

15110 PRINCIPLES OF COMPUTING – SAMPLE LAB EXAM 2

Directions:

1. Open the terminal window and create a private directory named `samplelabexam2`. Then go into this directory.

```
mkdir samplelabexam2
cd samplelabexam2
```

2. Write a function in Python for each of the following problems using `gedit` (or another text editor), and store these functions in `samplelabexam2` folder. For example, for question 1:

```
gedit target.py &
```

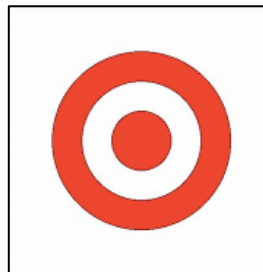
3. For each question, once you save your function in the editor, test your function in the terminal window by calling it within `python3`. Although we give you sample test runs, your function should work completely based on the given specifications and your output should match the sample usage as closely as possible for full credit. We will test your functions on additional test cases. Exit `python3` using `quit()`.

```
python3 -i target.py
```

4. Once you are finished, compress the `samplelabexam2` folder into a zip file and then you would submit `samplelabexam2.zip` to `autolab.andrew.cmu.edu` by the end of lab if this were the real exam. Do not delete the `samplelabexam2` folder from your account.

```
cd ..
zip -r samplelabexam2.zip samplelabexam2
```

1. (25 pts) Write a Python function `target(c)` (in the file `target.py` in your `samplelabexam2` folder) that draws a bull's-eye target centered on a canvas `c` in a window of size 450 by 450. The target consists of concentric red and white circles whose radii are 50, 100, and 150 pixels. Note: the order in which you draw the circles matters! The result should look just like the picture below, but if you're off by a pixel, it's okay.



Make sure
your outline
matches your fill
so you don't
have black
outlines!

Sample usage:

```
>>> from tkinter import *
>>> window = Tk()
>>> c = Canvas(window, width=450, height=450)
>>> c.pack()
>>> target(c)
>>> c.update()
```

2. (25 pts) Write a function `strsizes(filename)` (in the file `strsizes.py` in your `samplelabexam2` folder) that opens a file that has one string per line. It returns a list that contains the length of each string not including the newline.

Sample test file `testfile1.txt`:

```
Principles
of
Computing
rocks!
```

Sample usage:

```
>>> strsizes("testfile1.txt")
[10, 2, 9, 6]
```

3. (25 pts) In the file `product.py` in your `samplelabexam2` folder, write a recursive function `product(numlist)` that returns the product of all the numbers in the given list. Do not use a loop; you must use recursion. The product of an empty list is 1 by default. Otherwise, the product of a list is the first element times the product of the rest of the list.

Sample usage:

```
>>> product([2,3,7])
42
>>> product([2,5,2,5])
100
>>> product([])
1
```

4. (25 pts) Write a function `sumints(matrix)` (in the file `sumints.py` in your `samplelabexam2` folder) that takes as input a matrix containing a mix of integers and strings, and returns the sum of the elements that are integers. You may assume that each row of the matrix has the same number of elements in it. Hint: you can use the Boolean expression `type(x)==int` to check if `x` is an integer.

Sample usage:

```
>>> sumints([[5, 1], ["blueberry", 12]])
18
>>> sumints([[1,2,"moose",4],[5,"squirrel","boris",7],["Natasha",10,11,13]])
53
>>> sumints([["north", "west"]])
0
```