

CMPT 120—Midterm (Mar. 21st)

BEFORE YOU START DOING THE MIDTERM:
PRINT using CAPITAL LETTERS
YOUR SFU ID AT THE TOP LEFT OF EVERY PAGE.

YOUR SFU ID: is with letters (mine is “jlumbros”).

A few instructions to be **read before taking the midterm:**

- **If you are using a crib/cheat sheet**, please hand it in with your exam, for an extra two bonus points. This is for research purposes, and it will not affect your grade in any way (other than the two bonus points). **Please put your SFU ID in printed capital letters on every page of your crib sheets to get the extra points.**
- There are two exercises in this midterm, and they are both mandatory.
- **Notice that the exercises run on several pages.** There are **8 pages** (on **4 sheets**). Count all pages to make sure you have all of them.
- When you are done, **remain seated until the end of the quiz;**
- I know that other faculties ask for your student number (30114762...), but I am asking for your SFU ID (somename...); **ALL QUIZZES THAT HAVE A STUDENT NUMBER INSTEAD OF AN SFU ID WILL GET A -25.00 GRADE; NOT JUST 0.00, -25.00, CONSIDER YOURSELF FOREWARNED!!!!**
- If what I have seen of Assignment 1, and of Course Exercise 7 is any indication, this midterm should be right down your alley. Keep calm: be more concerned with answering questions right, than doing as many as you can. If the midterm seems like it was too long, grades will be adjusted accordingly.

EXERCISE 1: correct code that does not work [20 pts + 2 extra pts]

Recall the CodeWrite exercise for which you had to write a function `matching_parentheses(s)` which took one string parameter `s`, and returned `True` if the parentheses in `s` were matching and `False` if not:

- "`((hello))(a)`" and "`((())())`" are examples of strings with matching parentheses (the function would return `True`);
- "`)hello(",")()`" and "`((()))()`" are examples of strings with non-matching parentheses (the function would return `False`)

Here are two actual submissions that did not work. Your goal is to **make the minimum number of modifications** to make the submissions work. You can:

- remove a line: strike it out
- modify a line: for this, strike out the line in the original, and write in the box to the right the line you want replacing it
- add a line: for this, simply write in the box to the right, with an arrow pointing at zone where you want to add a line

For example:

<pre>def product_of_range(n): result = 0 for i in range(1,n+1): result = result * i</pre>	<pre>result = 1 return result</pre>
--	--------------------------------------

1) [10 pts] For help, try running the function with "`)()`" and with "`)()`".

<pre>def matching_parentheses(s): exists = False for i in range(len(s)): if s[i] == "(" or s[i] == ")": match = 0 seen = 0 exists = True if exists: for i in range(len(s)): if s[i] == "(": match = match + 1 seen = seen + 1 elif seen > 0 and s[i] == ")": match = match - 1 seen = seen - 1 if exists: if match == 0: return True else: return False else: return True</pre>	
--	--

2) [10 pts] In the following code, "hey) (ho" is an example that fails.

```
def matching_parentheses(s):
    openbrackets = 0
    closebrackets = 0

    for k in s:
        if k == "(":
            openbrackets = openbrackets + 1
        elif k == ")":
            closebrackets = closebrackets + 1

    if closebrackets == openbrackets:
        return True

    return False
```

3) [2 extra pts] Of these two submissions, which do you find easier to understand, and why? Which would you recommend as a preferred coding style?

.....

.....

.....

.....

EXERCISE 2: Tic-Tac-Toe, Connect 4... with lists of lists [28 pts + 5 extra pts]

1) [4 pts] Fill in the following table, knowing that $L = [2, 3, 5, 7, 11]$.

Expression	Value	Explanation
<code>L[1]</code>	3	<code>L[1]</code> returns the second element of list <code>L</code>
<code>L[-1]</code>		
<code>L[0:3]</code>		
<code>L[:]</code>		

We consider that we have a variable `boxes` containing a list of list. You may for instance consider that

```
boxes = [ [ "O", "O", "X", "X" ],
          [ "X", "X", "O", "X" ],
          [ "O", "O", "X", "O" ],
          [ "X", "X", "O", "O" ] ]
```

2) [2 pts] How can you access the list of the row containing the **bold** element (in other words, the row ["X", "**X**", "O", "X"]) from the variable `boxes`?

.....

3) [2 pts] How can you access the element that is in **bold**, from variable `boxes`?

.....

