

ECS 50 Winter 2019
Homework Assignment 5
Due no later than 10:00pm Friday, March 8

This assignment deals with characters and strings, as described in Chapter 7. Assignment 6 may be posted before Assignment 5 is due, so finishing Assignment 5 early may be advantageous.

Problem 1 (50 points)

The Python programming language provides the programmer with a variety of string manipulation methods (functions). For this assignment, you'll implement two of those functions as CUSP subroutines.

The first of these Python methods is `swapcase()`. This function expects a string as an argument and returns a copy of the original string but with a lower case letters converted to upper case, and all upper case letters converted to lower case. Characters that are not letters remain unchanged. Here is an example of the function in action:

```
>>> test = "lEaRn pYtHOn iN 30 YeaRS!"
>>> result = test.swapcase()
>>> print(result)
LeaRn PyThoN In 30 yEaRs!
```

The task for you is to write a subroutine in CUSP that does what `swapcase()` does. Your subroutine should expect the following parameters to be pushed on the stack by the calling procedure in this order:

- The starting address of the original string will be pushed first on the stack

- The number of characters in that string will be pushed next

- The starting address of the converted string will be pushed last

Name your subroutine `swapcase`. Your subroutine should create a new string as described above and place the new string in memory starting at the address given in the last parameter pushed on the stack. Note that the original string should not be altered in any way by your subroutine.

As with the previous homework assignment, you'll want to write a "main program" to serve as the calling procedure to test your subroutine. Have your calling procedure load into memory at \$000, and have your `swapcase` subroutine load into memory at \$020. Put your calling procedure and your

subroutine in a text file named hw5p1.csp. Put the object code in a file named hw5p1.obj.

Problem 2 (50 points)

The second of the Python string methods is `title()`. This function expects a string as an argument and returns a copy of the original string but with the first letter in each word is changed to upper case and with all other letters changed to lower case. Characters that are not letters remain unchanged. Here is an example of the function in action:

```
>>> test = "lEArN pYtHOn iN 30 YeaRS!"
>>> result = test.title()
>>> print(result)
Learn Python In 30 Years!
```

The task for you is to write a subroutine in CUSP that does what `title()` does. Your subroutine should expect the following parameters to be pushed on the stack by the calling procedure in this order:

- The starting address of the original string will be pushed first on the stack

- The number of characters in that string will be pushed next

- The starting address of the converted string will be pushed last

Name your subroutine `title`. Your subroutine should create a new string as described above and place the new string in memory starting at the address given in the last parameter pushed on the stack. Note that the original string should not be altered in any way by your subroutine. Also note that in Python, the `title()` method considers apostrophes to be word boundaries. Ignore that fact, so your subroutine should only treat spaces as word boundaries, not apostrophes.

Once again, you'll want to write a "main program" to serve as the calling procedure to test your subroutine. Have your calling procedure load into memory at \$000, and have your `title` subroutine load into memory at \$020. Put your calling procedure and your subroutine in a text file named hw5p2.csp. Put the object code in a file named hw5p2.obj.

Include all four files (the two .csp files and the two .obj files) in your submission through Canvas.