Sample Final

The actual one will be longer, but these are the types of questions likely to be on it.

1. Multiple choice: which of the following expressions will cause an error to occur when it is evaluated?
   (a) `3 + 5.0`
   (b) `10 % 4 + 7 // 2`
   (c) `abs(5 - 20 // 3) ** 4`
   (d) `"If %d + %d = %2.2f, then %s"% (2, 2, 4, "bye")`
   (e) `4 // "3"`

2. Multiple choice: if `s = "abcdefg"`, what is the value of `s[2:-4]`?
   (a) `"bc"`
   (b) `"c"`
   (c) `""` (that is, the empty string)
   (d) `"fed"`
   (e) It causes an exception

3. Convert the following into Python; you may assume the string and math modules are imported already:
   (a) The volume *vol* of a sphere is $4\pi r^3$ divided by 3 (remember the result is a floating point number!)
   (b) The value of the string variable *str* with all occurrences of the letter "`e`" replaced by the character "`3`"
   (c) Subtract 159 from the product of 3 and 27, using integers

4. The A–F grading system assigns the following grades to scores. If your score is less than 1 point, you get an F; if it is less than 2 points, you get a D; if it is less than 3 points, you get a C; if you get less than 4 points, you get a B; and if you get 4 points or more, you get an A. Write an "if" statement that, given a score in the variable `score`, prints the corresponding grade.

5. What does the following function do when given a list of numbers as the argument?

```
def f(lst):
        a = i = 0
        n = len(lst)
        while i < n:
                if lst[i] <= 0:
                        i += 1
                        continue
                a += lst[i]
                i += 1
        return a / n
```

6. Rewrite the function in the previous problem so that it uses a "for" loop, not a "while" loop.

7. What does the following program do:

```
d = dict()
while True:
    try:
        line = input("EOF to stop> ")
    except EOFError:
            break
        for i in line:
            d[i] = d.get(i, 0) + 1
u = d.keys()
for i in sorted(u):
    print(i, d[i])
```

8. What does the following program do:

```
def y(n):
    if n < 10:
        return str(n)
    else:
        d = str(n % 10)
        return y(n // 10) + d

print(y(174))
```

9. When given a non-empty list *lst* of integers, the following function is supposed to return the largest element of the list. But it does not work. Please state what is missing and fix the program:

```
def largest(lst):
    x = largest(lst[1:])
    if x < lst[0]:
        return lst[0]
    else:
        return x
```