CMPSCI 119 MIDTERM #2

UMass Spire ID Number

Enter your name as: **Last, First**, as in: **Smith, Fred**

You may use any of your notes, the *Computer Science Companion*, or any materials on the class Web site.

NOTE: Your **LAST** submission will be the one graded. You get ONE FREE RESUBMIT in case of an "OMG, I screwed up an answer" moment. However, there will be a 5-point penalty assessed for the third submission, a 10-point penalty for the fourth submission, a 15-point penalty for the fifth submission, and so on. You will **NOT** receive an immediate feedback of your score when you submit the exam. Make sure it is right before you hit the SUBMIT button!

Click here to Submit Midterm 2 Information

Question #1 - 20 Points (1 point per answer)

Give the result for each of the following Python expressions. You do not have to specify the data type, but make sure that strings are surrounded by quotes (single or double), lists are surrounded by square brackets, tuples by parentheses, and dictionaries by curly braces.

Make certain that floats contain a decimal point, and if a float is a whole number it must end with both a decimal point and a trailing zero. For example, you must enter the result to the expression `6 / 2` as `3.0` and not as `3.` (no trailing zero) or as `3` (int rather than float).

Make sure as well that in case of string answers you be VERY CAREFUL about upper-case vs. lower-case letters. If, for example, the expected answer is `"Dog"` then answering `"dog"` will be marked as wrong.

Answer **ERROR** if an expression cannot be legally computed by Python.

A few variables have been given values as follows:

```
X = 5
Y = 2.5
L = [4,6,2,9,1]
S = "doghouse"
T = (8,2,4,1,6,3)
D = {5:6.7, 2:8.0, 3:-6.1}
```

Q	PROBLEM	ANSWER

1.	`6 / 2`	`3.0 (freebie)`
2.	`6 // 2`	`3`
3.	`X // 3`	`1`
4.	`X % 3`	`2`
5.	`Y * 2`	`5.0`
6.	`chr(65)`	`"A"`
7.	`ord("B")`	`66`
8.	`S[15]`	`ERROR`
9.	`len(T)`	`6`
10.	`L[0]`	`4`
11.	`L[-1]`	`1`
12.	`L[1:3]`	`[6,2]`
13.	`L[:2]`	`[4,6]`
14.	`D[2]`	`8.0`
15.	`D[0]`	`ERROR`
16.	`D[2]+Y`	`10.5`
17.	`L+T`	`ERROR`
18.	`[0 for M in range(3)]`	`[0,0,0]`
19.	`[M*M for M in [4,1,2]]`	`[16,1,4]`
20.	`[M for M in "Frog"]`	`["F","r","o","g"]`

Question #2 - 10 Points (1 point per answer)

In the Python assignment statement **x = _____** what would I write in the blank to assign to **x** each of the following things?

In your answers, please do not put in blanks where they aren't needed. For example, enter **[]** for an empty list, not **[ ]**.

Q	PROBLEM	ANSWER
1.	An empty list	[] (freebie)
2.	An empty dictionary	{}
3.	The fifth item from list L	L[4]
4.	The last item from list L (without using len)	L[-1]
5.	A list of ints from 1 through 100 (inclusive)	range(1,101)
6.	A list of 1000 zeroes (without using a list comprehension)	[0]*1000
7.	A list from 50 up to 100 by 3s	range(50,100,3) or range(50,101,3)
8.	The integer portion of Z divided by 7.0	int(Z/7.0)
9.	The length of list L	len(L)
10.	The 3rd through 7th characters of string S	S[3:8]

Question #3 - 10 Points (1 point per answer)

For problems invoving color or graphics, use the **[R,G,B]** three-element list representation and the dictionary canvas definition that we've using in the graphics projects. For problems involing polynomials, use the list-of-coefficients technique we explored from the book.

Q	PROBLEM	ANSWER
1.	**What color is [255,255,0]?**	**Yellow**
2.	**In HTML, RebeccaPurple is defined as #663399. What is that in [R,G,B] notation?**	**[102,51,153]**
3.	**What is the result of polyAdd([1,6,0,-3], [2,-3,4])?**	**[3,3,4,-3]**
4.	**X = [] while (len(X) < 5): X = X + [0] What is X afterwards?**	**[0,0,0,0,0]**
5.	**X = [] while (len(X) > 5): X = X + [0]**	**[]**

	What is X afterwards?	
6.	X = [] if (len(X) < 5): X = X + [0] What is X afterwards?	[0]
7.	X = [] if (len(X) > 5): X = X + [0] What is X afterwards?	[]
8.	X = [] for I in range(3): X = [I] + X What is X afterwards?	[2,1,0]
9.	X = 3 Y = 4 X,Y = Y,X Z = str(X)+","+str(Y) What is Z afterwards?	"4,3"
10.	X = lambda A : A*A What is X(4)?	16

Question #4 - 10 Points

Rewrite the following **for**-loop as a **while**-loop:

```
for X in range(7,19,4):
    print (X*X)
```

```
X = 7
while (X < 19):
    print (X*X)
    X = X + 4
```

Question #5 - 10 Points

Here are two functions. The purpose of function **Frog** is to set a variable **X** to a value passed in through parameter **N**. The purpose of function **Toad** is to print the value of that same variable **X**. However, something is not right. One of the functions crashes and the other doesn't seem to do anything.

```
def Frog (N):
    X = N
    return

def Toad ():
    print (X)
    return
```

A. (5 points) Which function crashes, **Frog** or **Toad**?

B. (5 points, short answer) What do I add, and where, to make the two functions work as intended?

A: Toad crashes (It references X, which does not exist in that function. Frog does not crash, but simply sets local variable X to N and then discards that value on return.)

B: Add the statement global X after each def.

Question #6 - 10 Points

Write a function called **Frog** with one integer argument **N** that returns the string **Ribbet** if **N** is less than 100, that returns the string **Chugarump** if **N** is greater than 200, and returns the string **Croak** otherwise.

```
def Frog(N):
    if (N < 100): return "Ribbet"
    if (N > 200): return "Chugarump"
    return "Croak"

-or-

def Frog(N):
    if   (N < 100): Result = "Ribbet"
    elif (N > 200): Result = "Chugarump"
    else: Result = "Croak"
    return Result
```

Question #7 - 10 Points

The **WriteBMP** code in the *Companion* calls the **WriteBytes** function, but that function as written below contains an error (the **Outfile.write** statement is indented one level too far):

```
def WriteBytes (Outfile,N,TotalBytes=1):
    L = []
    for I in range(TotalBytes):
        L = L + [N % 256]
        N = N // 256
        Outfile.write(bytes(L))
    return
```

When the code is written *correctly*, the call:

```
WriteBytes(Outfile,3567,4)
```

would write the list **[239,13,0,0]** to the file.

A. (5 Points) What is written to the file by that same call when the function still contains the error?

B. (5 Points, short answer) How is the correct version of **WriteBytes** like a base conversion problem?

A: [239,239,13,239,13,0,239,13,0,0]

B: You are converting a number to base-256 by dividing it up byte-by-byte.

Question #8 - 10 Points

Short answer

When opening a file for writing (with the **`open`** function), what is the difference between **`"w"`** and **`"wb"`** file modes? When would you use each one?

The "w" form is for text files only, where the values written to the file are represented by readable character strings.

The "wb" form is for binary files, where the values written to the file are one-byte integers and may take on any value in 0…255 in any order, according to the required format of the file being constructed.

Question #9 - 10 Points

We've looked at two Python representations for polynomials: the list of coefficients (as described in the *Companion*), and a dictionary where each key is an exponent and its corresponding value is the coefficient. How would I represent the polynomial:

$$3x^5 + 2x^3 - x^2 + 7x - 4$$

under each interpretation? In the box below, on the first line show the list-of-coefficients representation (5 points), and on the second line show the dictionary representation (5 points).

```
List of coefficients (low-order coefficient
first): [-4,7,-1,2,0,3]

Dictionary: {5:3, 3:2, 2:-1, 1:7, 0:-4}

Notice that there is no X⁴ term.
```