1. Evaluate the following Python expressions:

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

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

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

\[
\text{float}(\text{str}(2 + 2) + "5") + 1
\]

\[
\text{itemgetter}(1)(["to", "boldly", "go"])
\]

2. Write a function that sorts a list of numbers by their absolute value, and returns a new sorted list as the result.
For example: \texttt{sort_abs([2, -1, 4, -5, -2, 1])} returns \{-1, 1, 2, -2, 4, -5\}.

3. 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: \texttt{duplicates([1, 3, 2, 4, 3, 1, 1])} returns \{1, 3\}.
4. 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}`.

5. 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_list([1, 2, 3])` returns `[3, 2, 1]`. 
6. 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.