COMP110 QZ02 - Version A

Sara Huston

TOTAL POINTS

32 / 35

QUESTION 1

4 pts

1.1 **1 / 1**

✓ **+ 1 pts** *Correct: `d["Dracula"] += 3` or `d["Dracula"] = 10`*

+ 0 pts Incorrect

Partial Credit

+ 0.5 pts Almost correct, but used parentheses `()` or curly braces `{}` instead of hard brackets `[]`

+ 0.5 pts Incorrect, but used correct subscription notation of `d["Dracula"]`

+ 0.5 pts Correct, but called dictionary the wrong name (e.g. `dict`) instead of `d`

1.2 **1 / 1**

✓ **+ 1 pts** *Correct: `d["Nessie"] = 2`*

+ 0 pts Incorrect

Partial Credit

+ 0.5 pts Almost correct, but used parentheses `()` or curly braces `{}` instead of hard brackets `[]`

+ 0.5 pts Correct, but called dictionary the wrong name (e.g. `dict`) instead of `d`

+ 0.5 pts Incorrect, but used correct subscription notation `d["Nessie"]`

+ 0.5 pts Correct, but missing quotations around `Nessie`

1.3 **1 / 1**

✓ **+ 1 pts** *Correct: `len(d)` or `print(len(d))`*

+ 0 pts Incorrect

Partial Credit

+ 0.5 pts Almost correct, but used hard brackets `[]` or curly braces `{}` instead of parentheses `()` or missing parenthesis

+ 0.5 pts Correct, but called dictionary the wrong name (e.g. `dict`) instead of `d`

1.4 **0 / 1**

+ 1 pts Correct: `d.pop("Mothman")`

✓ **+ 0 pts** *Incorrect*

Partial Credit

+ 0.5 pts Almost correct, but used hard brackets `[]` or curly braces `{}` instead of parentheses `()`

+ 0.5 pts Incorrect, but called `d.pop()`

+ 0.5 pts Correct, but called dictionary the wrong name (e.g. `dict`) instead of `d`

QUESTION 2

4 pts

2.1 **1 / 1**

✓ **+ 1 pts** *`hayride,sweater,cider,leaves`*

+ 0 pts `0,1,2,3`

+ 0 pts `IndexError`

2.2 **0 / 1**

✓ **+ 0 pts** *`hayride,sweater,cider,leaves`*

+ 0 pts `0,1,2,3`

+ 1 pts `IndexError`

2.3 **1 / 1**

+ 0 pts `hayride,sweater,cider,leaves`

✓ **+ 1 pts** *`0,1,2,3`*

+ 0 pts `IndexError`

2.4 **0 / 1**

+ 1 pts `hayride,sweater,cider,leaves`

✓ **+ 0 pts** *`0,1,2,3`*

+ 0 pts `IndexError`

QUESTION 3

10 pts

3.1 Output **1 / 1**

✓ **+ 1 pts** *Correct: `[8,14,12]`*

+ 0 pts Incorrect

Partial Credit

+ 0.5 pts Correct values, wrong order `[8,12,14]`

+ 0.5 pts Almost correct, one wrong value (e.g. `[8,4,12]` or `[8,14,16]`)

+ 0.5 pts Has correct values, but extra incorrect values

3.2 Diagram **9 / 9**

Globals

✓ **+ 0.5 pts** *`main` in globals as `fn 1-5`*

✓ **+ 0.5 pts** *`f` in globals as `fn 7-11`*

✓ **+ 0.5 pts** *`g` in globals as `fn 13-19`*

`main` frame

✓ **+ 0.5 pts** *`RA` of 22*

✓ **+ 0.5 pts** *`y` points to a list on the heap with indexes `0,1,2` and values `5,8,7`*

✓ **+ 0.5 pts** *`z` points to a list on the heap with indexes `0,1,2` and initial values `4,7,6`*

✓ **+ 0.5 pts** *`z` points to a list on the heap with indexes `0,1,2` and final values `8,14,12`*

`g` frame

✓ **+ 0.5 pts** *Frame made and titled `g`*

✓ **+ 0.5 pts** *`RA` is 3*

✓ **+ 0.5 pts** *`inp_list` points to same list as the 'main' function's variable `y` on heap*

✓ **+ 0.5 pts** *`RV` and `x` point to same list on heap (NOT the same list as `y` and `inp_list`)*

✓ **+ 0.5 pts** *`idx` initialized as 0*

✓ **+ 0.5 pts** *`idx` final value of 3*

`f` frame

✓ **+ 0.5 pts** *Frame made and titled `f`*

✓ **+ 0.5 pts** *`RA` is 4*

✓ **+ 0.5 pts** *`inp_list` points to same location on heap as the 'main' function's variable `z`*

✓ **+ 0.5 pts** *`idx` initialized as 0*

✓ **+ 0.5 pts** *`idx` final value of 3*

+ 0 pts Incorrect or Blank

- 1 pts Extra, incorrect value on diagram

QUESTION 4

10 pts

4.1 Output **2 / 2**

✓ **+ 2 pts** *Correct:*

`4.0`

`True`

+ 0 pts Incorrect

Partial Credit (Pick one)

+ 1 pts Included `4.0`

+ 1 pts Included `True`

4.2 Diagram **8 / 8**

Globals

✓ **+ 0.5 pts** *`f` in globals as `fn 1-6`*

✓ **+ 0.5 pts** *`d` defined as dict on heap with keys `"ghosts", "bats", "candy"` and values `5.0,9.0,1.0`*

✓ **+ 0.5 pts** *`x` defined in globals as "ghosts" (WITH quotes)*

✓ **+ 0.5 pts** *`y` defined in globals as "bats" (WITH quotes)*

✓ **+ 0.5 pts** *`z` defined in globals as "candy" (WITH quotes)*

✓ **+ 0.5 pts** *`result1` defined in globals as `True`*

`f` frame

✓ **+ 1 pts** *`RA` is 12*

✓ **+ 1 pts** *`RV` is `True`*

✓ **+ 1 pts** *`my_dict` points to same dictionary on heap as global variable `d`*

✓ **+ 1 pts** *`x` is "candy" (WITH quotes)*

✓ **+ 0.5 pts** *`y` initialized as "ghosts" (WITH quotes)*

✓ **+ 0.5 pts** *`y` final value is "candy" (WITH quotes)*

+ 0 pts blank

QUESTION 5

5 **7 / 7**

✓ **+ 1 pts** *Correct input parameter `(a: list[int]) ` where the name `a` is of the student's choosing*

✓ **+ 0.5 pts** *Includes a doctstring.*

✓ **+ 0.5 pts** *Correct return type `list[int]`*

✓ **+ 1 pts** *Declares new empty `list[int]` with `[]` or `list()`, with correct typing.*

Loops through list correctly

✓ **+ 1 pts** *Uses correct syntax for for or while loop, including indexing*

✓ **+ 1 pts** *Correctly expresses conditional `if element % 2 == 0 and element < 7:`*

✓ **+ 1 pts** *Correctly `append`s elements to list with `l.append(elem)` where `l` is the list.*

✓ **+ 1 pts** *Returns a `list[int]` using a `return` statement*

+ 0 pts Incorrect or blank

gradescope

Quiz 02 - A

COMP 110: Introduction to Programming and Data Science Fall 2023

October 31, 2023 😀

Name: Sara Huston

9-digit PID: 730459812

Do not begin until given permission.

Honor Code: I have neither given nor received any unauthorized aid on this quiz.

Signed: 

Question 1: Dictionary Short Answer Suppose you have the following dictionary. Answer the related questions below.

```
1 d: dict[str, int] = {"Dracula": 7, "Mothman": 1, "Bigfoot": 6}
```

1.1. Write a line of code to increase the value associated with key "Dracula" by 3.

```
d["Dracula"] += 3
```

1.2. Write a line of code to insert the key "Nessie" with the value 2 into the dictionary

```
d["Nessie"] = 2
```

1.3. Write a line of code to print the number of key-value pairs in the dictionary d.

```
print(len(d))
```

1.4. Write a line of code to remove key "Mothman", whose value is 1, from the dictionary

```
pop. d["Mothman"]
```

Question 2: Multiple Choice Suppose you have the following list. For each code sample, choose the correct corresponding output. (Separate lines of output are represented by a comma.)

```
1 my_l: list[str] = ["hayride", "sweater", "cider", "leaves"]
```

2.1. 0,4

```
1 for x in range(0, len(my_l)):
2   print(my_l[x])
```

- ● hayride,sweater,cider,leaves
- ○ 0,1,2,3
- ○ IndexError

2.2.

```
1 for x in my_l:
2   print(my_l[x])
```

- ● hayride,sweater,cider,leaves
- ○ 0,1,2,3
- ○ IndexError

2.3.

```
1 for x in range(0, len(my_l)):
2   print(x)
```

- ○ hayride,sweater,cider,leaves
- ● 0,1,2,3
- ○ IndexError

2.4.

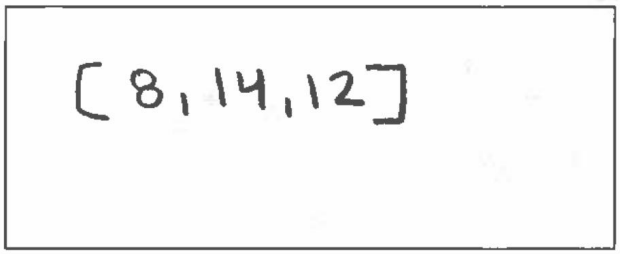
```
1 for x in my_l:
2   print(x)
```

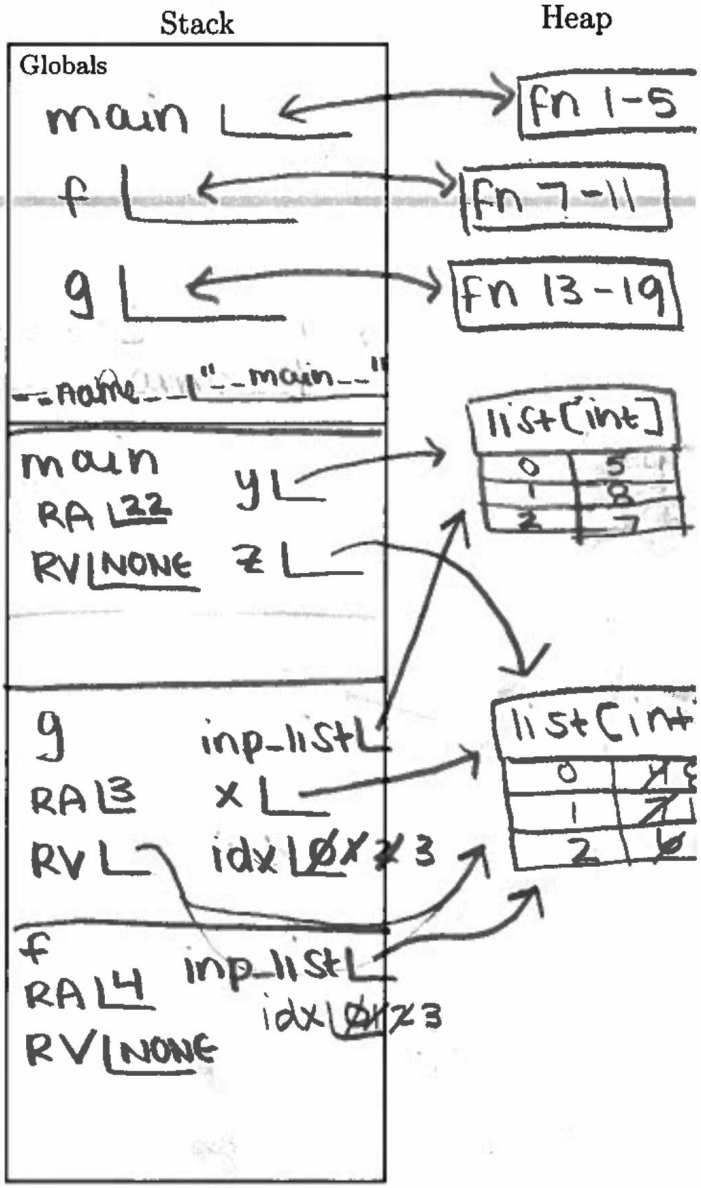
- ○ hayride,sweater,cider,leaves
- ● 0,1,2,3
- ○ IndexError

Question 3: Trace a memory diagram of the following code listing.

```
1  def main():
2      y: list[int] = [5, 8, 7]
3      z: list[int] = g(y)
4      f(z)
5      print(z)
6
7  def f(inp_list: list[int]) -> None:
8      idx: int = 0
9      while idx < len(inp_list):
10         inp_list[idx] *= 2
11         idx += 1
12
13 def g(inp_list: list[int]) -> list[
       int]:
14     x: list[int] = list()
15     idx: int = 0
16     while idx < len(inp_list):
17         x.append(inp_list[idx] - 1)
18         idx += 1
19     return x
20
21 if __name__ == "__main__":
22     main()
```

Output

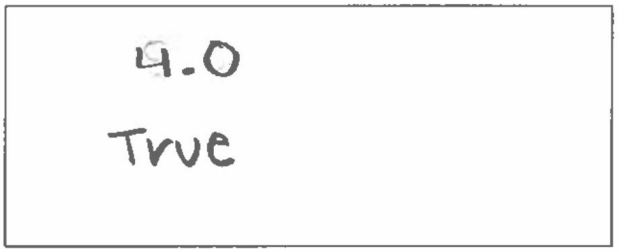


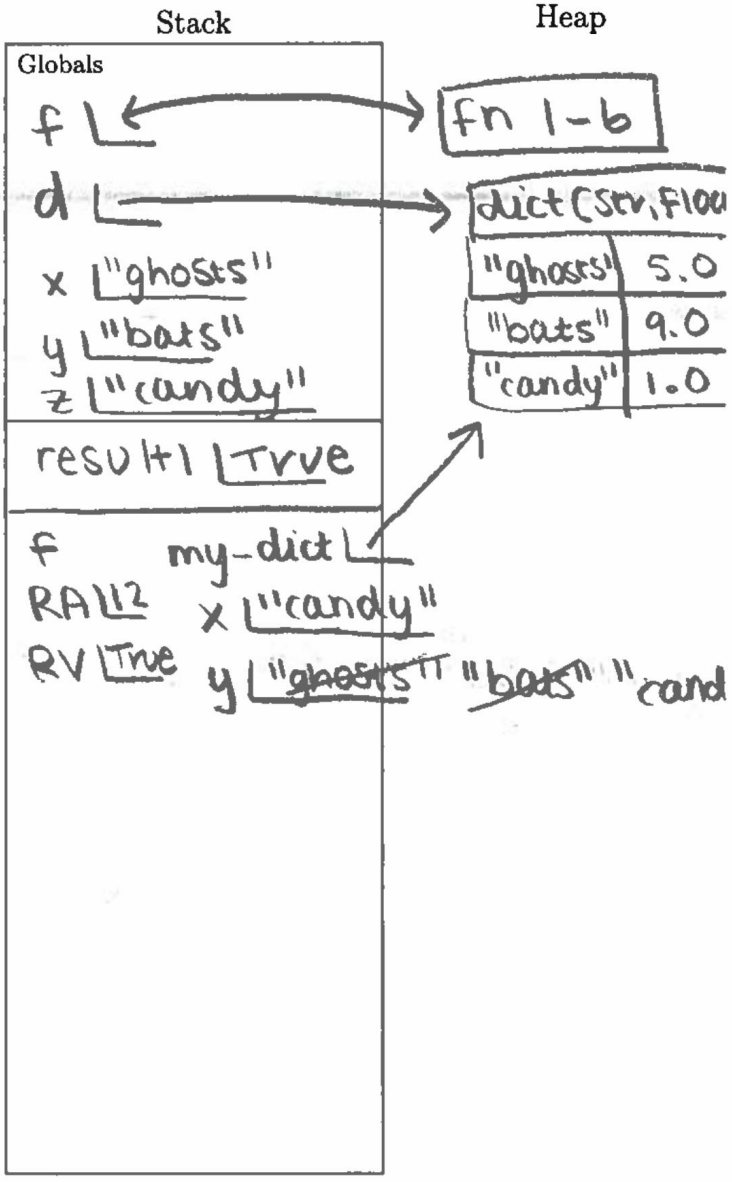


Question 4: Trace a memory diagram of the following code listing.

```
1  def f(my_dict: dict[str,float],
          x: str) -> bool:
2    for y in my_dict:
3      if y == x:
4        print(my_dict[y] + 3.0)
5        return True
6    return False
7
8  d: dict[str,float] = {"ghosts": 5.0,
       "bats": 9.0, "candy": 1.0}
9  x: str = "ghosts"
10 y: str = "bats"
11 z: str = "candy"
12 result1: bool = f(d,z)
13 print(result1)
```

Output





Question 5: Function Writing Write a function definition with the following expectations:

- The function name is shrink, takes a `list[int]` as input, and returns a `list[int]`.
- The function should return a `list[int]` of only the items from the input that are both **even** and < 7.
- The function should not modify the input list.
- The function should have the docstring: `Get values of list that are even and < 7`
- Explicitly type variables, parameters, and return types.
- The following REPL examples demonstrate expected functionality of your `shrink` function:

```
>>> shrink([5, 4, 8])
[4]
>>> shrink([4, 2])
[4, 2]
>>> shrink([8, 9])
[]
```

5.1. Write your function definition here:

```
def shrink(list_1: list[int]) -> list[int]:
    """Get values of list that are even and <7."""
    new_list: list[int] = list()
    i: int = 0
    while i < len(list_1):
        if list_1[i] % 2 == 0 and list_1[i] < 7:
            new_list.append(list_[i])
        i += 1
    return new_list
```

This page intentionally left blank. Do not remove from exam packet.