CMPSCI 120 Fall 2018
Lab #4
Professor William T. Verts

In this assignment you will enter and run a simple Python program. This program receives an integer input from the user, and then computes the factorial of that integer. The final version of the program will automatically email its answer to me at the class mail drop. ***Do not even think about using this program to email anyone but me!***

Setting Up

Using either PuTTY on a PC or ssh on a Mac, log in to your account on the UNIX server. Immediately create a new folder called `python_tests` by typing the command:

> `mkdir python_tests`

Normally you will not need to set any permissions on this folder as you will be the only one using it, but if you need me to look at the files you place there, I will ask you to set the permissions to 755 (`rwxr-xr-x`) with the `chmod` command. Don't do this now.

The Assignment Part 1

Open the `python_tests` folder. By now you should be able to do this without any guidance. Do you remember how to open a folder in UNIX?

Use the emacs text editor to create a new file called `factorial.py`, then type in the program on the next page, but only <u>down to</u> where the comment says **# Stop here**… <u>Do not</u> enter the **# Stop here**… line itself. The line #!/usr/bin/python <u>must</u> be the <u>first</u> line of text in the file, and the line print "Sending: " + Message must be the <u>last</u> line in the file at this point.

Enter ***your own name*** instead of mine and ***your own username*** instead of mine in the lines that assign values to the From and Subject variables (shown underlined). Be careful about quotes, upper-case and lower-case letters, and indentation! These matter a lot to Python! ***Do not*** enter any of the text from after the comment at this point.

Exit emacs, and then use the `chmod` command to set the permissions on `factorial.py` to 755 (we need execute permission because the file is a program).

Run the program by typing `factorial.py` at the UNIX command line. If you get a permissions error, fix the problem with `chmod`. If Python gives you one or more syntax errors, try to figure out what it is complaining about, then go back into emacs and fix the error(s). Do this until running the program allows you to enter a number, compute a factorial of that number, and print out a ***correct*** email template containing the answer.

The Assignment Part 2

Once the program is running, go back into emacs and enter all of the code shown <u>below</u> the **# Stop here**... comment line. Exit emacs. Run the program once again, and fix any errors. <u>I will tell you in class</u> what to use instead of `localhost`. When all is well, run the program with some random number **between 100 and 200**. When your program prints the message `Email sent successfully` you are done with the assignment.

The Python Code

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

import smtplib

N = input("Enter a number --- ")
F = 1
I = 1

while (I <= N):
    F = F * I
    I = I + 1

From    = "Bill Verts <verts@elsrv3.cs.umass.edu>"
To      = "literacy@cs.umass.edu"
Subject = "Factorial of " + str(N) + " from Bill Verts"
Text    = "The Factorial of " + str(N) + " is " + str(F)

Message = "From: "    + From    + "\r\n" + \
          "To: "      + To      + "\r\n" + \
          "Subject: " + Subject + "\r\n" + \
          Text

print "Sending: " + Message

# Stop here until all code above this point is working

Server  = smtplib.SMTP("localhost")

try:
    Code = Server.sendmail(From, [To], Message)
finally:
    Server.quit()

if Code:
    print "Error sending email"
else:
    print "Email sent successfully"
```

Grading

Points will be removed for the following items:

A.   Spacing errors in the text. This will most often be seen in the **Subject:** line or in the message body where spaces between words were absent.

B.   N! where N is not in the range 100…200. The code may work correctly, but the assignment states that the final result is to be for $100 \leq N \leq 200$. Students who send in an out-of-range case but who <u>also</u> sent in a correct version will not be penalized.

C.   The Factorial is incorrectly computed. In most cases this is a bug in the Python program stating **F = F + I** instead of **F = F * I** (addition instead of multiplication). The program works, but computes and sends the wrong answer.

D.   The **Subject:** and/or **From:** lines state **Bill Verts** instead of the name of the student.

E.   The subject line is missing entirely (often due to a mistake where the subject line appears as part of the email body text).

F.   The student's name is missing, either partially or completely, in the **From:** line of the email. Often this line is just an email return address, but no name.

G.   The result of the Factorial computation was in the **Subject:** line (instead of or in addition to being in the message body).

H.   Error in the **To:** line, causing that line to be omitted or combined with another line.