Midterm notes

q1: Looping

Reasoning about a series of values and loop termination

```python
def q1():
x = 10
y = 10
while y > 0:
y = y - 2
x = x - 1
print(x)
```

Common error: One iteration too many

Less common: One iteration too few

q2: Function call

Control flow, function call/return

```python
def area(x1, y1, x2, y2):
    """There should be a good docstring here,
    but it's an exam so I left it off.
    """
    width = x2 - x1
    if width < 0:
        width = 0 - width
    height = y2 - y1
    if height < 0:
        height = 0 - height
    a = width * height
    return a

def q2():
ar = area(5, 5, 10, 10)
print(ar)
```

Few errors.

q3: Passing a list

Lots of confusion about passing a list

```python
def q3():
    lis = [7, 4, 7, 7, 3]
    occur_num(7, lis)
    for elem in lis:
        print(elem)

def occur_num(pat, ar):
    count = 0
    for i in range(len(ar)):
        if ar[i] == pat:
            count += 1
        ar[i] = count
    return
```
q3: Passing a list

```python
def q3():
    lis = [7, 4, 7, 3]
    occur_num(7, lis)
    for elem in lis:
        print(elem)

def occur_num(pat, ar):
    count = 0
    for i in range(len(ar)):
        if ar[i] == pat:
            count += 1
            ar[i] = count
    return
```

Stack

```
<table>
<thead>
<tr>
<th>Occur_num()</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar: [7]</td>
</tr>
<tr>
<td>pat: 7</td>
</tr>
</tbody>
</table>
```

Heap

```
lis: [7, 4, 7, 7, 3]
```

---

q4: multiple function calls

```python
def double(x):
    x = x + x
    return x
def q4():
    x = 10
    y = 3
    while y < x:
        y = double(y)
    print(y)
```

Common errors:

- Loop too many times, or not enough
- Scope and lifetime of `x`

---

q5: Sum of selected elements

```python
def sum_even(ar):
    """Sum of the even numbers in ar.
    Args:
    ar: a list of integers (not checked for validity)
    Returns:
    The sum of elements in ar that are divisible by two.
    Examples:
    sum_even([1, 2, 3, 4, 5]) == 2 + 4 == 6
    sum_even([3, 5, 7, 9]) == 0
    sum_even([]) == 0
    """
    Generally pretty good.
    Several people built an extra array. Not the best way.
q5:  
def sum_even(ar):
    """Sum of the even numbers in ar.
    Args:
    ar: a list of integers
    (not checked for validity)
    Returns:
    The sum of elements in ar that are divisible by two.
    Examples:
    sum_even([1, 2, 3, 4, 5]) == 2 + 4 == 6
    sum_even([3, 5, 7, 9]) == 0
    sum_even([]) == 0
    """
    sum = 0
    for el in ar:
        if el % 2 == 0:
            sum += el
    return sum

q6:  
def dedup(str):
    """Return copy of str except that any run of characters
    is reduced to a single occurrence.
    Args:
    str: A string that may contain runs of duplicate
    characters.
    Return:
    A new string, which is a copy of str except that
    every run of identical characters is replaced by
    a single occurrence.
    Examples:
    dedup("abcdeffgh") => "abcdefgh"
    dedup("abc") => "abc"
    dedup("aaaaaaa") => "a"
    dedup("") => ""
    """
    result = ""
    prev = ""
    for char in str:
        if char != prev:
            result += char
        prev = char
    return result

Overall impressions

The exam was a little harder than I expected

We might need more review on
- Loop patterns with indexing
- Functions calls and scope, esp. references

Don’t look at only your score ... look at what
problems you had, and need to work on.