

**CMPSCI 119**  
**Spring 2018**  
**Monday, February 26, 2018**  
**Midterm #1 Solution Key**  
**Professor William T. Verts**

<1> 20 Points – Do any 20; do more for extra credit. Correct answers are worth +1 point, blank answers are worth 0 points, but wrong answers are worth a -½ point penalty; if you don't know an answer, leaving it blank is usually better than a bad guess. The following statements have all been executed:

```
Bucky = 15
Steve = 3.5
Tony = [3, "LOKI", 7.5, "HAWKEYE"]
Nat = ["TESSERACT", [4,1,6], "SCARLET"]
Nick = (5, 10, "HYDRA", 4.5)
```

Show the computed result for each problem; all are independent of one another. Indicate where a computation fails because of some form of error. Be careful about the *type* of the result, particularly int, float, long, bool, and complex types, and put proper quotes around string results (can be " or '), square brackets around lists, and parentheses around tuples. (Types are not required in the answers.)

	Question	Answer
1.	Bucky + 3	18 (int)
2.	Bucky / 2	7 (int)
3.	Steve * 2	7.0 (float)
4.	len(Tony)	4 (int)
5.	len(Tony[0])	ERROR
6.	Tony[-1]	"HAWKEYE" (string)
7.	Nat[0][6]	"A" (string)
8.	Nat[1][2]	6 (int)
9.	Nat[-1][8]	ERROR
10.	Steve + Bucky	18.5 (float)
11.	Nat[1] + [7]	[4,1,6,7] (list)
12.	Nat[1] + 7	ERROR
13.	Nick[1]	10 (int)
14.	Nat + Nick	ERROR
15.	range(Nick[0])	[0,1,2,3,4] (list)
16.	range(1, Bucky, Tony[0])	[1,4,7,10,13] (list)
17.	range(len(Nick[2]))	[0,1,2,3,4] (list)
18.	range(3) + Nat[1]	[0,1,2,4,1,6] (list)
19.	Nat[0] + "S"	"TESSERACTS" (string)
20.	[0 for X in range(Bucky/3)]	[0,0,0,0,0] (list)
21.	[X+1 for X in [2,8,3]]	[3,9,4] (list)
22.	[X for X in Tony[1]]	["L","O","K","I"] (list)
23.	[X*X for X in range(-1,2)]	[1,0,1] (list)
24.	Bucky + (Steve > 1.2)	16 (int)
25.	Bucky + Tony	ERROR

- <2> 10 points – Show what is printed by the following code fragment for each given case (2 points each answer):

<pre> if (N &gt;= 90):     print "A" elif (N &gt;= 80):     print "B" elif (N &gt;= 70):     print "C" elif (N &gt;= 60):     print "D" else:     print "F"                     </pre>	Case #1: N=87	<b>B</b>
	Case #2: N=62	<b>D</b>
	Case #3: N=95	<b>A</b>
	Case #4: N=40	<b>F</b>
	Case #5: N=75	<b>C</b>

- <3> 15 Points – Convert the following for-loop into an equivalent while-loop:

```

for X in range(5,24,7):
    print X*X
    
```

```

X = 5 # 3 points
while (X < 24): # 5 points
    print X*X # 3 points
    X = X + 7 # 4 points
# Remove 1 point per syntax error
    
```

- <4> 20 Points – Show what is printed out as the result from calling **Main()** (four lines total) (5 points each answer):

```

def F1 (X,Y,Z):
    W = X * Z
    return Y + W

def F2 (A,B,C):
    return F1 (C,A,B)

def F3 (L,M,N):
    return F2 (M,N+2,L-1)

def Main():
    print F1 (4,3,2) # Answer #1: 11
    print F2 (2,1,5) # Answer #2: 7
    print F3 (5,2,1) # Answer #3: 14
    print F1 (2,4,-1) # Answer #4: 2
    return
    
```

- <5> 15 Points – The code below contains syntax errors. Locate each one and indicate what the correction(s) should be. Don't rewrite any code statements; just correct the mistakes. Remove 1 point for each error not found, remove ½ point for each error properly identified but not properly corrected, remove ½ point for each error not found, and remove ½ point for each correct item misidentified as an error, but do not go below zero. I believe there are 11 errors.

<pre> define Frog (P,Q):     Print P + Q return         </pre>	<pre> define → def Print → print Must indent return         </pre>
<pre> def Main():     K = input("Enter a number --- ")     for I <del>from</del>in range(10):         for J in [9,1,4]:             if (I+J == K):                 Frog(I,J,<del>K</del>)     <u>Return</u>         </pre>	<pre> Missing () Missing quotes from → in Missing [] = → ==, missing : Too many params Return → return         </pre>

- <6> 20 Points – The following code loads in a graphic picture from a file. Finish the function by doing the same process to each pixel **PX**, as follows: Save the red value (2 points). Set the new red value to the average of the green and blue values (5 points). Set the green value to the saved red value (5 points). Set the blue value to 255 if the saved red value is greater than 128, but set it to zero otherwise (8 points). Give partial credit as appropriate. Allow answers different from below as long as they compute the correct values. Remove 1 point per syntax error.

```

def Main():
    Filename = pickAFile()
    Canvas = makePicture(Filename)
    show(Canvas)
    for Y in range(getHeight(Canvas)):
        for X in range(getWidth(Canvas)):
            PX = getPixel(Canvas,X,Y)

            SaveRed = getRed(PX)
            setRed(PX, (getGreen(PX) + getBlue(PX))/2)
            setGreen(PX, SaveRed)
            if (SaveRed > 128):
                setBlue(PX,255)
            else:
                setBlue(PX,0)

        repaint(Canvas)
    return
        
```