CSCI.UA.0002
Midterm Exam #2 – PRACTICE
Introduction to Computer Programming (Python)

Name: ______________________

NetID / Email: ______________________

Test Version: LINCOLN

General notes:

• When writing programs, comment your code as necessary.
• You will lose a small # of points for syntax errors so double-check your work to ensure that it is syntactically correct.
1. Trace the output of the following code samples

```python
for number in [100, 130, 10]:
    print("Number is: ", number)
```

```python
for number in range(100, -20, 20):
    print("Number is: ", number)
```

```python
for number in range(5, 10, 2):
    print("Number is: ", number)
```

```python
for x in range(2):
    for y in range(3):
        print (x,y)
```

```python
for x in range(2):
    y = 2
    while y > 0:
        print (x, y)
        y -= 1
```
2. Given the following String:

   phrase = "dora the explorer"

Identify the value of each of the following items:

phrase[0]

phrase[5]

phrase[5:8]

phrase[:8]

phrase[::3]

phrase[5::]

phrase[0:4].isalnum()

phrase[0:5].isalnum()

phrase[-1]

phrase.replace("dora", "python")

phrase.find("dora")

phrase.isalpha()
3. Trace the output of the following programs:

```python
def b(y):
    print("$\):", y)
    y += 1
    print("^\):", y)

y = 0
print("(:", y)
b(y)
print("):", y)

def m(i):
    i = i ** 2
    return i

def p(i):
    i = i // 2
    return i

print(m(2), p(2))

def one(a):
    print("a. one:\), a)
    a = two(a)
    print("b. one:\), a)

def two(a):
    print("a. two:\), a)
    a += 1
    print("b. two:\), a)
    return a

one(5)
```
4. An airplane has a total of 25 rows. Each row has 6 seats labeled A, B, C, D, E and F. Seats A and F are “window” seats, seats B and E are “middle” seats and seats C and D are “aisle” seats.

You are a programmer working for an airline that prices seats on its planes using the following algorithm:

- Seats in the “exit row” (rows 10 & 11) cost $200
- Seats in the first 5 rows of the airplane (rows 1-5) cost $500
- All other seats cost $150
- All middle seats (B and E) in any section are discounted by $50 due to the fact that they are awful.

Write a **FUNCTION** that accepts two arguments (the row # and a desired seat). Compute the cost of that seat using the pricing chart above and **RETURN** the result. Note that your function should be able to handle invalid seats (i.e. row 40 seat Q doesn’t exist) – in this case you should return an “invalid seat” string. Comment your function using IPO notation. Here’s a sample program that uses your function:

```
print ("20 A: ", seat_cost(20, 'A'))  # 20 A: 150
print ("25 B: ", seat_cost(25, 'B'))  # 25 B: 100
print ("10 A: ", seat_cost(10, 'A'))  # 10 A: 200
print ("11 B: ", seat_cost(11, 'B'))  # 11 B: 150
print ("1 A: ", seat_cost(1, 'A'))    # 1 A: 500
print ("2 B: ", seat_cost(2, 'B'))   # 2 B: 450
print ("25 Q: ", seat_cost(25, 'Q')) # 25 Q: invalid seat
print ("99 A: ", seat_cost(99, 'A')) # 99 A: invalid seat
print ("85 X: ", seat_cost(85, 'X')) # 85 X: invalid seat
```

Note: you are not writing a full program for this question - you will do that in the next question. Just write the function as specified.
5. Write a program that prompts the user to enter a row # and seat #. Next, check the cost of that seat using the function you wrote for the previous question. If the seat is valid you should add the cost of the seat to the user's cart and provide them with a running total of their bill. Here's a sample running of your program:

Enter a row number (1-25): 20
Enter a seat (A,B,C,D,E or F): A
This seat will cost 150
Your total is currently 150
Would you like to buy another seat? yes

Enter a row number (1-25): 25
Enter a seat (A,B,C,D,E or F): B
This seat will cost 150
Your total is currently 300

Would you like to buy another seat? yes
Enter a row number (1-25): 1
Enter a seat (A,B,C,D,E or F): Q
Invalid seat
Enter a row number (1-25): 1
Enter a seat (A,B,C,D,E or F): A
This seat will cost 500
Your total is currently 800
Would you like to buy another seat? no

The total cost for your flight will be: 800
6. A DNA strand consists of a sequence of smaller molecules called “nucleotides” – there are four main nucleotides represented by the following four characters:

- C = Cytosine
- T = Thymine
- A = Adenine
- G = Guanine

Biologists often need to analyze a sequence of DNA to determine what nucleotides are present in order to learn more about the genes that are represented in that strand of DNA.

Your task for this problem is to write a program that analyzes a strand of DNA. Here’s an example strand:

```
DNA_sequence = '_CTAG ttt gABBa!
```

Note that the sequence above contains valid nucleotides (C, c, T, t, A, a, G, g) as well as a number of invalid nucleotides (spaces, invalid characters, etc.) You will need to “clean up” the sequence so that only the “good” characters remain. For example, the sequence above would “clean up” to the following sequence:

```
good_DNA_sequence = 'CTAGTTTGAA'
```

Once you have done this you should print the following information:

1. The cleaned up DNA sequence (convert valid lowercase nucleotides to uppercase, remove invalid characters)
2. Total number of valid nucleotides found
3. Total number of invalid characters found

Here’s a sample running of the program:

```
Original DNA Sequence: _CTAG ttt gABBa!
Good DNA Sequence: CTAGTTTGAA
Valid nucleotides found: 10
Invalid characters found: 6
```
7. Write a program that lets a student in our class input their scores for 10 homework assignments. You can assume that all homework assignments are worth 20 points. Ensure that the numbers entered are greater than 0, but numbers greater than 20 are OK (we like extra credit!). You can assume that the user will enter an integer or a float. If they input a negative number (or zero) you should continually prompt them to enter a valid number until they do so. Then use their input to generate a summary report that includes the total points earned, total points possible and average score. Below is a sample running of this program. Don’t worry about formatting your numbers for this problem.

Score 1: 19
Score 2: 18
Score 3: -1

Sorry, -1 is not a valid score. Please try again.

Score 3: 17
Score 4: 19
Score 5: 20
Score 6: 10
Score 7: 15
Score 8: 14
Score 9: 18
Score 10: 19

Total points earned: 169.0
Total points possible: 200
Average homework score: 16.9 / 20.0
## Python Command Index

<table>
<thead>
<tr>
<th>Core Language Elements and Functions</th>
<th>Module Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>and chr def del elif else except float for format global if import in input int max min not open or ord print range return str str.lower str.upper try while</td>
<td>random.randint</td>
</tr>
</tbody>
</table>

### String Testing Methods
- isalpha()
- isdigit()
- islower()
- isupper()
- isspace()
- isalnum()
- find()

### String Modification Methods
- rstrip()
- lstrip()
- lower()
- upper()
- capitalize()
- title()
- swapcase()
- replace()
<table>
<thead>
<tr>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>Nul</td>
<td>32</td>
<td>20</td>
<td>Space</td>
<td>64</td>
<td>40</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>01</td>
<td>Start of heading</td>
<td>33</td>
<td>21</td>
<td>!</td>
<td>97</td>
<td>61</td>
<td>1a</td>
</tr>
<tr>
<td>2</td>
<td>02</td>
<td>Start of text</td>
<td>34</td>
<td>22</td>
<td>&quot;</td>
<td>98</td>
<td>62</td>
<td>2b</td>
</tr>
<tr>
<td>3</td>
<td>03</td>
<td>End of text</td>
<td>35</td>
<td>23</td>
<td>#</td>
<td>99</td>
<td>63</td>
<td>3c</td>
</tr>
<tr>
<td>4</td>
<td>04</td>
<td>End of transmit</td>
<td>36</td>
<td>24</td>
<td>$</td>
<td>100</td>
<td>64</td>
<td>4d</td>
</tr>
<tr>
<td>5</td>
<td>05</td>
<td>Enquiry</td>
<td>37</td>
<td>25</td>
<td>%</td>
<td>101</td>
<td>65</td>
<td>5e</td>
</tr>
<tr>
<td>6</td>
<td>06</td>
<td>Acknowledge</td>
<td>38</td>
<td>26</td>
<td>&amp;</td>
<td>102</td>
<td>66</td>
<td>6f</td>
</tr>
<tr>
<td>7</td>
<td>07</td>
<td>Audible bell</td>
<td>39</td>
<td>27</td>
<td>'</td>
<td>103</td>
<td>67</td>
<td>7g</td>
</tr>
<tr>
<td>8</td>
<td>08</td>
<td>Backspace</td>
<td>40</td>
<td>28</td>
<td>(</td>
<td>104</td>
<td>68</td>
<td>8h</td>
</tr>
<tr>
<td>9</td>
<td>09</td>
<td>Horizontal tab</td>
<td>41</td>
<td>29</td>
<td>)</td>
<td>105</td>
<td>69</td>
<td>9i</td>
</tr>
<tr>
<td>10</td>
<td>0A</td>
<td>Line feed</td>
<td>42</td>
<td>2A</td>
<td>*</td>
<td>106</td>
<td>6A</td>
<td>;</td>
</tr>
<tr>
<td>11</td>
<td>0B</td>
<td>Vertical tab</td>
<td>43</td>
<td>2B</td>
<td>+</td>
<td>107</td>
<td>6B</td>
<td>k</td>
</tr>
<tr>
<td>12</td>
<td>0C</td>
<td>Form feed</td>
<td>44</td>
<td>2C</td>
<td>,</td>
<td>108</td>
<td>6C</td>
<td>l</td>
</tr>
<tr>
<td>13</td>
<td>0D</td>
<td>Carriage return</td>
<td>45</td>
<td>2D</td>
<td>-</td>
<td>109</td>
<td>6D</td>
<td>m</td>
</tr>
<tr>
<td>14</td>
<td>0E</td>
<td>Shift out</td>
<td>46</td>
<td>2E</td>
<td>.</td>
<td>110</td>
<td>6E</td>
<td>n</td>
</tr>
<tr>
<td>15</td>
<td>0F</td>
<td>Shift in</td>
<td>47</td>
<td>2F</td>
<td>/</td>
<td>111</td>
<td>6F</td>
<td>o</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>Data link escape</td>
<td>48</td>
<td>30</td>
<td>0</td>
<td>112</td>
<td>70</td>
<td>p</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>Device control 1</td>
<td>49</td>
<td>31</td>
<td>1</td>
<td>113</td>
<td>71</td>
<td>q</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>Device control 2</td>
<td>50</td>
<td>32</td>
<td>2</td>
<td>114</td>
<td>72</td>
<td>r</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>Device control 3</td>
<td>51</td>
<td>33</td>
<td>3</td>
<td>115</td>
<td>73</td>
<td>s</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>Device control 4</td>
<td>52</td>
<td>34</td>
<td>4</td>
<td>116</td>
<td>74</td>
<td>t</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>Neg acknowledge</td>
<td>53</td>
<td>35</td>
<td>5</td>
<td>117</td>
<td>75</td>
<td>u</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>Synchronous ide</td>
<td>54</td>
<td>36</td>
<td>6</td>
<td>118</td>
<td>76</td>
<td>v</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
<td>End trans. block</td>
<td>55</td>
<td>37</td>
<td>7</td>
<td>119</td>
<td>77</td>
<td>w</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>Cancel</td>
<td>56</td>
<td>38</td>
<td>8</td>
<td>120</td>
<td>78</td>
<td>x</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>End of medium</td>
<td>57</td>
<td>39</td>
<td>9</td>
<td>121</td>
<td>79</td>
<td>y</td>
</tr>
<tr>
<td>26</td>
<td>1A</td>
<td>Substitution</td>
<td>58</td>
<td>3A</td>
<td>;</td>
<td>122</td>
<td>7A</td>
<td>z</td>
</tr>
<tr>
<td>27</td>
<td>1B</td>
<td>Escape</td>
<td>59</td>
<td>3B</td>
<td>&lt;</td>
<td>123</td>
<td>7B</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>File separator</td>
<td>60</td>
<td>3C</td>
<td>&gt;</td>
<td>124</td>
<td>7C</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>1D</td>
<td>Group separator</td>
<td>61</td>
<td>3D</td>
<td>=</td>
<td>125</td>
<td>7D</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>Record separator</td>
<td>62</td>
<td>3E</td>
<td>?</td>
<td>126</td>
<td>7E</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td>Unit separator</td>
<td>63</td>
<td>3F</td>
<td>#</td>
<td>127</td>
<td>7F</td>
<td></td>
</tr>
</tbody>
</table>