Python (Basic) – Revision

Cyrille Jegourel, Matthieu de Mari – Singapore University of Technology and Design

Summary

- Some typical mistakes
- What to revise?
- Conclusion

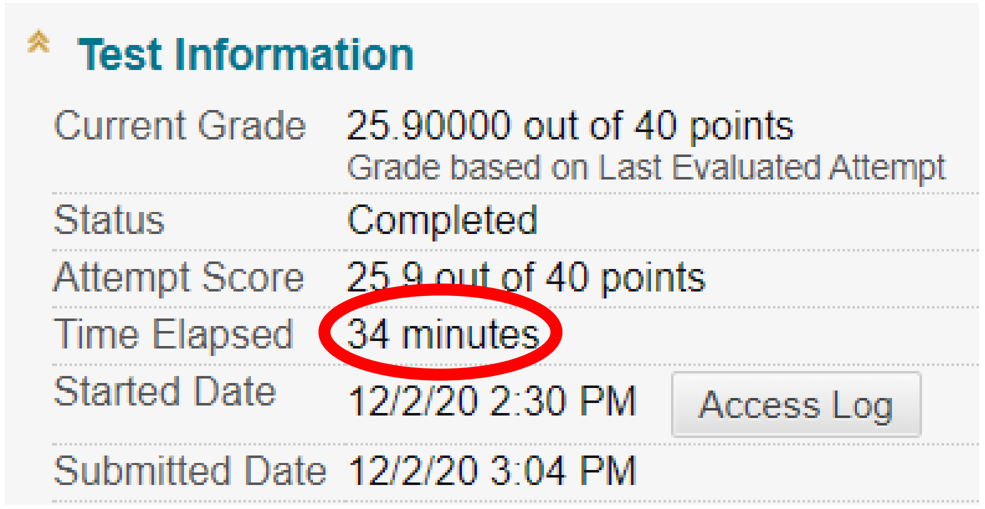
Time management

If we give you 90 minutes to complete the exam,

USE THEM!

This is not a race!

However, don't play with the time limit!
No matter what, **submit 2 minutes before the deadline.**



Test Information

Current Grade	25.90000 out of 40 points Grade based on Last Evaluated Attempt
Status	Completed
Attempt Score	25.9 out of 40 points
Time Elapsed	34 minutes
Started Date	12/2/20 2:30 PM Access Log
Submitted Date	12/2/20 3:04 PM

Modulo operator

- 7 % **4** = (1 * **4** **+ 3**) % 4 = **3**
- 11 % **4** = (2 * **4** **+ 3**) % 4 = **3**
- 4 % **7** = (0 * **7** **+ 4**) % 7 = **4**

- -7 % **4** = (-2 * **4** **+ 1**) % 4 = **1**

- To remember:
  - After a % b = r, we always have 0 <= r < b.
  - So, if 0 <= a < b, we always have r = a.
  - The green values (1, 2, 0, -2) can be found by integer division:
    - 7 // **4** = (1 * **4** **+ 3**) // 4 = **1**
    - 11 // **4** = (2 * **4** **+ 3**) // 4 = **2**
    - 4 // **7** = (0 * **7** **+ 4**) // 7 = **0**
    - -7 // **4** = (-2 * **4** **+ 1**) // 4 = **-2**
    - For positive numbers, integer division is equivalent to a division where we get rid of the decimal part.
    - For negative numbers, integer division is equivalent to a division where we get rid of the decimal part minus 1.

Confusion between elements and indexes

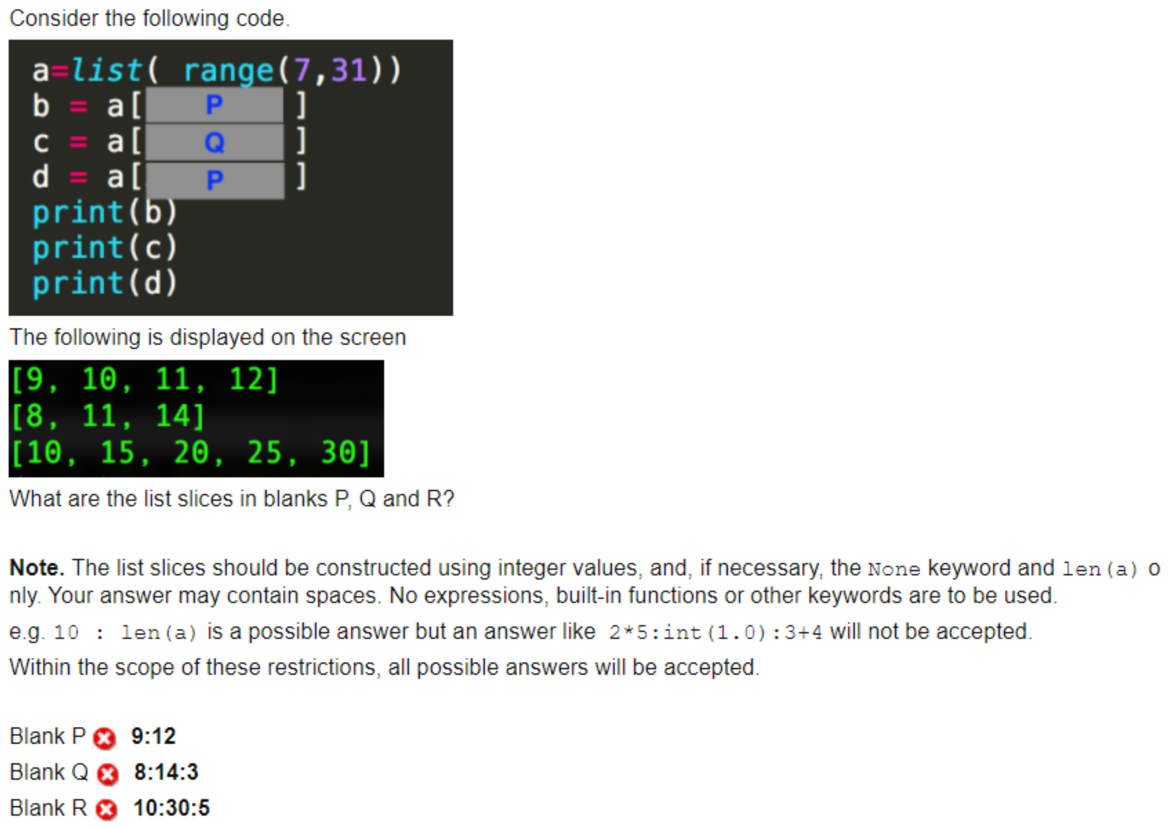
- Tips:
- Given a[**start** : **end** : **step**],
  1. Determine **step** (usually easy)

     a[**start** : **end** : **1**]
  2. Determine **start**, the index of the 1st element displayed on screen. E.g., 9 is the 3rd element in list a, so, its index is 2.

     a[**2** : **end** : **1**]
  3. Determine **end**, the index of the last element displayed on screen **and add 1** because a[end] is excluded. E.g., 12 is the 6th element in list a, so, its index is 5. **Add 1**.

     a[**2** : **6** : **1**]
  4. Adjust if necessary.

     a[**2** : **6**]



Consider the following code.

```
a=list( range(7,31))
b = a[   P   ]
c = a[   Q   ]
d = a[   P   ]
print(b)
print(c)
print(d)
```

The following is displayed on the screen

```
[9, 10, 11, 12]
[8, 11, 14]
[10, 15, 20, 25, 30]
```

What are the list slices in blanks P, Q and R?

Note. The list slices should be constructed using integer values, and, if necessary, the `None` keyword and `len(a)` only. Your answer may contain spaces. No expressions, built-in functions or other keywords are to be used.

e.g. `10 : len(a)` is a possible answer but an answer like `2*5:int(1.0):3+4` will not be accepted.

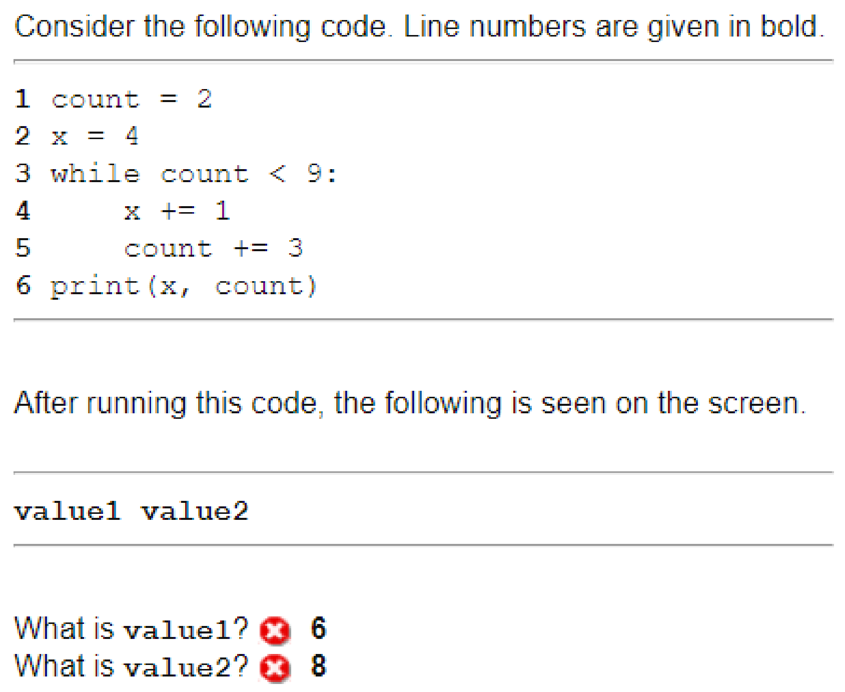
Within the scope of these restrictions, all possible answers will be accepted.

Blank P ✗ **9:12**
Blank Q ✗ **8:14:3**
Blank R ✗ **10:30:5**

Value of the counter after a while loop

- The last iteration is often forgotten.

- Tip:
    - Check the value of the counter after exiting the while loop.
    - It should violate the condition of the while loop (unless break or return…)



Consider the following code. Line numbers are given in bold.

```
1 count = 2
2 x = 4
3 while count < 9:
4     x += 1
5     count += 3
6 print(x, count)
```

After running this code, the following is seen on the screen.

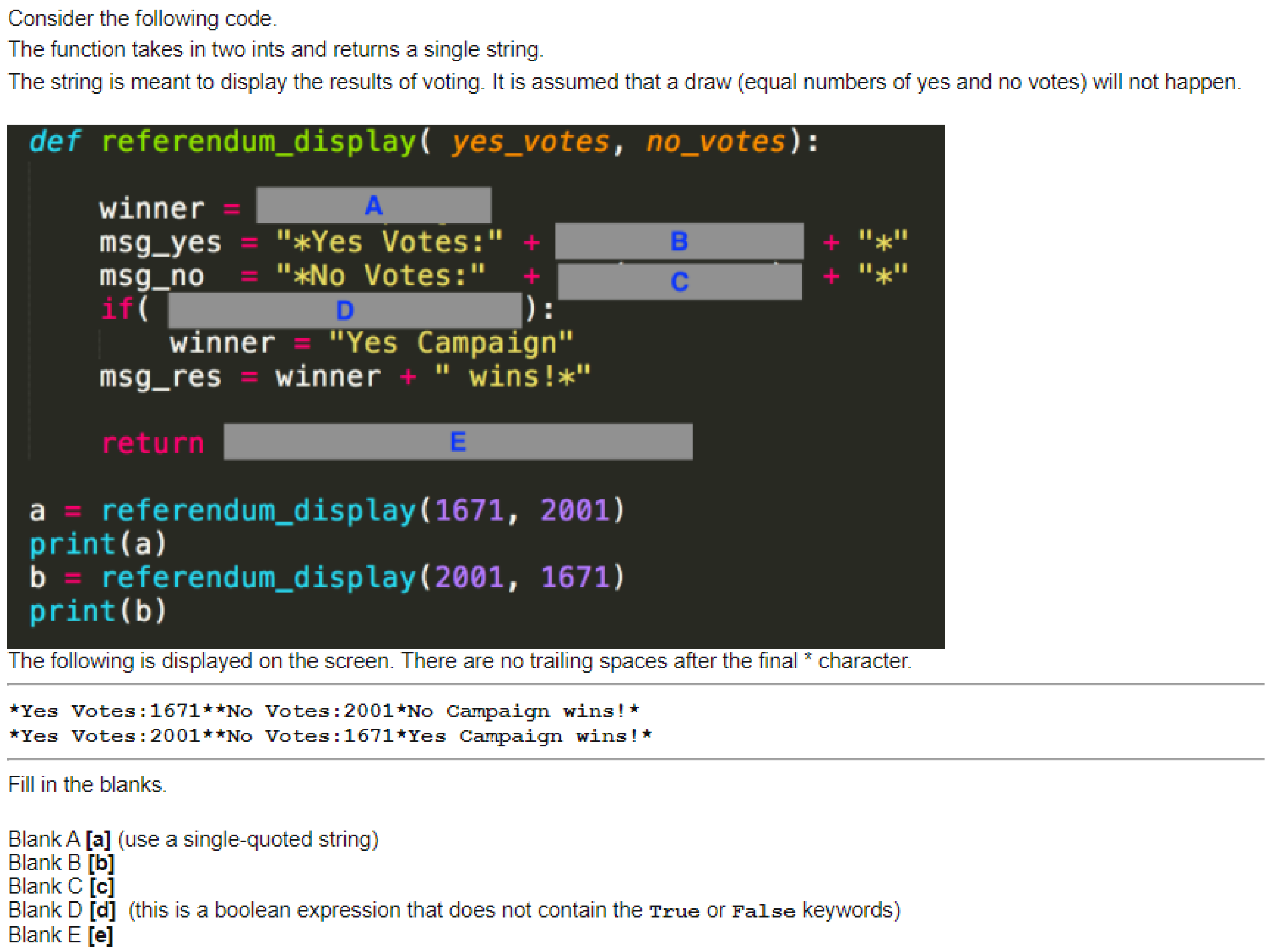
```
value1 value2
```

What is `value1`? ✗ **6**
What is `value2`? ✗ **8**

Filling the blanks…

Anyone who failed this exercise should do it again.
There is a lot of marks on this type of questions. So, here a few tips:

- Take your time
- Read carefully the question
- What are the inputs?
- What are the outputs?
- What is this function supposed to do?
- Look at what is displayed.



Consider the following code.
The function takes in two ints and returns a single string.
The string is meant to display the results of voting. It is assumed that a draw (equal numbers of yes and no votes) will not happen.

```
def referendum_display( yes_votes, no_votes):

    winner =       A
    msg_yes = "*Yes Votes:" +       B       + "*"
    msg_no  = "*No Votes:"  +       C       + "*"
    if(          D          ):
        winner = "Yes Campaign"
    msg_res = winner + " wins!*"

    return           E

a = referendum_display(1671, 2001)
print(a)
b = referendum_display(2001, 1671)
print(b)
```

The following is displayed on the screen. There are no trailing spaces after the final * character.

```
*Yes Votes:1671**No Votes:2001*No Campaign wins!*
*Yes Votes:2001**No Votes:1671*Yes Campaign wins!*
```

Fill in the blanks.

Blank A **[a]** (use a single-quoted string)
Blank B **[b]**
Blank C **[c]**
Blank D **[d]** (this is a boolean expression that does not contain the `True` or `False` keywords)
Blank E **[e]**

What to revise?

- You tell me

What to revise?

- Strings:
    - **Slicing**, Strip(), split(), replace(), format(), concatenation
- Lists and Nested lists
    - Matrices, memory management, alias, shallow and deep copy problems
- Nested loops
    - Nested loop over a matrix, value of some variables during/after a nested loop.
- Dictionaries
    - Keys, values, items, etc.
- Object oriented
- Turtle
- File handling
- Lab session:
    - Database, Robots, Sensors
- Others?

Revision

- Have a look on the Jupyter notebook.

- Regarding the lab session:
  - What do you think you are expected to know?

Memory management

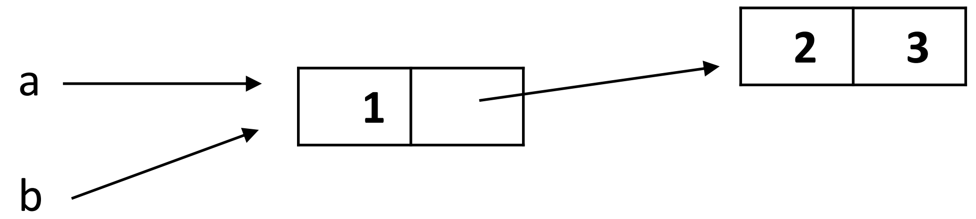
- Which of the following situations on the right describe respectively an alias, a shallow copy, a deep copy in the memory?

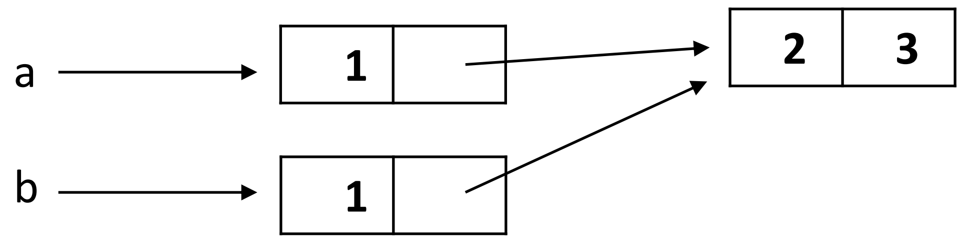
- Which of the following code results in the following situations in the memory?

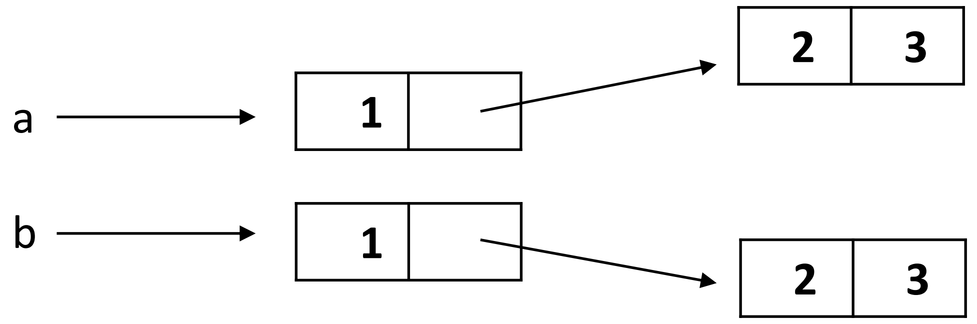
```python
a = [1, [2, 3]]
b = a
```

```python
a = [1, [2, 3]]
b = a[:]
```

```python
import copy
a = [1, [2, 3]]
b = copy.deepcopy(a)
```







Conclusion

- Good luck for the final exam ☺