Name: Jakie Yeshayov

Show work for all questions. Submissions will be on Gradescope as a .pdf document.

This homework is meant to help you review for exam 1. Since the exam will be written, I recommend you first attempt all problems *without* running or writing any code on the computer. As with the exam, avoid using python features or syntax not discussed in class, and do not use any built-in python functions which trivialize a problem.

Written HW 1: Intro, Variables, Data Types, Assignment, Printing, Errors, Selection, Intro to Functions, Loops

---

1. What is the final value and type of each of the variables after the following lines of code execute?

```
1  x = 31
2  y = 'internet'
3  z = 2 + x 33
4  x + 10
5  w = 'x'
6  x = y + w
```

w = 'x' | string

x = 'xinternet' | string

y = 'internet' | string

z = 33 | int

2. Circle each line that has an error that would have to be fixed if the program is to execute to the end. For each one, classify them as a syntax error, `NameError`, or `TypeError`. *Only circle lines that would cause the Python interpreter to stop executing.*

```
1  fgh = 14
2  a = 25
3  x = 4a      -> syntax error
4  bb = bb + 1 -> Name error
5  a = 'fgh'
6  bb = a[0] + a[2]
7  ccc = a[3]  -> Index error
8  ddd = a[-1]
```

3. Write code that generates two random reals between 0 and 1 – call them x and y – and then prints the value of $\sin(2\pi x)\sqrt{-2 \ln y}$. (Fun fact: this program uses the Box-Muller transform to generate a random sample from the standard Gaussian distribution with mean 0 and standard deviation 1.)

```
import random
import mat

x = random.random()
y = random.random()

print((math.sin(2*math.pi*x))*(math.sqrt(-2*math.log(y))))
```

4. Evaluate all the following logical expressions, WITHOUT using your compiler, by reducing the expressions one bit at a time (as we did in class).

a. `not (2>4) and (5<5)`

true and false

false

b. `( (1>2) or (2>3) ) and ('a'>'A')`

false and anything = false

c. `(1!=2) or (1 < 2) and (1==2)`

true and false

true or false = true

5. Determine the output of the following code:

```
1  first = 1
2  second = -3
3  comparison = first < second
4  if comparison:
5      print(first, second)
6      if first > 0:
7          print(first, first)
8  else:
9      if second < 0:
10         print(second)
11     else:
12         print(first)
13 print(second, first)
```

Output: -3 1

(space)

6. On a given day, whether I run outside, run on the treadmill, or rest depends on several factors:

   - If it is **NOT** Tuesday, Thursday, Saturday, or Sunday, then it is a rest day, and I do not run.
   - If it is a running day and 80 degrees Fahrenheit or below, I run outside.
   - If it is a running day and above 80 degrees Fahrenheit, then I run on the treadmill.

   Assume that the variables `day` and `temp` have already been obtained from the user input and converted to the appropriate data types if needed. The variable `day` is one of `'mon'`, `'tue'`, `'wed'`, `'thu'`, `'fri'`, `'sat'`, `'sun'`, and the variable `temp` is any numerical value.

   The following program attempts to output `outside`, `treadmill`, or `rest` according to the criteria above, but it has a bug. (Do NOT attempt to fix it or rewrite the program!)

```
1 if day == 'mon' or day == 'wed' or day == 'fri':
2     print('rest')
3 if temp <= 80:
4     print('outside')
5 else:
6     print('treadmill')
```

   (a) Give a *specific example* of inputs for `day` and `temp` that would result in the **correct** output by the program. What does the program print for your values?

   day = 'tue' prints 'outside'
   temp = 78

   (b) Give a *specific example* of inputs for `day` and `temp` that would result in **incorrect** output by the program. What should the output be, and what does the program print?

   day = 'mon' correct = 'rest'
   temp = 78 actual = 'rest
   outside'

7. If the number of items bought is less than 5, then the shipping charges are $5.00 for each item bought; if the number of items bought is at least 5, but less than 10, then the shipping charges are $2.00 for each item bought; if the number of items bought is at least 10, there are no shipping charges. Write code that allows the user to enter the number of items, and them prints out the shipping charges.

```
items = int(input("How many # of items."))
if (items <5):
    print(f'Your shipping charge is {(5*items):.2f}.')
elif(5<= items <10):
    print(f'Your shipping charge is {(5*items):.2f}.')
else:
    print('Your shipping charge is 0 dollars.')
```

8. Continue the code below, which gets three integers entered on a single line separated by spaces as input from the user. Write code that determines whether there is a repeated number among the three entered numbers. If there is, your code should output `repeat`; otherwise, *it should do nothing.*

   Your code should **not** ask for more input and use the input provided on a single line separated by spaces, obtained from the line of code as follows:

   ```
   three_nums = input('Enter three numbers:  ')
   ```

   A sample run might look like this: `Enter three numbers:  10 3 5321`

   where `10 3 5321` are input from the user, and nothing is output since there are no repeats; or

   ```
   Enter three numbers: 15 323 15
   repeat
   ```

   where `15 323 15` are input from the user, and `repeat` is output since `15` occurs twice in the input.

   num_list = list(three_nums)
   counter = False
   for x in num_list:
   if

   num_list = three_nums.split()

   if (num_list[0] == num_list[1] or num_list[1] == num_list[2] or num_list[2] == num_list[0]):
   print('repeat')

9. Define a function `funSquareOfSum` that takes two input parameters `x` and `y`. The function returns the square of their sum, i.e., the value of $(x + y)^2$ if $x > y$, and otherwise returns 0.

   For example,

   ```
   print(funSquareOfSum(1,2))
   ```

   should result in `0` being printed, and

   ```
   print(funSquareOfSum(2,1))
   ```

   should result in `9` being printed.

   def funSquareOfSum(x,y):
   if (x>y):
   return (x+y)**2
   else:
   return 0

10. Write a program that defines a function, asks the use for a numerical input, and prints `'The functions value at <input> is f(<input>)'` based on the piecewise function. This should also tell the user if their input has no output value.

$$f(x) = \begin{cases} -2x + 11 & x > 10 \\ \dfrac{e^{3x}}{\sin(3x)} & x < 0 \\ 7 & 0 \leq x \leq 10 \end{cases}$$

```
import math
def f_x (x):
    if (x >10):
        return (-2* x) +11
    elif( 0 <=x  and  x<=10):
        return (math.e **(3*x))/math.sin(3*x)
    else:
        return 7
```

11. Consider the following code chunk. What is the output of the following function.

```
1 my_list = [1, 'blue', 5, 'red', 3, 2]
2 for i in range(4):
3     my_list[i] = my_list[i] * 2
4 print(my_list)
```

answer/output

[2, 'blueblue', 10, 'redred', X, X]

3 2

12. Write code which asks the user to input a positive integer `n`, and then prints out the sum of all the **positive odd integers LESS than** `n`. (For example, if the user had entered '8', the program should print out '16', because 1 + 3 + 5 + 7 = 16.)

```
n = int(input("Gimme positive int."))
total = 0
for i in range(n):
    if (i%2!=0):
        total += i

print(total)
```

13. Suppose the variables `m` and `n` have already been defined. Write code to print out the sum of all numbers that are either a multiple of 3 or a multiple of 7 and greater than or equal to `m` and strictly less than `n`.

```
%7==0  %3==0   ≥m   <n
total=0
for i in range(n):
    if (i>=m):
        if(i%7==0 or i%3==0):
            total += i

print(total)
```

14. Assume that a positive integer variable `n` greater than or equal to 2 has already been obtained from user input, as well as a variable `x`, which is less than or equal to `n`.

    You are playing a game of chance where you spin a roulette wheel with `n` equally probably sections, labeled 1 through n. If you spin the wheel and land on

    - `x`, you win `3*x` dollars;
    - i, for any i > 1 and not equal to `x`, you win i dollars;
    - 1, you win nothing and have to pay $3 for playing.

    Use 50,000 simulations of the game to estimate the probability that you end up winning $5 or more. Print the estimate of the probability at the end of your code. A single play consists of spinning the wheel one time.

No credit for theoretical solutions – you must use a simulation.

Assuming X != 1

~~import random~~
~~lim = 50 000~~
~~for i in range(lim):~~
~~y = random.randrange(1, n+1)~~
~~if~~

```
import random
fiveCount = 0
for i in range(50000):
    y = random.randrange(1,n+1)
    if(y==1):
        winnings = -3
    elif(y==X):
        winnings = 3*X
    else:
        winnings = y
    if(winnings >=5):
        fiveCount += 1
print(fiveCount/50,000)
```