CSE 160 Spring 2023 - Midterm Exam

Instructions:
- You have **40 minutes** to complete this exam.
- The exam is **closed book**, including no calculators, computers, phones, watches or other electronics.
- You are allowed a single sheet of notes for yourself.
- We also provide a syntax reference sheet.
- Turn in **all sheets** of this exam, together and in the same order when you are finished.
- When time has been called, you must put down your pencil and stop writing.
  - **Points will be deducted if you are still writing after time has been called.**
- You may only use parts and features of Python that have been covered in class.
- All questions assume Python version 3.7, as we have been using all quarter.
- You may ask questions by raising your hand, and a TA will come over to you.

**Good luck!**

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td></td>
</tr>
<tr>
<td>Question 2</td>
<td></td>
</tr>
<tr>
<td>Question 3</td>
<td></td>
</tr>
<tr>
<td>Question 4</td>
<td></td>
</tr>
<tr>
<td>Question 5</td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
</tr>
</tbody>
</table>
1. [6 pts] For each of the below expressions, write what the expression evaluates to and the type of that value.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Evaluation</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>$11 / 2$</td>
<td>5.5</td>
<td>float</td>
</tr>
<tr>
<td>$2 + 11$</td>
<td>13</td>
<td>int</td>
</tr>
<tr>
<td>$11 % 2 == 0$ or not False</td>
<td>True</td>
<td>bool</td>
</tr>
<tr>
<td>not(len(‘hello’) &lt; 5 and 10.5 &gt; 2)</td>
<td>True</td>
<td>bool</td>
</tr>
<tr>
<td>‘hello’[1] + str(3 * 7.0)</td>
<td>“e21.0”</td>
<td>str</td>
</tr>
<tr>
<td>a = 10, b = ‘hello’; {&quot;a&quot;: 1, 1: &quot;a&quot;, &quot;b&quot;: b}[‘b’][2] + f[1]</td>
<td>“la”</td>
<td>str</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level</th>
<th>Category</th>
<th>Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>7(high)</td>
<td>exponent</td>
<td>**</td>
</tr>
<tr>
<td>6</td>
<td>multiplication</td>
<td>*,,//,%</td>
</tr>
<tr>
<td>5</td>
<td>addition</td>
<td>+,-</td>
</tr>
<tr>
<td>4</td>
<td>relational</td>
<td>==,!=,&lt;,&gt;,&gt;=,&gt;,&lt;</td>
</tr>
<tr>
<td>3</td>
<td>logical</td>
<td>not</td>
</tr>
<tr>
<td>2</td>
<td>logical</td>
<td>and</td>
</tr>
<tr>
<td>1(low)</td>
<td>logical</td>
<td>or</td>
</tr>
</tbody>
</table>
2. Write a function `tricky_function(arr)` that modifies a given list in the following way: at index `i`, if the value at index `i` is divisible by 3, then the function should multiply the value at index `i` by its index; otherwise leave the value as it is.

Examples:
```
tricky_function ([0, 1, 2, 3, 4, 5, 6, 7, 8]) = [0, 1, 2, 9, 4, 5, 36, 7, 8]
tricky_function ([2, 1, 21, 7, 4, 5, 4, 7, 18]) = [2, 1, 21, 21, 4, 5, 24, 7, 18]
tricky_function ([2, 3, 2]) = [2, 3, 2]
tricky_function ([1, 1, 1, 1]) = [1, 1, 1, 1]
```

Possible/Example Answer(s):
```
def tricky_function(numbers: list[int]) -> numbers: list[int]
    for i in range(len(numbers)):
        if i % 3 == 0:
            numbers[i] = numbers[i] * i
    return numbers
```
3. Write a function `join(list1, list2)` that, given two lists of characters `list1` and `list2`, returns the number of pairs in `list1` and `list2` whose values are equal.

Examples:

- `join ([a,b,c,e], [a,a,e,b,c,c,d]) = 6`
  
  *[[a,a], [a,a], [b,b], [c,c], [c,c], [e,e]]*

- `join ([a,a], [a,a]) = 4`
  
  *[[a,a], [a,a], [a,a], [a,a]]*

- `join ([a,b,c], [a,b,c]) = 3`
  
  *[[a,a], [a,a], [b,b], [c,c]]*

- `join ([x,y,z], [a,a,e,b,c,c,d]) = 0`
  
  *[]*

Possible/Example Answer(s):

```python
def join(list1: list[str], list2: list[str]) -> int:
count = 0
for i in range(len(list1)):
    for j in range(len(list2)):
        if list1[i] == list2[j]:
            count += 1
return count
```
4. After running this snippet of code, what will this function print?

```python
def foo(n1, n2, n3, b, s, d):
    if n1 % 2 == 0 or not b:
        print("A")
    elif str(n2 * n3) == 21.0:
        print("B")
    else:
        print("C")
    if d["b"] + s[1] == "LA":
        print("D")
    if d["a"] == n1 - n2:
        print("E")
    else:
        print("F")

a = 11;
b = "hello"
c = False
d = 10.5
e = 2
f = { "a": 1, 1: "a", "b": b }

foo(a, d, e, c, b, f)
foo(17, 21, 1, True, "hallo", {"a": 1, 1: "a", "b": "hello"})
foo(0, 3, 7.0, False, "hallo",{"a": -3, 1: "a", "b": "hallo"})
```

Example Answer(s):

A
F
C
F
A
E
5. Write a function `max_consecutive_sum` that, given a list of integers, returns the absolute value of the smallest sum between two consecutive numbers in the list. Return -1 if the length of the list is less than or equal to 1.

Examples:
`max_consecutive_sum ([36, 45, 7]) = 81`
`max_consecutive_sum ([3]) = -1`
`max_consecutive_sum ([2, 3, 2]) = 5`
`max_consecutive_sum ([1, 1, 1, 1]) = 2`

Possible/Example Answer(s):
```
def max_consecutive_sum(numbers: list[int]) -> int:
    max = 0
    for i in range(0, len(numbers) - 1):
        sum = numbers[i] + numbers[i + 1]
        if sum > max:
            max = sum
    return max
```