CSC 120 **Lab 5**

Mock Midterm. Work alone for the first 30 minutes. (This will NOT be graded, but you will get attendance points for the lab.)

1. Write a Python function called `slice_and_reverse(input_str)` that takes a string `input_str` as input and returns a new string that contains the first three characters of the input string, followed by the last two characters, all reversed. If the input string is less than five characters long, return the string `"Input string is too short."`
   Example:
   `slice_and_reverse('abcdefxy')` returns `'yxcba'`

   Note: `s[::-1]` will create a string that is the reverse of `s`.

2. Write a Python function called `every_other(alist)` that takes a list `alist` as an argument and returns a new list consisting of every other element of `alist`.
   Example:
   `every_other(["sun","moons","stars","planets","asteroid"])`

   returns the list `['sun', 'stars', 'asteroids']`

3. Write a Python function `print_diag_pairs(grid)` that takes a 2-d list `grid` as an argument and prints each element of the main diagonal along with the element adjacent to it (if there is one). In the example grid below, each element on the main diagonal and its adjacent pair is bolded:

```
grid = [[ 2, 6, 3, 4 ],

        [ 8, 9, 5, 12],

        [ 6, 4, 2, 7 ],

        [ 9, 5, 3, 10]]
```

Using the list `grid` defined above, the call `print_diag_pairs(grid)` would have the output below:

```
2 6
9 5
2 7
10
```

Note: You can assume that the argument `grid` is in fact a grid (that is, a list of equal-length lists of numbers). Do not modify the argument list `grid`.

4. (Short answer.)

a) Name two types that cannot be used as keys in a dictionary

b) What must the `__str__(self)` method return?

c) In a diagram, we draw a reference using an arrow. What does the arrow represent?

d) The function below determines if a string is a sentence. (See the Note below.)

```
def find_sentence(sentence):
  if sentence[0].isupper() and sentence[-1] == ".":
    return True
  if sentence[0].islower() or sentence[-1] != ".":
    return False
```

Give the return value for each of these calls:

`find_sentence("The cat is cute.")` _______

`find_sentence("dog")` _______

`find_sentence("123.")` _______

Note:
`isupper()` returns `True` if the letter is capitalized and `False` otherwise
`islower()` returns `True` if the letter are lowercase and `False` otherwise
Both return `False` for a numeric value

5. Write a Python function called `count_words(filename)` that takes the string argument `filename` which is the name of a file. The function will read the file and count the number of words that are of length ***greater*** than two. (Punctuation is counted in the length.) A word is defined as a sequence of characters separated by whitespace. The function ignores the first line of the file, which starts with a "#" character. For example, a file containing the following lines:

```
#poem
I wandered lonely as a cloud
When all at once I saw a crowd,
```

the function will return the integer **8**.

6. Define a Python class called Rectangle that has two attributes: width and height. Your class must define the following methods:

   - __init__(self,w,h): creates a Rectangle object and initializes the attributes
   - get_width : returns the width of a Rectangle
   - get_height: returns the height of a Rectangle
   - __str__ : returns a string representation of a Rectangle; if a Rectangle has width 3 and height 4, it would return the string "Rectangle(3,4)"
   - __eq__(self, other): returns True if the widths and heights of two Rectangle objects are the same and False otherwise