

NAME (print): \_\_\_\_\_

Netid: \_\_\_\_\_

Community Standard Acknowledgement (signature): \_\_\_\_\_

Do NOT spend too much time on any one question.

In writing code you do not need to worry about specifying the proper `import` statements. Assume that all libraries we've discussed are imported in any code you write.

Do **not** discuss this test **with anyone** until the test is returned. Do not use the web, PyCharm, or any programming environment that checks your code.

|           | value   | grade |
|-----------|---------|-------|
| Problem 1 | 3 pts.  |       |
| Problem 2 | 24 pts. |       |
| Problem 3 | 8 pts.  |       |
| Problem 4 | 12 pts. |       |
| Problem 5 | 28 pts. |       |
| TOTAL:    | 75 pts. |       |

**PROBLEM 1 : (*Honor code (3 pts)*)**

Print your name to acknowledge the Duke Community Standard

Print your name again to say you have read all the rules for the exam and agree to follow them (including not using the web, classmates, Pycharm, or other environments to determine answers to your program).

**PROBLEM 2 : (*Short code segments (24 pts)*)**

For each of the following problems, use only what is indicated to write ONE line of code to either assign a value to result or change the specified value. You may use Python methods. However, you cannot simply create a new list, for example, that includes elements that correctly solve the problem.

**Here are two examples.**

Example 1

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. It does not simply assign the string result = 'by'.

Example 2

```
alist = ["cat", "dog", "egg", "bat", "ape", "ant"]
```

In one line of code, change the value of alist using a python method so it's value is now ["cat", "dog", "bat", "ape", "ant"].

Answer: alist.remove("egg")

**PART A (4 pts)**

In one line of code, change the value of numbers using a Python method, such that its value is now ([3,4], [1,2,3,4], [7,5]).

```
numbers =([3,4], [1,2,3], [7,5])
```

**PART B (4 pts)**

Write the command that uses indexing of `z` and concatenation to create the string `'pa'`:

```
z = ('blue', 'tan', 'sip', 'rod')
result = _____
```

**PART C (4 pts)**

Write the command that uses set operations to determine what `s1`, `s2`, and `s3` have in common (that is, `{20}`):

```
s1 = {5,10,15,20}
s2 = {10,20,30,40}
s3 = {20,40,60,80}
result = _____
```

**PART D (4 pts)**

Write the command that adds 3 to each element from `lst` and stores those numbers to create the list named `result` that contains `[3,4,5,8,10]`:

```
lst = [5,2,7,1,0]
result = _____
```

**PART E (4 pts)**

Given the tuple `a`, write the command that uses indexing of `a` to create the list named `result` that contains `[[3, 4], [7, 5]]`.

```
a = ([3,4], [1,2,3], [7,5])
result = _____
```

**PART F (4 pts)**

Given the dictionary `d`, use Python methods to create the list named `result` as a sorted list of the dictionary's keys. In this case, `result` would be `['b', 'd', 'l', 'm', 't']`.

```
d = {"t":13, "b":4, "l":21, "m":1, "d":3}
result = _____
```

**PROBLEM 3 :** (*Simple Functions (8 points)*)

**Sorting Integers (8 pts)**

Given a list of integers, create the function called `sortNumbers` that takes one parameter: `values` is a list of integers. There must be at least one integer stored in `values`.

The function `sortNumbers` determines the integer remainder of each element divided by 4 and returns a list of those remainders in descending order.

This function must use a list comprehension in solving the problem.

| <b>call</b>                         | <b>returns</b>         | <b>comment</b>   |
|-------------------------------------|------------------------|--|
| <code>sortNumbers([5,1,2,7])</code> | <code>[3,2,1,1]</code> | The remainder of 5 divided by 4 is 1, 1 divided by 4 is 1, 2 divided by 4 is 2, and 7 divided by 4 is 3. These remainders are then returned in a list, sorted in descending order. |
| <code>sortNumbers([16,11,3])</code> | <code>[3,3,0]</code>   | The remainder of 16 and 4 is 0, 11 divided by 4 is 3, and 3 divided by 4 is 3.   |
| <code>sortNumbers([6,5,4,3])</code> | <code>[3,2,1,0]</code> |  |

Complete the function `sortNumbers` below.

```
def sortNumbers(values):
```

**PROBLEM 4 :** (*Reading and Debugging Programs (12 pts)*)

**PART A) (6 pts)**

The following program contains list of dates, where each element is stored in the format (month, day, year). The program is supposed to sort the list of dates by year(descending), month, then day:

```
1 lst = [(4,10,2021), (12,25,2007), (10,31,2020), (2,16,2019), (5,25,2020), (2,14,2021)]
2 lst2 = sorted(lst, key=lambda x: x[0])
3 lst3 = sorted(lst2, key=lambda x: x[1])
4 lst4 = sorted(lst3, key=lambda x: x[2], reverse=True)
5 print(lst4)
```

a) What is the value of lst4 when the program completes?

b) The value of lst4 was supposed to be the following:

```
[(2,14,2021), (4,10,2021), (5,25,2020), (10,31,2020), (2,16,2019), (12,25,2007)]
```

What line(s) of code is/are producing this error?

c) How would you modify the line(s) of code identified in part b) to make this provide the correct output?

## PART B) (6 pts)

The following code first stores the number of students who have declared their major as African & African American Studies ('AAAS'), Computer Science ('CS'), Electrical & Computer Engineering ('ECE'), and Biology ('BIO'). The program then adds 100 students who are double majors to both Biology and Computer Science.

```
1 majors = [('AAAS', 321), ('ECE', 400), ('CS', 900), ('BIO', 750)]
2 majors[-1][-1]+=100 #add 100 to the number of Biology majors
3 majors[-2][-1]+=100 #add 100 to the number of Computer Science majors
4 print(majors)
```

This program doesn't execute. It produces a TypeError: 'tuple' object does not support item assignment

a) What lines of code create this error?

b) What is the reason for the error(s)?

c) Assuming you delete the lines of code you identified in part b, what one line of code can replace them to correctly create a new entry for ('BIO/CS', 100) in the list?

**PROBLEM 5 :** (*Creating Functions (28 pts)*)

**PART A) (14 pts)**

We are interested in identifying the state (or DC) with the most historically Black colleges and universities (HBCUs). Create the function `countSchools` that has one parameter `hbcus`, which is a list of strings, where each string is school information representing the name of the school and the state the school is located, in the format 'School:State'.

Assume the list `hbcus` contains at least one school. The function returns the state with the most HBCUs. Assume (for simplicity), that there will only be one state with the highest number of HBCUs.

| call   | returns | comment  |
|--|---------|--|
| <code>countSchools(['NCCU:NC', 'Morehouse:GA', 'Howard:DC', 'Spelman:GA'])</code>                                | 'GA'    | GA has 2 schools, NC and DC have 1 each. However, that doesn't matter. |
| <code>countSchools(['NCCU:NC', 'Morehouse:GA', 'Howard:DC', 'Spelman:GA', 'Shaw:NC', 'Clark Atlanta:GA'])</code> | 'GA'    | GA has 3, NC has 2, and DC has 1.                                      |
| <code>countSchools(['Bowie:MD', 'Coppin:MD', 'Morgan:MD'])</code>  | 'MD'    | MD has 3.  |

Complete the function `countSchools` below.

```
def countSchools(hbcus):
```

## PART B) (14 pts)

Krispy Kreme has a fundraiser competition to see which university in Durham and Chapel Hill (Duke, NCCU, and UNC) sells the most boxes. Each university provides weekly updates to Krispy Kreme on the total number of boxes sold. At the end of the program, the university that sold the most boxes is the winner.

Create the function `topSeller` that takes two parameters. The parameter `orders` is a list of tuples of all weekly totals for all schools, where each tuple is a string and an integer in the format ('School', # boxes). The parameter `topNum` is an integer that is the minimum number of boxes to sell to be considered a top-selling school.

The function then determines and returns a sorted list of all schools that sold at least `topNum` boxes.

Constraint: Assume the list `orders` has at least one school with at least one box sold.

| call   | returns                              | comment  |
|--|--------------------------------------|--|
| <code>topSeller([( 'Duke',6), ( 'NCCU',7), ( 'UNC',1)], 7)</code>                              | <code>['NCCU']</code>                | NCCU sold at least 7 boxes.                              |
| <code>topSeller([( 'Duke',3), ( 'NCCU',3), ( 'UNC',7), ( 'UNC', 8), ( 'NCCU', 10)], 10)</code> | <code>['NCCU', 'UNC']</code>         | UNC and NCCU sold at least 10 boxes (UNC sold the most). |
| <code>topSeller([( 'Duke',10), ( 'NCCU',11), ( 'UNC', 12)], 9)</code>                          | <code>['Duke', 'NCCU', 'UNC']</code> | All three schools sold at least 9 boxes.                 |

Complete the function `topSeller` below.

```
def topSeller(orders, topNum):
```