Section 4 Handout – Midterm Practice

1. Evaluate the following Python expressions:

\[(5 / 2) + 2 * 2\]

\[\['live', 'long', 'and', 'prosper']][1][1:\]

\[len({1:'one', 2:'two', 3:'three'}[2])\]

\[float(str(2 + 2) + '5') + 1\]

\[itemgetter(1)(['to', 'boldly', 'go'])\]

1. Write a function that sorts a list of numbers by their absolute value, and returns the result. For example:

\[sort_abs([2, -1, 4, -5, -2, 1])\]

returns \([-1, 1, 2, -2, 4, -5]\).

2. Write a function that takes a list as a parameter, and returns a set containing the elements that appear more than once in the list. For example:

\[dups([1, 3, 2, 4, 3, 1, 1])\]

returns \(\{1, 3\}\).

3. Write a function that takes a string as an argument, and returns a dictionary that maps each character to its frequency in the given string. For example:

\[freq('Star Wars')\]

returns \(\{'S':1, 't':1, 'a':2, 'r':2, ' ':1, 'W':1, 's':1\}\)

4. Write a function that reverses a list, without using the built-in reverse function. Your function should return the reversed list, and not modify the list passed as a parameter. For example,

\[reverse([1, 2, 3])\]

returns \([3, 2, 1]\).
5. Consider the following Python program:

```python
def pos_dif(y, x):
    '''
    Returns the positive difference of two numbers.
    '''
    # Location B
    return abs(x - y)

def percent_error(actual, expected):
    '''
    Returns the percent error of an experimental result.
    '''
    # Location A
    x = pos_dif(actual, expected)
    y = expected
    # Location C
    return x / y

a = 15.0
b = 10.0
print percent_error(a, b)
```

For each of the locations indicated above, draw the environment frame(s) at that moment during execution.
Section 4 Solutions

\[
\frac{5}{2} + 2 \times 2 = 6
\]

\[
\text{len({}1:'one', 2:'two', 3:'three'}) = 3
\]

\[
\text{len('two')} = \text{len('two')} = 3
\]

\[
\text{float(str(2 + 2) + '5')} + 1 = \text{float(str(4) + '5')} + 1 = \text{float('45')} + 1 = 46.0
\]

\[
\text{itemgetter(1)(['to', 'boldly', 'go'])} = \text{f(['to', 'boldly', 'go'])}
\]

```
def sort_abs(items):
    return sorted(items, key=abs)
```

```
def dups(list):
    seen = set()
    result = set()
    for elem in list:
        if elem in seen:
            result.add(elem)
        seen.add(elem)
    return result
```

```
def freq(str):
    result = {}
    for c in str:
        if c not in result:
            result[c] = 0
        result[c] = result[c] + 1
    return result
```
def reverse(list):
    result = []
    for i in range(len(list) - 1, -1, -1):
        result.append(list[i])
    return result

def reverse(list):
    result = []
    for e in list:
        result.insert(0, e)
    return result

Location A:
Global Environment percent_error
da → 15.0 actual → 15.0
b → 10.0 expected → 10.0
pos_dif → (function)
percent_error → (function)

Location B:
Global Environment percent_error pos_dif
da → 15.0 actual → 15.0 y → 15.0
b → 10.0 expected → 10.0 x → 10.0
pos_dif → (function)
percent_error → (function)

Location C:
Global Environment percent_error
a → 15.0 actual → 15.0
b → 10.0 expected → 10.0
pos_dif → (function) x → 5.0
percent_error → (function) y → 10.0

For more information, execute the code using the Python Tutor.