*These are problems from the first midterm given for this course several years ago I have revised the problems to be Python 3 compatible. I also a question at the end, since I gave this exam earlier in the semester before having covered as much of the material on string as we did this year. As a result, this document is somewhat longer than what you should expect to see on the actual midterm.*

*I will post a file with all the code functions and fragments in this document, so you can find most the answers yourself. However, it is best to first do this under exam conditions, and work the problems on paper in one sitting before turning to Python.*

1. For each of the following expressions, determine both the **type** and the **value**. One or more of the expressions is incorrectly formed, and in that case, you should just say that it is incorrect. By the way, the function round with only one argument rounds a float to the nearest integer and returns the resulting `int`. So for example `round(3.8)` has the value 4.

```
(a) 50//21
(b) 7%4/1
(c) '6'+'3'*5
(d)  6 + '3'*5
(e) 'computer' > 'computes'
(f) (round(math.sqrt(10))>3)  and (2**3 <10)
```

2. Consider the following function definition. (The function abs returns the absolute value of its argument.)

```
def whoami(x,y,z):
    if (x==y) and (y==z):
        return False
    elif (abs(x)==abs(y)) or (abs(x)==abs(z)) or (abs(y)==abs(z)):
        return True
    else:
        return False
```

(a) What are the values of the following expressions?

`whoami(3,0,-3), whoami(2,2,2), whoami(-1,3,2)`

(b) Rewrite `whoami` so that your new function definition does exactly the same thing, but does not use `if, elif,` or `else`. (It can be done in a single very long line, but it is smarter to begin by creating a few variables of boolean type.)

3. Consider the following function definition:

```
def whoami2(n):
    j=1
    while(j<=n):
        if j**2 == n:
            return True
        j+=1
     return False
```

(a) What are the values of the following expressions?

`whoami2(1), whoami2(2*5), whoami2(5//2*8)`

(b) Give a succinct description of what the function `whoami2` does. (**Don't** answer the question by describing the code, with something like 'it assigns 1 to j, then it checks to see if j is less than or equal to the parameter n...' Instead tell me what problem the function is solving. What would be a good name for the function?)

4. Determine the output printed by the following code. (Recall that the print function called with an empty argument list just advances to the next line.)

```
j=1
for r in range(1,6):
      for c in range(r):
          print(j,end=’ ‘)
          j+=r
      print()
```

(Why are the variables `r` and `c` called `r` and `c`?)

5. In this problem you will have to correct the bugs in a program that contains a number of errors.

The gravitational force of the sun on an object of mass m kilograms at a distance of d meters from the sun is:

$$F = 1.33\times10^{12}\times\frac{m}{d^2}$$

where the mass is given in Newtons. (You don't have to know what any of the physics means.) A programmer writes a function that is supposed to return this force, given m and d as arguments.

```
def gforce(m,d)
    print(1.33e12 * m/d*d)
```

(By the way, `1.33e12` is the same thing as `1.33*10**12`.)
(a) The programmer tries to test the function, but when they try to run it they get the message

```
Syntax error
```

highlighting the first line of the code. What is the error, and how does the programmer fix it? (Address only the syntax error; there are more errors on the way.)

(b) After fixing the syntax error, the programmer then tries to call the function with some sample arguments, resulting in the following exchange with the Python Shell:

```
>>> print gforce(1,2000)
1.33e+12
None
>>> print round(gforce(1,2000),2)
1.33e+12

Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    print round(gforce(1,2000),2)
TypeError: a float is required
```

Something must be going wrong. What error in the function definition produced this behavior, and how can you fix it?

(c) Finally, after the problem has been resolved, the programmer noticed that *every* input they tried with m=1 produced the same result 1.33e12, regardless of the value of d. What caused this error, and how can it be fixed?

7. (a) The function defined below takes a string as an argument and prints something (rather than returning a value). What gets printed when you call `design('Boston')`?

```
def design(s):
    for j in range(len(s)+1):
        print(s[:j].lower()+s[j:].upper())
```

NOTE: You might be surprised to discover that the function does NOT cause a runtime error. As you know, an expression like 'cats'[4] causes an 'index out of range' error, because the last character of the string has index 3. However, the slice `cats[4:]` is perfectly legal, and has the empty string as a value.

(b) The function defined below takes a string as an argument and returns a string. What is returned when you call `big_or_little('Boston')`?

```
def big_or_little(s):
    new=''
    for ch in s:
        if ch.islower():
            new+=ch.upper()
        else:
            new+=ch.lower()
    return new
```

NOTE: As you can probably guess, if `t` is a string, then `t.islower()` returns `True` if t consists entirely of lower-case letters, and `False` otherwise. In particular, if t is a single character, then `t.islower()` is equivalent to ‘`a’<=t<=’z’`.