## Question 1. True or False

Circle **T** if the statement is true, otherwise circle **F** if the statement is false.

1. Dynamically-typed languages do not perform type checking. **T** **F**

2. In Python, the `pass` statement has no semantic significance. **T** **F**

3. In C++, the expression `7/3 == 2` is evaluated to true. **T** **F**

4. Let `a` be a variable of type `list`, then `a.append(5)` will return a new list with the element 5 added to the end of the new list. **T** **F**

5. Using lazy iterators (instead of non-lazy iterators) improves the performance (speed) of the program. **T** **F**

## Question 2. Multiple Choices

Pick all answer(s) that are correct.

a) Which of the following statements are true about immutable objects?

i. All methods and operators that return an immutable object will always make a new object.

ii. It is safe to have aliases to immutable objects.

iii. Immutable containers cannot have references to mutable ones (e.g. a tuple containing a list).

iv. In Python, literals are, by definition, immutable objects.

v. In general, operations on immutable objects are less efficient than their mutable counterparts (e.g. tuple vs. list)

b) Which of the following is true about the assignment operator in Python?

i. It is a statement.

ii. It is an expression.

iii. In chained assignment, the expression on the right hand side is only evaluated once.

iv. It is always done by reference.

v. It is illegal to assign an existing variable to a value of a different type.

## Question 3. Short Questions

a) What slice of the word “washington" will give the result of “ogisw”? (Give answer in the form `[i:j:k]`)

`[-2::-2]` or `[8::-2]`

b) What is the output of this program? (Try to do this by hand)

```
for t in enumerate(range(5, 0, -1)):
    print("%d-%d"%t, end=" ")
```

`0-5 1-4 2-3 3-2 4-1`

## Question 4. Programming Questions

a) Complete a short Python script such that it will sort each character in ASCII order for the string variable `input`, and store the result in a variable named `output`. Bonus mark is given if you can do it in one line of Python code.

```
input = "incommodious"

# process input here



output = "".join(sorted(input))



# should print "cdiimmnooosu"
print(output)
```

b) A CSV file allows data to be saved in tabular form. For each row, items are separated by the comma character. For example, this table,

| | | |
|---|---|---|
| 4 | 8 | 15 |
| 16 | 23 | 42 |

can be written into a CSV file that looks like this:

```
csv = "4, 8, 15 \n 16, 23, 42 \n"
```

Assume there are only numbers in your table, process the `csv` variable such that you get a two-dimensional list that looks like this:

```
[[4, 8, 15], [16, 23, 42]]
```

Note: use the `int()` function to convert string to integer (e.g. `int("12")` will return the integer 12).

Stores the result in a variable named `output`. You may not assume how many rows and columns there are in the table.

```
output = []

for line in csv.split("\n"):
    row = []
    cleaned = line.strip()
    if len(cleaned) > 0:
        for token in cleaned.split(","):
            row.append(int(token))
        output.append(row)
```

c) In a Python script, approximate Euler's number, *e*, using the infinite series:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!}$$

up to `N` terms. For example, if `N` is 4 then *e* is 2.6666666666666665.

```
ans = 0.
factorial = 1
for x in range(1, N+1):
    ans += 1/factorial
    factorial *= x
```