

MergeSort PRQ

Score: _____

1. What is **concatenation** (????????)?

- ☐ A An agricultural process
- ☐ B Growing cats
- ☐ C Finding square root of a number
- ☐ D Combining two lists into one
?????? ?????????????? ??????? ?????????

2. If **al** = [1, 2, 3] and **bl** = [4, 5, 6], what is the code to generate a **new list** that contains [1, 2, 3, 4, 5, 6]?

- ☐ A al + bl
- ☐ B bl + al
- ☐ C al.extend(bl)
- ☐ D al.append(bl)

3. If **al** = [4, 5, 6] and **bl** = [1, 2, 3], what is the code to generate **a new list** that contains [1, 2, 3, 4, 5, 6]?

- ☐ A al + bl
- ☐ B bl + al
- ☐ C al.extend(bl)
- ☐ D al.append(bl)

4. If **al** = [1, 2, 3] and **bl** = [4, 5, 6], what is the code to modify **al** to contain [1, 2, 3, 4, 5, 6]?

- ☐ A al + bl
- ☐ B bl + al
- ☐ C al.extend(bl)
- ☐ D al.append(bl)

5. If **al** = [4, 5, 6] and **bl** = [1, 2, 3], what is the code to modify **bl** to contain [1, 2, 3, 4, 5, 6]?

- ☐ A al + bl
- ☐ B bl + al
- ☐ C bl.extend(al)
- ☐ D al.append(bl)

6. If `al = [4, 5, 6]` and `bl = [1, 2, 3]` and `cl = [0]`, what is the code to *merge* `al` and `bl` (in that order) into `cl`?

- ☐ A `cl = cl + al + bl`
- ☐ B `cl += al + bl`
- ☐ C `cl.extend(al)`
`cl.extend(bl)`

7. The default parameter value for the list method `.pop()` is `-1`.

- ☐ A True
- ☐ B False

8. If `al = [1, 2, 3]` and `cl = []`,
The code
`smallest = al.pop(0)`
`cl.append(smallest)`
`smallest = al.pop(0)`
`cl.append(smallest)`

will result in

- ☐ A `al = [3]` and `cl = [1, 2]`
- ☐ B `al = [3]` and `cl = [2, 2]`
- ☐ C `al = [1, 2, 3]` and `cl[1, 2]`
- ☐ D `al = []` and `cl = [1, 2, 3]`

9. If `al = [1, 2, 3]` and `cl = []`, to make
`cl = [1, 2]` and `al = [3]`, what is the right code?

- ☐ A `cl = al.pop(0)`
`cl = al.pop()`
- ☐ B `cl = al.pop(0)`
`cl = al.pop(1)`
- ☐ C `cl += [al.pop(0)]`
`cl += [al.pop(0)]`
- ☐ D `cl += [al.pop(1)]`
`cl += [al.pop(2)]`

10. Write a program to demonstrate ternary operator in Python.
Use the ternary operator to assign the minimum of variable `a` and variable `b` to variable `c`.

11. list **al** = [1, 2] and **bl** = [3, 4].

What is the code to print the first elements of both al and bl?

- ☐ A print(1, 3)
- ☐ B print(al[1], bl[1])
- ☐ C print(al, bl)
- ☐ D print(al[0], bl[0])

12. The integer **100** has to be added as the last element of list **al**. Which code will is correct?

- ☐ A al.append(100)
- ☐ B al.insert(-1, 100)
- ☐ C al_len = len(al)
al.insert(al_len-1, 100)
- ☐ D al.extend([100])

13. If **al** = [1, 3, 5] and **bl** = [2, 4, 6], what is the code to extract **al** and **bl** into **cl** so that **cl** = [1, 2, 3, 4, 5, 6]?

- ☐ A cl = cl + al + bl
- ☐ B while al and bl:
cl = al.pop(0)
cl = bl.pop(0)
- ☐ C #---- start ----
while len(al) != 0 and len(bl) != 0:
cl.insert(-1, al.pop(0))
cl.insert(-1, bl.pop(0))
#---- end ----
- ☐ D #---- start ----
while al and bl:
cl.append(al.pop(0))
cl.append(bl.pop(0))
#---- end ----

14. Extract the minimum of the first elements that occur in the list **al** and **bl** and put it at the tail of the list **cl**

- ☐ A `cl = (al if al[0] < bl[0] else bl).pop(0)`
- ☐ B `cl = (al if al[0] < bl[0] else bl).pop(1)`
- ☐ C

```
# ----- start -----
if al[0] < bl[0]:
    cl.append( al[0] )
    al.pop(0)
else:
    cl.append( bl[0] )
    bl.pop(0)
# ----- end -----
```
- ☐ D

```
# ----- start -----
candidate = (al if al[0] < bl[0] else bl).pop(0)
c.append(candidate)
# ----- end -----
```
- ☐ E All of the above

15. Write code that creates a list **cl** by merging numbers in sorted list **al** and sorted list **bl** in ascending order. Print the newly formed list **cl**.

Clue: Combine answers from previous two questions.

16. Write a **merge** function which accepts two sorted lists A and B as parameters. It should return a list that is a merge of the elements of A and B so that they are in ascending order.

17. Modify the code provided in #16 (above) to avoid use of **pop(0)** which is expensive (i.e. takes too much time). Instead refactor the code to use **pop()**, which is relatively less expensive.

18. A properly tested and working merge function is available.
The output of `merge([1,3, 9, 7], [2, 8, 6, 4])` will return

- ☐ A [1, 2, 3, 4, 6, 7, 8, 9]
- ☐ B [1, 3, 9, 7, 2, 8, 6, 4]
- ☐ C [1, 2, 3, 8, 6, 4, 9, 7]
- ☐ D **None** of the above