

Instructions:

- Please write clearly. What I cannot read, I will not grade.
- Show all your work in detail. I give partial credit.
- This exam has 8 pages including the title page. Please check to make sure all pages are included.
- This exam is closed book, closed notes, no calculators.
- Don't get bogged down on any one question. You will have 50 minutes to complete this exam.

I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community.

Signature: _____

Question	Points	Score
1. Vocabulary	12	
2. Searching	8	
3. Map, Filter & Reduce	6	
4. Mystery Code	4	
5. CountBright	8	
6. BigNumbersOnly	12	
Total:	50	

1. For each of the following vocabulary terms, write a concise 1-2 sentence definition. Be brief, and to the point.

- (a) (3 points) aliases

Solution: Two variables that point to the same data.

- (b) (3 points) compound data type

Solution: A data type that can hold primitive data types, such as a list, tuple, or dictionary.

- (c) (3 points) exception

Solution: An error that occurs at runtime.

- (d) (3 points) global variable

Solution: A global variable is a variable that can be seen (is visible) through a program module. Defined outside of all functions.

2. (a) (2 points) Here is a sequence of numbers: 1, 2, 3, 6, 8, 13, 15
If you were to perform a linear search on the above sequence, looking for the number '8', in what order does the computer search through the items in the sequence? Write them below.

Solution: 1,2,3,6,8

- (b) (2 points) If you were to perform a binary search on the original sequence for the number 8, in what order does the computer search through the items in the sequence? Write each item that is examined in the order that it is examined below.

Solution: 6, 13, 8

- (c) (2 points) What advantage does the binary search give us?

Solution: Two point answers: -By cutting the list in half each time, it saves a lot of time.
-It is $O(\log n)$, which executes much faster than a linear search.

2 points if they state that it's faster or saves time combined with WHY (cutting in half). Or know that it's $\log N$ and that means it goes faster. 1 point if they just say it's faster but not why...)

1 point answers: -It saves time -It's faster.

- (d) (2 points) Why might we use a linear search instead of a binary search?

Solution: Two point answers: -Because the list of numbers is not sorted. -Because it would take longer to sort a list of unsorted numbers than to just do the search.

Code Understanding

3. (6 points) After the following code runs, what would be the contents of varB, varC, and varD?

```
a = ['a', 'bb', 'ccc', 'dddd', 'eeee' ]

varB = map( lambda x: len(x), a )
varC = filter( lambda x: x%2 == 0, varB)
varD = reduce( lambda x,y: str(x) + str(y), varC)
```

varB = _____

varC = _____

varD = _____

Solution: varB = [1,2,3,4,5] varC = [2,4] varD = '24'

Grading: 2 points for each correct answer. -1 point for missing brackets on the lists or quotes on the string.

4. (4 points) Examine the following code:

```
def mystery( aParameter):
    try:
        aParameter = int(aParameter)
        result = aParameter ** 2
    except:
        result = None
        print "Sorry, aParameter is invalid!"
    return result
```

a) What does this mystery function try to do?

b) Briefly explain why the try/except block is needed:

c) Give a sample function call to the mystery function which would return None:

```
>>>
```

d) Give a sample function call to the mystery function with a valid input that does not trigger the try/catch. Indicate what the expected return value is.

```
>>>
```

Solution:

Answer: a) The function is trying to square the aParameter, after converting it to an integer.
b) A non-numeric parameter, or a string that is not all digits may be entered as a parameter. We need the try/except block to prevent the exception caused by the int() method from causing our program to stop executing. c) mystery([1,2,3]) would return None and print an error. d) mystery(5) or mystery('5') would work correctly (returning 25)

Grading:

1 point for each correct part.

Code Writing Questions

5. (8 points) Write a function named `countBright` that accepts a picture as a parameter. It should *return* the number of pixels in the picture where the pixel's Red, Green, and Blue values sum to 500 or more. For example, if a pixel has (200,100,200) for its color values, it should be counted because (200+100+200) is 500. A pixel with values of (0,255,0) should NOT be counted, because (0 + 255 + 0) is less than 500.

Some functions that might be helpful for this question:

- `getPixels(picture)`
- `getPixel(picture, x, y)`
- `getWidth(picture)`
- `getHeight(picture)`
- `getRed(pixel)`
- `setRed(pixel,value)`

Solution:

```
def countBright( aPic):
```

```
x = 0
for pix in getPixels(aPic):
    r = getRed(pix)
    g = getGreen(pix)
    b = getBlue(pix)
    # r,g,b = getRGB(pix)    #also acceptable
    if (r + g + b) >= 500:
        x = x+1
return x
```

Grading: 1 point for correct header, 1 point for looping through all pixels in the image. 2 points for getting the R/G/B values out of the pixel. 1 point for summing them together. 1 point for comparing greater than OR EQUAL to 500 (no point if they forget the or equal to part). 1 point for correctly keeping a count and 1 point for returning the correct value.

6. (12 points) Write a function named `bigNumbersOnly` that accepts the name of a file to open as a string parameter. The function should open the file, which will be formatted as follows (one number per line):

```
54
4.0
23.3
765.2
54
876.8
34
238
45
50
```

Your function must read in the file and return a *list* of all the numbers greater than 50 (numbers should be stored as floats). Remember to close your file after you are finished reading it. If you find duplicate numbers greater than 50, DO NOT include duplicates in the list!

Solution:

```
def bigNumbersOnly( filename ):
    f = open(filename, "r")
    lines = f.readlines()
    f.close()

    myList = []
    for line in lines:
        num = float(line)
        if num > 50 and (not (num in myList)): #Also works: if num > 50 and num not in myList
            myList.append(num)
    return myList
```

Grading: 1 point for correct header, 1 point for opening file, 1 point for using "read" mode. 3 points for reading everything from the file (one read, or readline() in a loop, or readlines()). 1 point for closing the file. 2 points for converting the numbers to float. 1 point for testing against 50, 1 point for checking for duplicates, 1 point for returning the list.

This page intentionally left blank.