

PROBLEM 1 : (*Types and values? (26 points) (ESTIMATE: 13 minutes)*)

Consider the following variables and their values. List in the table the type of variable and its value after being assigned the expression. The first one is done for you. If the expression causes an error, write 'error' for both type and value.

```
wlist = ['boat', 'sail', ['cool','hot','windy'], 'rope', 'engine']
word = "beautiful"
phrase = "run five miles"
```

variable = expression	Type	Value
a = 'truck'	string	'truck'
b = word[4] + word[-4]		
c = (wlist[-2] + wlist[0]) * 2		
d = len(wlist)		
e = wlist[2][2] + wlist[1][1]		
f = word < "apple"		
g = word[:3] + word[-2:]		
h = wlist[0:2] + wlist[2][2:]		
i = phrase.split()		
j = word.split("t")[1]		
k = "8".join(wlist[2])		
m = 13//5 + 4.2		
n = 7/2 + 4**2		
o = (5 % 3) * 4		

PROBLEM 2 : (*What is output? (10 points) (ESTIMATE: 8 minutes)*)**PART A (2 pts)**

```
lst1 = [9, 5, 2]
lst1.append(8)
print(lst1)
```

Output:

PART B (2 pts)

```
lst2 = [7, 2]
lst2.append([9, 1])
print(lst2)
```

Output:

PART C (2 pts)

```
lst3 = [3] + [5]
lst3 = lst3 + lst3
print(lst3)
```

Output:

PART D (2 pts)

```
lst4 = [6, 5, 2, 3]
lst5 = lst4
lst4[1] = 7
print(lst4)
print(lst5)
```

Output:

PART E (2 pts)

```
lst4 = [6, 5, 2, 3]
lst5 = lst4
lst5[-2] = [4]
lst4 = lst4 + [9]
print(lst4)
print(lst5)
```

Output:

PROBLEM 3 : (*Short code segments (13 pts) (Estimate: 10 minutes)*)

For each of the following **four parts**, use only what is indicated to assign the stated value to **result**. Do not use any Python methods or string constants unless indicated.

Here is an example.

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

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

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

Here is an example of a **WRONG** answer: **result = word[:1] + 'y'** This answer is wrong because it uses splicing and a string constant and it did not use indexing with **word**.

PART A (3 pts)

Use **lst** with indexing and the concatenation of two items to set **result** to the string 'it'

```
lst = ['danish', 'spruce', 'bot']

result =
```

PART B (3 pts)

Use **word** with splicing and concatenation of two items to set **result** to the string 'scran'.

```
word = "manuscript"

result =
```

PART C (3 pts)

Use `lst` with indexing and splicing to set `result` to the string `'rich'`

```
lst = [['rug', 'carpet'], ['light', 'fan'], ['ostrich']]
```

```
result =
```

PART D (4 pts)

Use string `word` below with only `join` and `split` with appropriate small string constants to set `result` to the value `plingplainplayplop`

You can use temporary variable(s) if you want to write the code in more than one line.

```
word = "plingstrainstraystrop"
```

```
result =
```

PROBLEM 4 : (*Do you like books?: (8 points) (Estimate: 8 minutes)*)

Write the function **purchaseBooks** that has one int parameter named **cost** that is the cost of books you want to purchase, and two float parameters, one named **taxPercent** that represents the tax percent applied to the cost, and one named **discount** that represents a discount percent that is applied before the tax is applied. This function returns the final cost of the books, after first the discount is applied and then the tax is applied. Note: if the discount is 10.0 and the taxPercent is 6.0, that means a 10% discount and 6% tax are applied, in that order.

For example, if the books cost \$430, the discount is 10.0 and the taxPercent is 6.0, then the 10% discount is \$43.00, lowering the cost of the books to $\$430 - \$43.00 = \$387.00$. The 6% tax applied to \$387.00 is \$23.22. The final cost of the books is $\$387.00 + \$23.22 = \$410.22$. This example and others are below.

call	returns	comment
<code>purchaseBooks(430, 6.0, 10.0)</code>	363.78	The example above
<code>purchaseBooks(500, 5.0, 5.0)</code>	498.75	discount is 25.00, tax is applied to 475.00, tax is 23.75
<code>purchaseBooks(600, 3.0, 10.0)</code>	556.20	discount is 60.00, tax is applied to 540.00 tax is 16.20

Complete the function **purchaseBooks** below.

```
def purchaseBooks(cost, taxPercent, discount):
```

PROBLEM 5 : (*Give away fruit: (10 points) (Estimate: 8 minutes)*)

Write the function `giveAwayFruit` that has three integer parameters: **amount1** is the number of apples you have, **amount2** is the number of bananas you have, and **limit** is the minimum number of each type of fruit you like to have. This function returns a string that tells you how many of each type of fruit to give away. If the number of apples is greater than **limit**, then give away the difference to get down to **limit**, plus 2 more. If the number of bananas is greater than **limit**, then give away the difference to get down to **limit** plus 4 more. If you have limit or less of a type of fruit, don't give away any of that type. The string returned is the number of apples to give away, followed by a blank, a capital "A", a semicolon, a blank, the number of bananas to give away, followed by a blank and then a capital "B".

For example, if you have 15 apples, 20 bananas and limit is 12, then you give away 3 apples to get to 12, plus 2 more apples. You also give away 8 bananas to get to 12, plus 4 more bananas. The function returns '5 A; 12 B'. Examples are below.

call	returns	comment
<code>giveAwayFruit(15, 20, 12)</code>	'5 A; 12 B'	3 + 2 apples, and 8 + 4 bananas
<code>giveAwayFruit(8, 5, 8)</code>	'0 A; 0 B'	0 apples, and 0 bananas
<code>giveAwayFruit(9, 14, 10)</code>	'0 A; 8 B'	0 apples, and 4 + 4 bananas

Complete the function `giveAwayFruit` below.

```
def giveAwayFruit(amount1, amount2, limit):
```

PROBLEM 6 : (*Random Words and Numbers: (10 points) (Estimate: 8 minutes)*)

Write the function `compose` that has three string parameters named `wordA`, `wordB`, and `wordC`; and two integer parameters named `numA` and `numB`.

This function returns a string that is randomly one of `wordA`, `wordB`, or `wordC`; followed by a random number that is between `numA` and `numB`, inclusively. Assume `numA ≤ numB`.

Here are some examples (there are other possible return values not shown):

call	possible return	another possible return
<code>compose("blue", "green", "red", 5, 20)</code>	<code>'blue8'</code>	<code>'red16'</code>
<code>compose('boat', 'car', 'plane', 80, 89)</code>	<code>'plane89'</code>	<code>'boat83'</code>

Complete the function `compose` below.

```
def compose(wordA, wordB, wordC, numA, numB):
```

PROBLEM 7 : (*What does code do?: (10 points) (Estimate: 8 minutes)*)

Consider the following numbered code that is supposed to print the three numbers passed in in increasing order, but does not work correctly. Then answer the questions below.

```
1 def sortNums(num1, num2, num3):
2     if num1 > num2:
3         if num1 > num3:
4             print(num3, num2, num1)
5         else:
6             print(num2, num1, num3)
7     else:
8         if num1 > num3:
9             print(num3, num1, num2)
10        else:
11            print(num1, num2, num3)
12
```

- A. What is printed with the call `sortNums(5, 8, 4)`
- B. What is printed with the call `sortNums(5, 4, 9)`
- C. What is printed with the call `sortNums(9, 5, 7)`
- D. What is printed with the call `sortNums(4, 8, 7)`
- E. Briefly explain why the function does not always work correctly.
- F. What type of error is this, a syntax error, a runtime error or a semantic error?
- G. Briefly explain in general how to fix the code so it works correctly. DO NOT GIVE CODE.