

Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.

Name: _____

Parsing Strings

1. Given the string `x = 'artificial intelligence for games'`, indicate a slicing operation that yields the following strings:

1 (a) `'o'`

1 (b) `'artificial'`

1 (c) `'icial'`

1 (d) `'atfca nelgnefrgms'`

1 (e) `'semag rof ecnegilletni laicifitra'`

Lists, Tuples, and Dictionaries

Use the following table for the questions in this section:

Unit	X	Y
Carrier	5	1
Observer	6	14
Probe	2	7
Corsair	3	12

- 5 2. Create a *list* called `units` containing all of the units in the above table.

- 5 3. Use the `sort` method to sort the above list. See Chapter 1 for details.

- 5 4. Create a *list of tuples* for each of the *X* and *Y* positions of the units.

- 5 5. Create a *dictionary* called `dct` using the table, so that `dct['Carrier']` is `(5, 1)`.



- 5 6. Add a `Scout` to the above dictionary having position `(7,125)`.

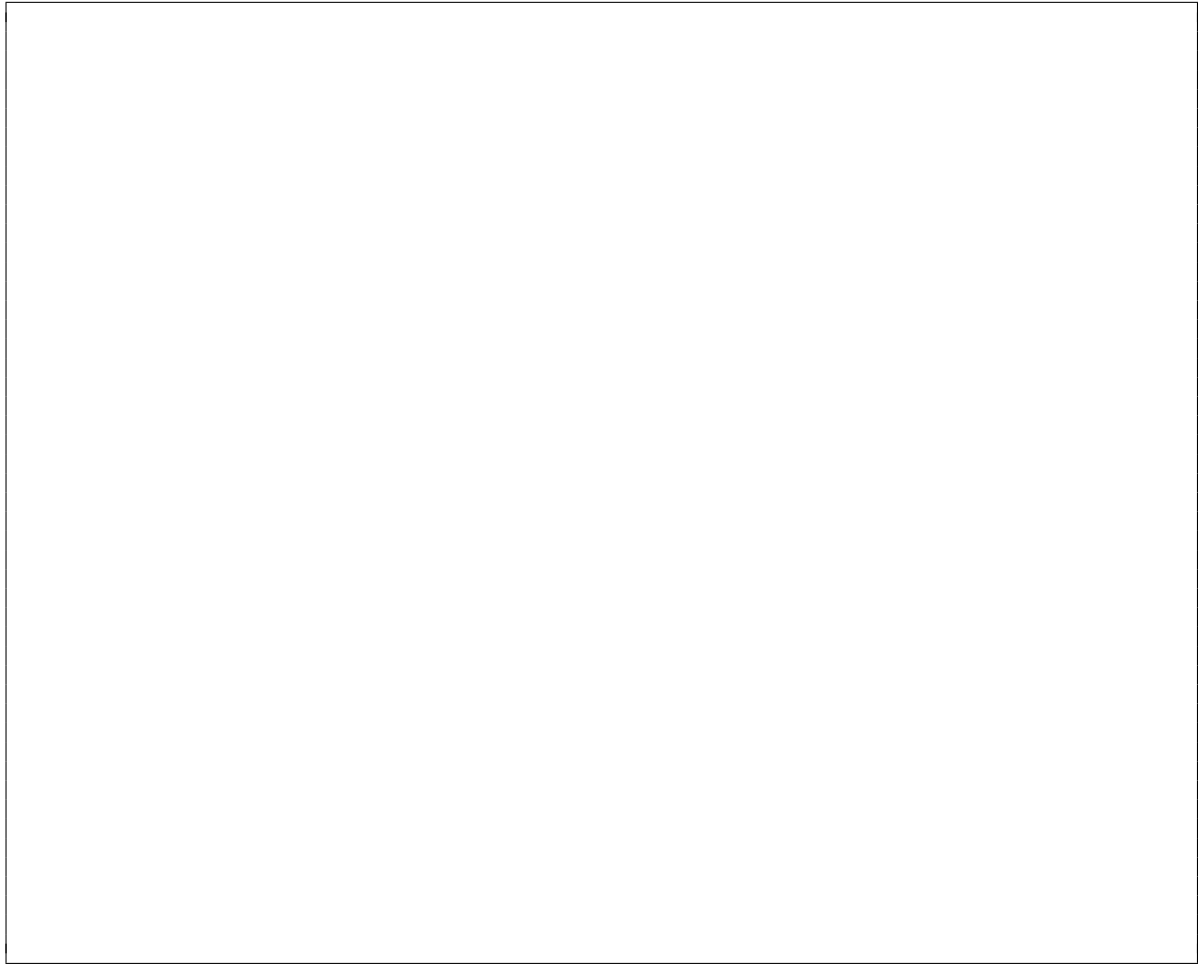


These three data structures are central to Python. Be sure that you can use these effectively!

Loops

- 10 7. Assume that you have a list containing the health of every unit in your game. Such a list might look like this: `health = [45, 3, 12, 80, 50, 100, 5, 5, 8, 10]`. A unit is considered low on health if they have less than (`<`) 10 health.

Using algorithms, generate a new list called `low_health` that contains these items. In the example above, `low_health` should contain four elements, `[3, 5, 5, 8]`.



Variables

- 10 8. Draw a memory diagram that represents the following code:

```
x = ['a', 'b', 'c', 'd']
```

```
y = ['a', 'b', 'c', 'd']
```

```
z = x
```

```
q = y
```

```
z[3] = 'c'
```

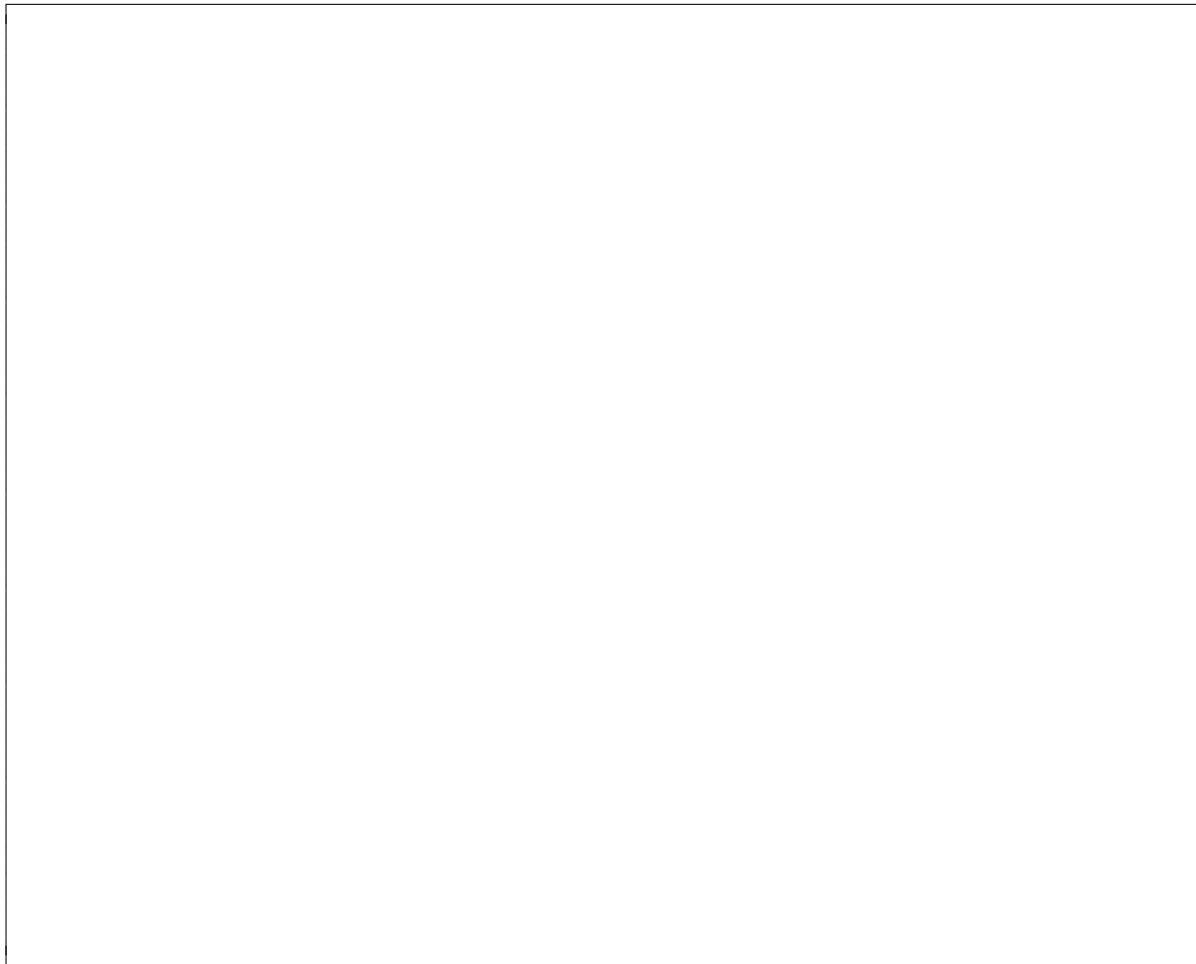
```
q[1] = 'z'
```

```
print x
```

```
print y
```

```
print z
```

```
print q
```



Conditionals

9. In a game such as Street Fighter, random number generators are often used to decide what action to take. Often, this is done with what is known as a **probability distribution function**. A probability distribution function can be encoded using a series of `if` and `elif` statements. Assume that you have a random number generator (see Chapter 2) which generates integers in the closed interval $[1, 100]$:

```
import random
r = random.randint(1, 100)
```

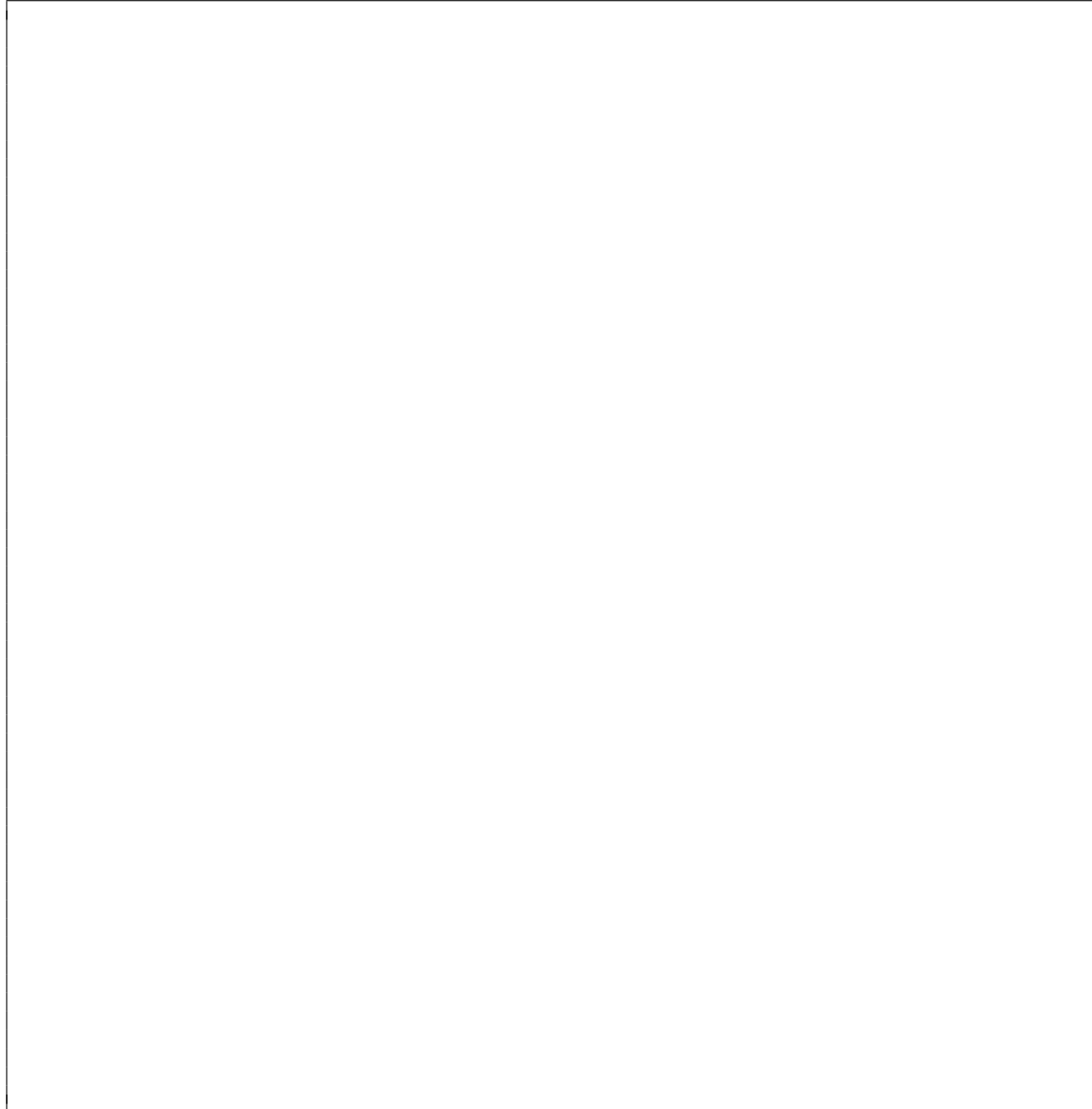
Depending on the random number generated, the opponent will perform the following:

Move	Low (\geq)	High (\leq)
Punch	1	40
Kick	41	60
Defend	61	90
Special Move	91	100

So, if $r \geq 1 \wedge r \leq 40$ (\wedge is the mathematical symbol for **and**), the opponent will print **Punch**.

10

- (a) Write a small Python script that has an opponent performing 10 different moves. Each move should be randomly generated, using the table given above.



5

- (b) Using the table above, on average, which move is performed by this opponent most frequently? Least frequently?

--

Question	Points	Score
1	5	
2	5	
3	5	
4	5	
5	5	
6	5	
7	10	
8	10	
9	15	
Total:	65	