

Name \_\_\_\_\_ Login name \_\_\_\_\_

TA \_\_\_\_\_

## General Instructions

- Only write your name on the *first* page of this document, in the blank provided.
- Turn off any noise making device, especially **CELL PHONES**. You may lose up to one letter grade if your device disturbs the peace of the exam
- You have 50 minutes. **Pace yourself**, and pay attention to the point values.
- You may leave the room at any time, but you must give your exam to the professor, and you may not return.
- **Do not add features** that are not required by the problem. For example, if the instructions don't say anything about user input, then your program should not take user input (similarly for printing). If you aren't sure, raise your hand and wait to be acknowledged by the professor.
- Do problems you are confident about first. If you finish the problems you know, write what you do know about other problems to gain partial credit; but erroneous information may detract from that credit or irritate the grader, so don't make stuff up.
- Read *all* the directions *carefully* on each problem.
- Often writing a fast, rough version of a program in English or pseudocode will make your coding faster and more accurate. It also enables me to give partial credit in some circumstances.
- You may assume that input will not produce errors for the code described, unless the questions say otherwise. What is your favorite cheese?
- Do not do unnecessary testing. For example, testing for both  $x < 0$  and  $x \geq 0$  instead of using one test and then `else` would be considered unnecessary testing.
- Have fun!

## Short Answer

1. (3 pts) Write an expression using the range function to **make a variable** hold the list of integers from 2 up to and including 7
2. (3 pts) Write one expression with slicing that uses the variable you made in problem 1 to replace the sublist containing only 3,4,5 with the sublist containing 1.
3. (3 pts) Write one expression with slicing that uses the variable you made in problem 1 to insert a 9 just before the 7.
4. (3 pts) Represent animals and their number of legs as follows: Write code to make a dictionary that associates the elements of [ 'dog', 'cat', 'emu' ] with the elements of [ 4, 4, 2 ].<sup>1</sup>
5. (3 pts) Write code to add (to your existing dictionary from problem 4) that a stoat has four legs.
6. (12 pts) You are **given** a function get\_time() which tells you the current system time on your computer in milliseconds. You are also given a working sort() function, which you want to time. Write lines of code (**not a function**) to show how to time the sort function running once on a list of 5000 unordered elements. Print the elapsed time at the end. Use any helper functions you need without importing.

---

---

---

---

---

---

---

---

<sup>1</sup>Long or short solution is fine.

7. (6 pts) The data record for an echidna requires about 2,000 bytes of memory. As discussed in class, approximately how many echidna records can we store in a 32 megabyte drive? Show your work to receive credit.
8. (6 pts) Suppose you are given a list of size 4000. Based on the complexity of the algorithm, what would be an estimate of the number of comparisons performed by insertion sort on this list, as discussed in class? Show your work using powers of two to receive credit.
9. (6 pts) Suppose you are given a list of size 32,000,000. Based on the complexity of the algorithm, what would be an estimate of the number of comparisons performed by mergesort on this list, as discussed in class? Show your work, using powers of two where possible, to receive credit.
10. (8 pts) In the blanks provided, show the value of each expression:

$$9//2 \text{ \_\_\_\_\_\_}$$

$$8\%3 \text{ \_\_\_\_\_\_}$$

$$13//3 \text{ \_\_\_\_\_\_}$$

$$11\%2 \text{ \_\_\_\_\_\_}$$

---

Do not put answers below this line.

11. (15 pts) Fill in the five blanks with numbers only<sup>2</sup>. This function may not be exactly the same as one we wrote in class. Watch variable names. Hint: write all the code first, *then* look for it.

```
def binarySearch(key, alist, start, end):
```

```
    _____  
    return None
```

```
    _____  
  
    if key < alist[mid]:
```

```
        _____  
        elif key == alist[mid]:
```

```
    else:
```

```
        _____
```

---

<sup>2</sup>Please write clearly. Unclear, hard to read, or partially erased or crossed out answers will be wrong.

```

1. print 'found it!'
2. print 'key not found'
3. return alist[mid]
4. return mid
5. index = mid
6. index = alist[mid]
7. if start == end:
8. if start < end:
9. if start > end:
10. if alist != []
11. print 'base case'
12. print locals()
13. index = binarySearch(alist, key, start, mid-1)
14. index = binarySearch(key, alist, start, mid-1)
15. index = binarySearch(alist, key, start+1, mid)
16. return binarySearch(alist, key, start, mid-1)
17. return binarySearch(key, alist, end, start)
18. return binarySearch(alist[:mid], key, start, mid)
19. return binarySearch(key, alist, start, mid-1)
20. index = binarySearch(alist, key, mid+1, end)
21. index = binarySearch(key, alist, mid-1, end)
22. index = binarySearch(alist, key, mid, end-1)
23. return binarySearch(key, alist, mid+1, end)
24. return binarySearch(alist, key, mid+1, end)
25. return binarySearch(key, alist, start, end)
26. return binarySearch(alist[mid:], key, mid, end)
27. mid = (start + end) // 2
28. mid = start + end // 2
29. mid = len(alist) // 2
30. index = len(alist) // 2
31. index = (start + end) // 2

```

12. (15 pts) Fill in the blanks to complete the function we wrote in class.

```
def mergesort(alist):
```

```
    if _____
```

```
        return alist
```

```
    else:
```

```
        mid = _____
```

```
        _____
```

```
        _____
```

```
    return _____
```

13. (17 pts)

Write a function that takes any 2D nested list as a parameter. The function goes through the entire nested list and replaces any instance of -999 with a 0<sup>3</sup>. For example, given `[[2, -999], [-999, 3, 4], [-999]]` the result would be `[[2, 0], [0, 3, 4], [0]]`. Include your purpose and contract.

---

<sup>3</sup>that is, it modifies the original list.