NETID: _________

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 = 'lion'`	string	'lion'
`b = word[3] + word[-3]`		
`c = (wlist[1] + wlist[-1]) * 2`		
`d = len(wlist)`		
`e = wlist[2][1] + wlist[0][2]`		
`f = word < "car"`		
`g = word[:2] + word[-3:]`		
`h = wlist[1:2] + wlist[2][1:]`		
`i = phrase.split()`		
`j = word.split("u")[1]`		
`k = "#".join(wlist[2])`		
`m = 13//4 + 2.1`		
`n = 4/2 + 3**2`		
`o = (5 % 2) * 3`		

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

PART A (2 pts)

```
lst1 = [7, 4, 1]
lst1.append(4)
print(lst1)
```

Output:

PART B (2 pts)

```
lst2 = [8, 2]
lst2.append([5, 4])
print(lst2)
```

Output:

PART C (2 pts)

```
lst3 = [8] + [9]
lst3 = lst3 + lst3
print(lst3)
```

Output:

PART D (2 pts)

```
lst4 = [7, 2, 9, 5]
lst5 = lst4
lst4[1] = 3
print(lst4)
print(lst5)
```

Output:

PART E (2 pts)

```
lst4 = [7, 2, 9, 5]
lst5 = lst4
lst5[-2] = [8]
lst4 = lst4 + [4]
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 ‘no’

```
lst = [’danish’, ’spruce’, ’bot’]

result =
```

PART B (3 pts)

Use **word** with splicing and concatenation of two items to set result to the string ‘ripus’.

```
word = "manuscript"

result =
```

PART C (3 pts)

Use **lst** with indexing and splicing to set result to the string `‘pet’`

```
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 `boatboyboilbowl`

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

```
word = "boattraytrailtrawl"




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**
purchaseBooks(430, 6.0, 10.0)	363.78	The example above
purchaseBooks(500, 5.0, 5.0	498.75	discount is 25.00, tax is applied to 475.00, tax is 23.75
purchaseBooks(600, 3.0, 10.0	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 : (***Nice cold popsicle: (10 points) (Estimate: 8 minutes)***)

Write the function `restockPopsicles` that has three integer parameters: **amount1** is the number of dairy popsicles you have, **amount2** is the number of nondairy popsicles you have, and **low** is the number of popsicles you want to at least have for each type. This function returns a string that tells you how many of each type popsicle to buy. If the number of dairy popsicles is less than **low**, then buy the difference to get to **low**, plus 10 more. If the number of nondiary popsicles is less than **low**, then restock the difference to get to **low** plus 5 more. If you have low or more of a type, you don't buy that type. The string returned is the number of dairy popsicles to buy, followed by a blank, a capital "D", a semicolon, a blank, the number of nondairy popsicles to buy, followed by a blank and then a capital "N".

For example, if you have 8 dairy, 5 nondairy, and low is 10, then you buy 2 dairy to have 10 dairy, plus you buy 10 more dairy. You also need to buy 5 nondairy to have 10 nondairy, plus you buy 5 more nondairy. The function returns '12 D; 10 N'. Examples are below.

call	**returns**	**comment**
restockPopsicles(8, 5, 10)	'12 D; 10 N'	2 + 10 dairy, and 5 + 5 nondairy
restockPopsicles(12, 15, 10)	'0 D; 0 N'	both are bigger than low
restockPopsicles(5, 9, 8)	'13 D; 0 N'	3 + 10 dairy, and no nondairy

Complete the function **restockPopsicles** below.

```
def restockPopsicles(amount1, amount2, low):
```

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 $\leq$ numB.

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

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

Complete the function **compose** below.

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

NETID: ________

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.