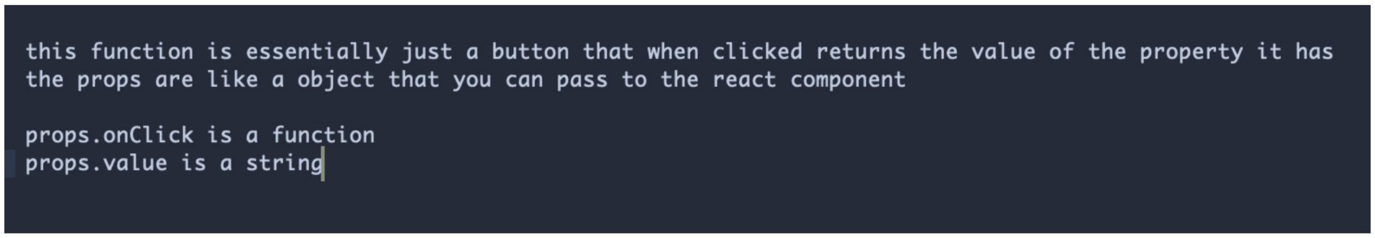
4 4 9 / 9

✓ - 0 pts Correct

5. Consider the following JSX function taken from the React tutorial used as the basis of Homework 3:

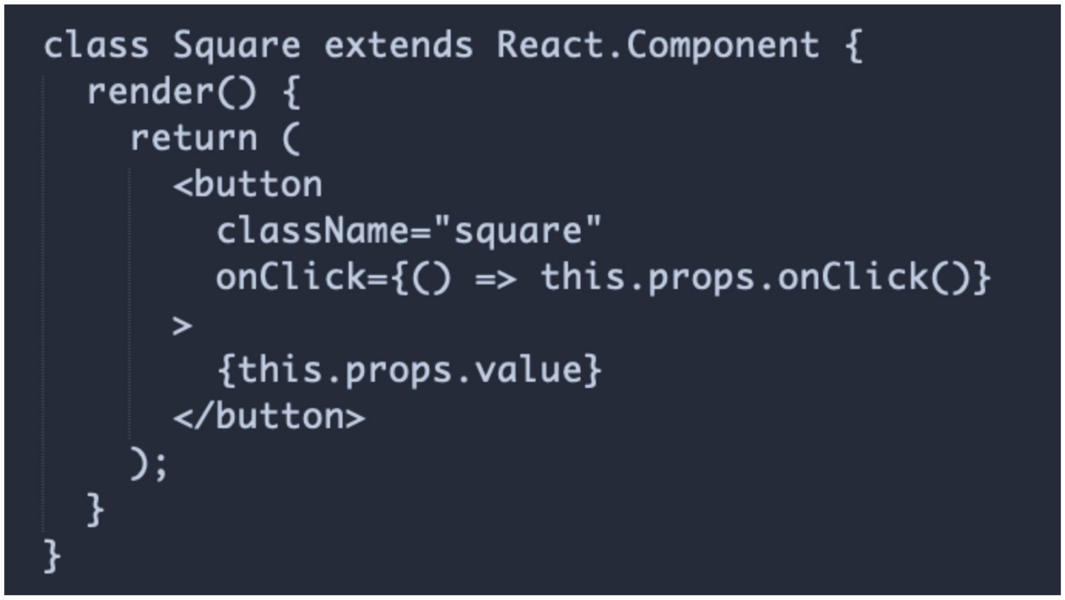
```
  function Square(props) {
    return (
      <button className="square" onClick={props.onClick}>
        {props.value}
      </button>
    );
  }
```

5a (8 minutes). Explain what this function is for and how it works. Give the types of all the identifiers and expressions used in this JSX code.



```
this function is essentially just a button that when clicked returns the value of the property it has
the props are like a object that you can pass to the react component

props.onClick is a function
props.value is a string
```

5b (4 minutes). Write a JSX class that behaves the same way as this function, when used as a React component.



```
class Square extends React.Component {
  render() {
    return (
      <button
        className="square"
        onClick={() => this.props.onClick()}
      >
        {this.props.value}
      </button>
    );
  }
}
```

5.1 5a **4 / 8**

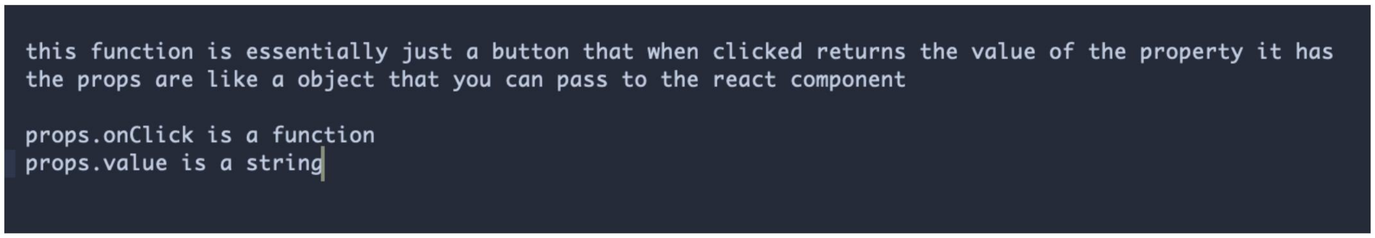
+ **0 pts** Blank
+ **2 pts** Wrote a very simple explanation or most of the detail is incorrect or missing a lot of stuff
✓ + **4 pts** **Explanation is missing a good amount of detail or contains wrong details**
+ **6 pts** Explanation is missing some detail or contains some wrong details
+ **8 pts** Full explanation

gradescope

5. Consider the following JSX function taken from the React tutorial used as the basis of Homework 3:

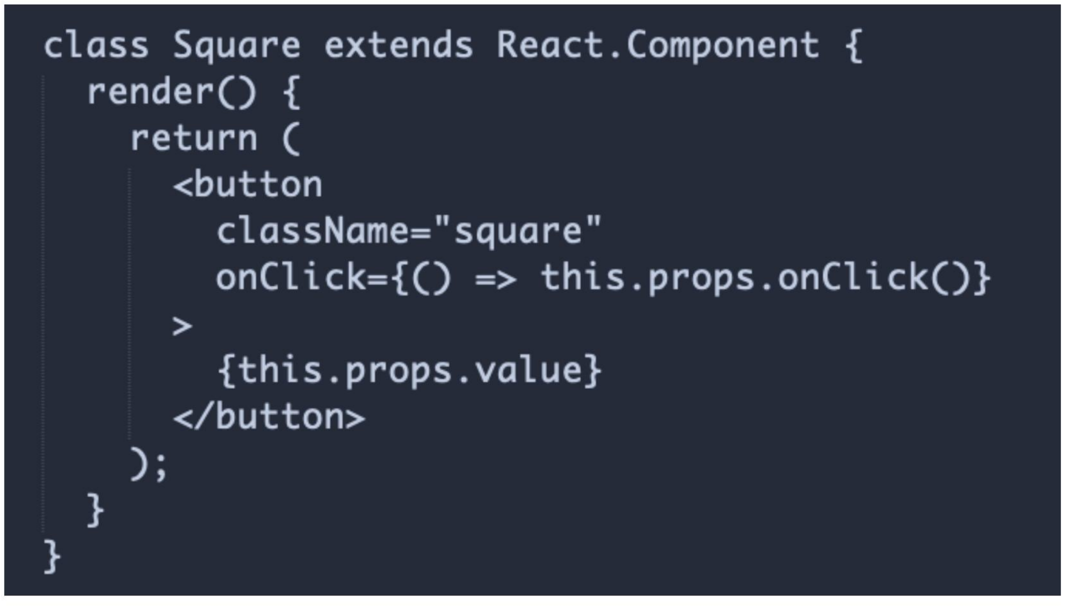
```
  function Square(props) {
    return (
      <button className="square" onClick={props.onClick}>
        {props.value}
      </button>
    );
  }
```

5a (8 minutes). Explain what this function is for and how it works. Give the types of all the identifiers and expressions used in this JSX code.



```
this function is essentially just a button that when clicked returns the value of the property it has
the props are like a object that you can pass to the react component

props.onClick is a function
props.value is a string
```

5b (4 minutes). Write a JSX class that behaves the same way as this function, when used as a React component.

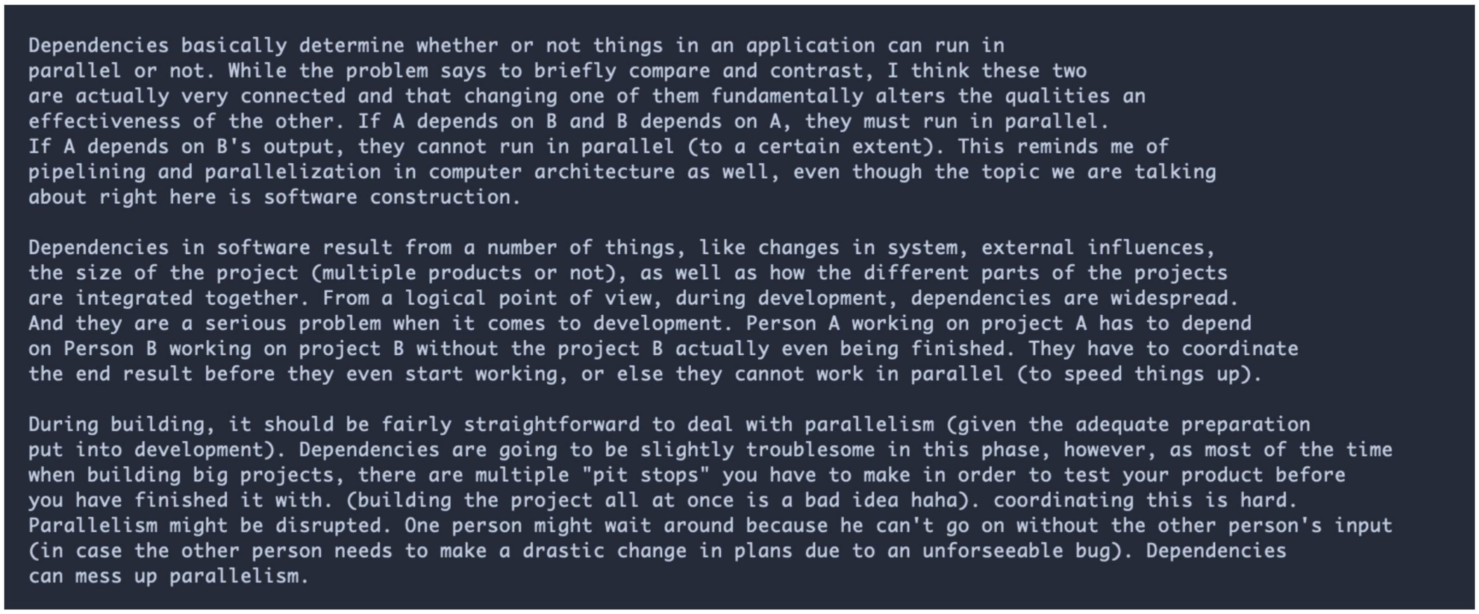


```
class Square extends React.Component {
  render() {
    return (
      <button
        className="square"
        onClick={() => this.props.onClick()}
      >
        {this.props.value}
      </button>
    );
  }
}
```

5.2 5b **4 / 4**

✓ **- 0 pts** Correct

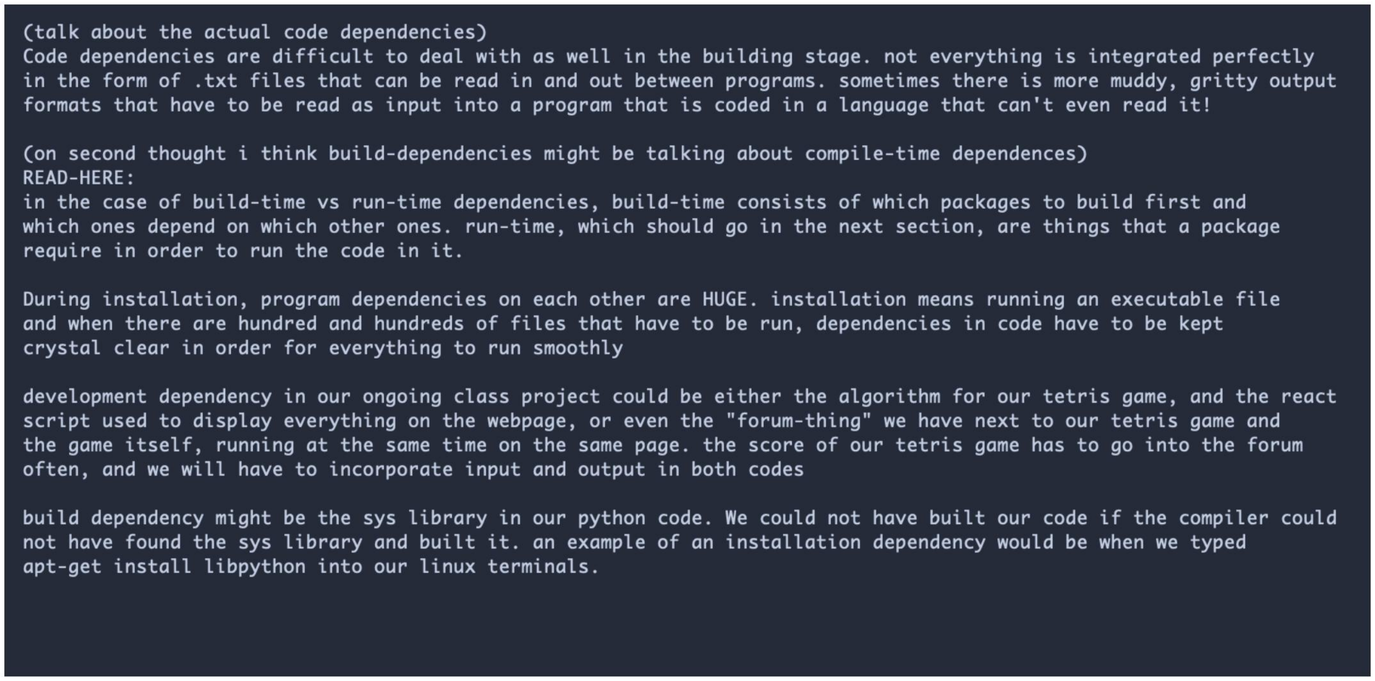
6 (15 minutes). Briefly compare and contrast the role of dependencies and parallelism during development, during builds, and during installation. What do the dependencies have in common, and how do they differ? Give an example of a development dependency in your ongoing class project, and give an example of a build and of an installation dependency; take the latter two examples from the assignments.



Dependencies basically determine whether or not things in an application can run in
parallel or not. While the problem says to briefly compare and contrast, I think these two
are actually very connected and that changing one of them fundamentally alters the qualities an
effectiveness of the other. If A depends on B and B depends on A, they must run in parallel.
If A depends on B's output, they cannot run in parallel (to a certain extent). This reminds me of
pipelining and parallelization in computer architecture as well, even though the topic we are talking
about right here is software construction.

Dependencies in software result from a number of things, like changes in system, external influences,
the size of the project (multiple products or not), as well as how the different parts of the projects
are integrated together. From a logical point of view, during development, dependencies are widespread.
And they are a serious problem when it comes to development. Person A working on project A has to depend
on Person B working on project B without the project B actually even being finished. They have to coordinate
the end result before they even start working, or else they cannot work in parallel (to speed things up).

During building, it should be fairly straightforward to deal with parallelism (given the adequate preparation
put into development). Dependencies are going to be slightly troublesome in this phase, however, as most of the time
when building big projects, there are multiple "pit stops" you have to make in order to test your product before
you have finished it with. (building the project all at once is a bad idea haha). coordinating this is hard.
Parallelism might be disrupted. One person might wait around because he can't go on without the other person's input
(in case the other person needs to make a drastic change in plans due to an unforseeable bug). Dependencies
can mess up parallelism.



(talk about the actual code dependencies)
Code dependencies are difficult to deal with as well in the building stage. not everything is integrated perfectly
in the form of .txt files that can be read in and out between programs. sometimes there is more muddy, gritty output
formats that have to be read as input into a program that is coded in a language that can't even read it!

(on second thought i think build-dependencies might be talking about compile-time dependences)
READ-HERE:
in the case of build-time vs run-time dependencies, build-time consists of which packages to build first and
which ones depend on which other ones. run-time, which should go in the next section, are things that a package
require in order to run the code in it.

During installation, program dependencies on each other are HUGE. installation means running an executable file
and when there are hundred and hundreds of files that have to be run, dependencies in code have to be kept
crystal clear in order for everything to run smoothly

development dependency in our ongoing class project could be either the algorithm for our tetris game, and the react
script used to display everything on the webpage, or even the "forum-thing" we have next to our tetris game and
the game itself, running at the same time on the same page. the score of our tetris game has to go into the forum
often, and we will have to incorporate input and output in both codes

build dependency might be the sys library in our python code. We could not have built our code if the compiler could
not have found the sys library and built it. an example of an installation dependency would be when we typed
apt-get install libpython into our linux terminals.

6 6 **12 / 15**

- **+ 15 pts** Complete solution
- ✓ **+ 12 pts** **missing minor points / details**
- **+ 9 pts** missing major points / details
- **+ 6 pts** only minor overlap with expected points / details; refer solutions
- **+ 3 pts** barely touches expected points / details; refer solutions
- **+ 0 pts** significantly incomplete / incorrect; refer solutons