

PROBLEM 1 : (*Honor code (2 pts)*)

Print your name to acknowledge the Duke Community Standard

Print your name to say you have read all the rules for the exam and abide to follow them

PROBLEM 2 : (*Short code segments (28 pts)*)

For each of the following problems, use only what is indicated to set result to a Python expression. Do not use any Python methods.

Here is an example.

Use phrase with indexing and the concatenation of two items to set result to the string 'by'

```
phrase = 'bicycle'
result = phrase[0] + phrase[3]
```

Note this answer uses only phrase and indexing, and the concatenation of two items.

PART A (3 pts)

Use phrase with indexing and the concatenation of three items to make the string 'egy'

```
phrase = 'The quick brown fox jumps over the lazy dog'
result = _____
```

PART B (3 pts)

Use lst with indexing and concatenation of two items to make the string 'or'.

```
lst = ['computer']
result = _____
```

PART C (3 pts)

Use phrase with splicing and the concatenation of two items to make the string 'adke'

```
phrase = 'DukeRoad'
result = _____
```

PART D (3 pts)

Use lst with indexing and the concatenation of two items to make the string 'catant'

```
lst = [['dog', 'cat'], ['ant', 'fox']]
result = _____
```

PART E (3 pts)

Use `lst` with indexing and the concatenation of two items to make the string `'xz'`

```
lst = [['elephant', 'fox','lion'], ['giraffe', 'zebra']]
result = _____
```

PART F (3 pts)

Use `lst` with `len` and indexing to make the number 5 from `lst` (no arithmetic)

```
lst = [['monkey', 'armadillo','turtle'], ['snake', 'dolphin', 'crab']]
result = _____
```

PART G (3 pts)

Use `phrase` with only `join` and `split` to make the new phrase `'bookoopor'`

```
phrase = 'beekeeper'
result = _____
```

PART H (4 pts)

Use `phrase` with only `join` and `split` to make the string `'say*see*hear*do'`

For this problem it is helpful to write this one in two steps with an additional variable we call `resulttmp`

```
phrase = "monkey say monkey see monkey hear monkey do"
resulttmp = _____
result = _____
```

PART I (3 pts)

Use `num` with only `mod` and `divide` to get 34.

```
num = 2346
result = _____
```

PROBLEM 3 : (*Simple Functions (14 points)*)

PART A: TShirts (7 pts)

At the Tshirt Shop you get discounts on orders depending on how many orders you have had and how much you have spent there. If you have had more than 20 orders previously, you get a 10% discount on your current order. If you have had more than 50 orders previously, instead of the 10% discount, you get a 20% discount. However, if you have spent less than 500.00 total including this order, you don't get any discount.

Write the function `tshirtShop` that has three parameters: **price** is a float representing the cost of the current order, **numOrders** is an integer representing the number of orders you have had prior to this order, and **spent** is an integer representing the total amount you have spent at the tshirt shop prior to this order. This function calculates and returns the price of the order, factoring in discounts described above.

call	returns	comment
<code>tshirtShop(22, 51, 470)</code>	22.0	470+22 is not greater than 500, no discount
<code>tshirtShop(22, 51, 490)</code>	17.6	490+22 > 500, more than 50 orders, 20% discount
<code>tshirtShop(22, 31, 550)</code>	19.8	22+550 > 500, 31 > 20, 10% discount

Complete the function `tshirtShop` below.

```
def tshirtShop(price, numOrders, spent):
```

PART B: The Count (7 pts)

For this problem, you are given a phrase and two letters. You are to calculate how many words in phrase start with let1 and end with let2.

Write the function `theCount` that has three parameters: **phrase** is a string of words, all lowercase, **let1** and **let2** are both strings that are a single letter lowercase. This function calculates and returns the number of words in phrase that start with let1 and end with let2.

call	returns	comment
<code>theCount("dog cat bat hare pig horse mouse", "h", "e")</code>	2	hare and horse
<code>theCount("dog cat bat hare pig horse mouse", "p", "g")</code>	1	pig
<code>theCount("sisters sas sarcasm sos ship", "s", "s")</code>	3	sisters, sas and sos

Complete the function `theCount` below.

```
def theCount(phrase, let1, let2):
```

PROBLEM 4 : (*Debugging, mystery and tracing, oh my! (10 pts)*)

PART A) (5 pts)

Consider the following function named `isNumber` that takes as input a string and is supposed to return `True` if the string is composed only of digits, where a digit is 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9, and `False` if it is not. This function does not work correctly.

```
1 def isNumber(word):
2     for ch in word:
3         if ch not in '0123456789':
4             return False
5     return True
```

Shown here are three calls to `isNumber`, the result and a comment about it.

call	returns	comment
<code>isNumber("56a")</code>	<code>True</code>	should return <code>False</code>
<code>isNumber("x")</code>	<code>False</code>	this is the correct answer
<code>isNumber("34")</code>	<code>True</code>	this is the correct answer

a) Give another example of a call to `isNumber` that passes a string of letters and/or digits that results in the wrong answer. The argument passed should be in a different format than `"56a"` which is in the format of a digit, digit, letter.

b) Explain which line is causing the error and what the error is.

c) Explain how to fix the code minimally so it works correctly. That is, do not completely rewrite it.

PART B) (5 pts)

Consider the following function named `mystery` that takes as input a string named `phrase`.

```
def mystery(phrase):  
    lst = phrase.split()  
    total = 0  
    cnt = 0  
    for item in lst:  
        if cnt%2 == 0:  
            total += int(item)  
        cnt += 1  
    return total
```

a) Give an example of a string named `phrase` that has three blanks in it that are not consecutive, and that does not result in an error when `mystery` is called with this string.

phrase is:

result of `mystery(phrase)` is:

b) Give an example of a string named `phrase` that has three blanks in it that are not consecutive, and that does result in an error when `mystery` is called.

phrase is:

Explain why there is an error:

c) Give a general description of what `mystery` does in one sentence (do not just repeat the code).

PROBLEM 5 : (*Dinner Time (23 pts)*)

In this problem we will be processing data related to food so we can answer questions about the food.

Part A (8 pts)

Write the function named **process** that has one parameter named **line**, which is a string that has four pieces of information in the following format: the type of food (one word), followed by a blank, followed by the name of the food (one or more words), followed by a colon, followed by the number of calories in this food (an integer), followed by a dash, followed by the cost of the item (a float). This function should return a list of the four items about that food: the type of food as a string, the name of the food as a string, the number of calories as an integer and the price of the item as a float.

Here are several examples of calls to this function.

call	returns
process('maindish chicken parmesan:1140-8.50')	['maindish', 'chicken parmesan', 1140, 8.5]
process('dessert chocolate ice cream:145-3.50')	['dessert', 'chocolate ice cream', 145, 3.5]
process('maindish lentil soup:430-6.00')	['maindish', 'lentil soup', 430, 6.0]
process('fruit mango:200-1.50')	['fruit', 'mango', 200, 1.5]

Complete the function below.

```
def process(line):
```

Part B (8 pts)

Write the function named **formatList** that has one parameter named **lst** that is a list of strings, where each string is in the format described in Part A. This function returns a list of lists in which each list has the four parts described in Part A. In writing **formatList**, you should call the function **process** from Part A. Assume **process** works correctly regardless of what you wrote in Part A.

For example, assume

```
lst = ['maindish chicken parmesan:1140-8.50',
       'dessert chocolate ice cream:145-3.50',
       'fruit apple:95-1.00',
       'maindish lentil soup:430-6.00',
       'fruit mango:200-1.50',
       'dessert key lime pie:350-4.00',
       'maindish chicken tikka masala:1050-9.70',
       'maindish shrimp kebab:260-11.20',
       'dessert peach cobbler:240-3.00',
       'fruit red grapes:105-0.90',
       'maindish eggplant parmesan:540-7.50' ]
```

Then **formatList(lst)** processes every string in the list above and returns this list of lists:

```
[ ['maindish', 'chicken parmesan', 1140, 8.5],
  ['dessert', 'chocolate ice cream', 145, 3.5],
  ['fruit', 'apple', 95, 1.0],
  ['maindish', 'lentil soup', 430, 6.0],
  ['fruit', 'mango', 200, 1.5],
  ['dessert', 'key lime pie', 350, 4.0],
  ['maindish', 'chicken tikka masala', 1050, 9.7],
  ['maindish', 'shrimp kebab', 260, 11.2],
  ['dessert', 'peach cobbler', 240, 3.0],
  ['fruit', 'red grapes', 105, 0.9],
  ['maindish', 'eggplant parmesan', 540, 7.5] ]
```

Complete the function below

```
def formatList(lst):
```


Part C (8 pts)

Write the function named **maxOfType** that has two parameters, one parameter named **lst** that is a list of lists, where each innerlist has the four items described in Part A, and one parameter named **type** that is a string. This function returns the name of the food from **lst** that is of type **type** and that has the most calories of all items of type **type**. If there is more than one food of that type with the most calories, then return any one of them that has the most calories.

For example, assume **lst** is the following list of lists.

```
lst = [ ['maindish', 'chicken parmesan', 1140, 8.5],
        ['dessert', 'chocolate ice cream', 145, 3.5],
        ['fruit', 'apple', 95, 1.0],
        ['maindish', 'lentil soup', 430, 6.0],
        ['fruit', 'mango', 200, 1.5],
        ['dessert', 'key lime pie', 350, 4.0],
        ['maindish', 'chicken tikka masala', 1050, 9.7],
        ['maindish', 'shrimp kebab', 260, 11.2],
        ['dessert', 'peach cobbler', 240, 3.0],
        ['fruit', 'red grapes', 105, 0.9],
        ['maindish', 'eggplant parmesan', 540, 7.5] ]
```

Here are several examples of calls to this function.

call	returns	comment
maxOfType(lst, "maindish")	"chicken parmesan"	1140 most calories maindish
maxOfType(lst, "fruit")	"mango"	200 most calories fruit
maxOfType(lst, "dessert")	"key lime pie"	350 most calories dessert

Complete the function below.

```
def maxOfType(lst,type):
```