Goals

By the end of this assignment, you will have
  • more practice with Python string methods and conditionals
  • practice with Python indefinite and definite repetition

DNA is housed within the nucleus of our cells. It controls cellular activity by coding for the production of proteins. The information in DNA is not directly converted into proteins, but must first be copied into RNA. This ensures that the information contained within the DNA does not become tainted.

DNA transcription is the process that involves transcribing genetic information from DNA to RNA. The transcribed DNA message, or RNA transcript, is used to produce proteins.

Write a function, transcribe, which has one parameter, a string S, which will have DNA nucleotides (capital letter As, Cs, Gs, and Ts). There may be other characters, too, though they will be ignored by your transcribe function -- these might be spaces or other characters that are not really DNA nucleotides. Then, transcribe should return as output the messenger RNA that would be produced from that string S. The correct output simply uses replacement:

'A' s in the input become 'U' s in the output.
'C' s in the input become 'G' s in the output.
'G' s in the input become 'C' s in the output.
'T' s in the input become 'A' s in the output.
Any other input characters should not appear in the output.

Test your code using the following examples:
>>> transcribe('ACGT	
  TGCA')
'UGCAACGU'
>>> transcribe('GATTACA')
'CUAAUGU'
>>> transcribe('cs5')
  # lowercase doesn't count
''

(0)(a) DNA is housed within the nucleus of our cells. It controls cellular activity by coding for the production of proteins. The information in DNA is not directly converted into proteins, but must first be copied into RNA. This ensures that the information contained within the DNA does not become tainted.

DNA transcription is the process that involves transcribing genetic information from DNA to RNA. The transcribed DNA message, or RNA transcript, is used to produce proteins.

Write a function, transcribe, which has one parameter, a string S, which will have DNA nucleotides (capital letter As, Cs, Gs, and Ts). There may be other characters, too, though they will be ignored by your transcribe function -- these might be spaces or other characters that are not really DNA nucleotides. Then, transcribe should return as output the messenger RNA that would be produced from that string S. The correct output simply uses replacement:

'A' s in the input become 'U' s in the output.
'C' s in the input become 'G' s in the output.
'G' s in the input become 'C' s in the output.
'T' s in the input become 'A' s in the output.
Any other input characters should not appear in the output.

Test your code using the following examples:
>>> transcribe('ACGT TGCA')
'UGCAACGU'
>>> transcribe('GATTACA')
'CUAAUGU'
>>> transcribe('cs5')  # lowercase doesn't count
''

(0)(b) If you wrote function transcribe (0)(a) using a while loop, rewrite the function using a for loop. If you wrote function transcribe using a for loop, rewrite the function using a while loop.

(1) Revise mars_explore_main function from project 4 to take one parameter, num_trips. Then change the body of the code to call the mars_explore function num_trips times.

For example:
>>> mars_explore_main (100)

Note: you may want to use the turtle speed method in mars_explore_main before the mars_explore function is called: speed(0) is the fastest turtle speed.
(2) Turtle Spirolaterals

To draw a name spirolateral, the turtle moves forward for each letter in the name. Then it turns 90 degrees. The amount the turtles moves is based on the letter’s position in the alphabet. A is 1 (the turtle could move forward 5 steps, for example), B is 2 (the turtle could move 10), C is 3 (the turtle could move 15), and so on.

The UTF-8 (ASCII) code of a letter can tell us its position in the alphabet. The UTF-8 code for A is 65, the UTF-8 code for B is 66. Recall that Python functions ord and chr translate from characters to their UTF-8 representation and back.

The spirolateral function is given to you here and in project6start.py:

```python
def spirolateral(name):
    """ (string) -> None
    Draw spirolateral of the input name.
    >>> spirolateral('Ducks')
    ""
    name = name.upper()
    start = ord('A')
    multiplier = 5

    ctr = 0
    while ctr < len(name):
        ch = name[ctr]
        letter_place = ord(ch) - start + 1
        fd(letter_place * multiplier)
        rt(90)
        ctr += 1

    return #None

>>> spirolateral('Ducks')
```

Write a function, spirolateral_main, that takes one input, name, and calls spirolateral (passing name as the input arg) as many times as there are letters in name. No value is returned. In the example below, spirolateral_main calls spirolateral five times.

```python
>>> spirolateral_main('Ducks')
```
(3) Write a function, rats, with two parameters, weight and rate, that returns the number of weeks it will take for a rat to weigh 1.5 times its original weight (weight), if it gains at rate (rate). Print the rat’s weight each week.

(4)
(a) Define a function, any_uc_alpha, with one parameter, astring, which returns True if any character in astring is an uppercase letter, and returns False otherwise. (A hint is at the end of the project description.)

(b) Define a function, two_numbers, with one parameter, astring, which returns True if astring has at least two numbers, and otherwise returns False. (Hint: investigate string method isdigit and refer to the hint for (4a).)

(c) Define a function, any_special_char, with one parameter, astring, which returns True if astring contains any of the special characters, '!','@','#','$','%','^','&', and otherwise returns False. (Hint: define a string variable which has the value of a string containing all of the special characters.)

(Challenge – CS Circles/Recursion)
A countdown function may be written using a while loop:

```python
def countdown(n):
    while n > 0:
        print(n)
        n = n - 1

    print('Blastoff')
```
The same function can also be written using a technique called recursion, where a function calls itself:

```python
def countdownr(n):
    if n == 0:
        print('Blastoff!')
    else:
        print(n)
        countdownr(n - 1)
```

You may want to explore countdownr by calling it using the Python visualizer:

Then edit the above function to write recursive function countdownBy2, which also counts down from n and prints 'Blastoff', but counts down by 2.

For example,
```python
>>> countdownBy2(7)
7
5
3
1
Blastoff!
```

**Grading Rubric**

This project will be marked out of 70 points. Each new function (transcribe, spirolateral_main, rats, and any_uc_alpha, two_numbers, any_special_char) is worth 10 points: 2 point for the function header, 3 points for the docstring, 2 points for the body of the function, 1 point for the return statement, and 2 points for correct output. The edits to mars_explore_main are worth 5 points: 1 point for the function header, 1 point for the docstring, 2 points for the edits in the function, and 1 point for correct execution. The rewrite of transcribe is worth 5 points: 3 point for the edits and 2 points for correctness.

**Getting Started**

**Write your program using an editor window**, so you can save your file.

Examples are for testing your code. You do not need to turn in the results of code tests.

Programming style is important! Remember
- Include a docstring in every function
- Use whitespace between operators and operands
- Use descriptive variable names
- Add appropriate comments.
Finishing & submitting your work

When you have completed all of the problems, add additional comments to your code to make sure the functions for each problem are clear. Check the list of functions above to make sure that your Python file includes all of them. Do a final Save command to save the code in the editor window as a file with the name cis122project6.py.

To submit your project, login to Blackboard. From the menu on the left hand side of the screen, choose "Projects". In the projects folder, choose "Submit In the Loop". In Section 2 of the page that is displayed, scroll down to "Attach file" and choose "Browse My Computer". Locate cis122project6.py (the file you just created), and double click on it. The file name will appear on the "Attached Files" list.

At the bottom of Section 2, you will see a "Comments" window. This is where you credit all of the sources of any help you may have received on this assignment, including your partner if you are working in a programming pair. This is also the place to include any feedback you may have about the assignment and/or any remaining questions you may have.

Scroll down to Section 3 and hit the "Submit" button. You may re-submit your project up until the project deadline. Only the final submission will be graded.

HINT: Look at each character in the string separately, using an appropriate string method.