

1 Hashing

(a) Imagine we have the following class:

```
1 public class Course {
2     public final int CCN;
3     public final String instructor;
4     public Student[] students;
5     public int audited; //when the course was last audited
6     public Course(int CCN, Student[] initial) {
7         this.CCN = CCN;
8         this.students = initial;
9         this.instructor = "Sohum";
10    }
11    //implementation
12    public void audit() {
13        this.audited = System.currentTimeMillis();
14        //implementation
15    }
16    public void addStudent(Student s) {
17        //implementation
18    }
19 }
```

Which of the following hashing functions for the Course class are **valid**?

A)

```
1 @Override
2 public int hashCode() {
3     return CCN;
4 }
```

B)

```
1 @Override
2 public int hashCode() {
3     return this.students.length;
4 }
```

C)

```
1 @Override
2 public int hashCode() {
3     return this.audited;
4 }
```

2 Hashing

D)

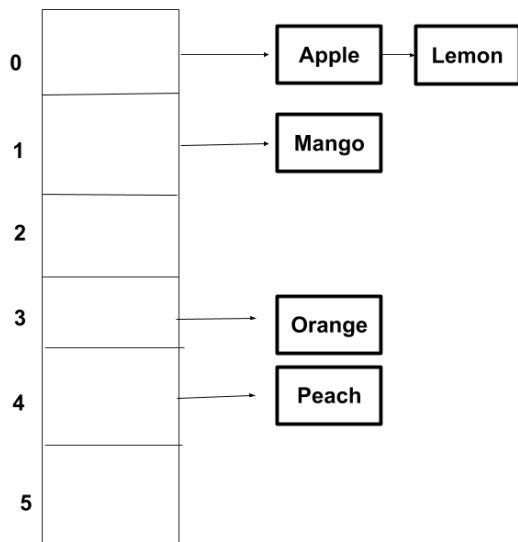
```
1 @Override
2 public int hashCode() {
3     return 5;
4 }
```

E)

```
1 @Override
2 public int hashCode() {
3     return getNumericValue(this.instructor.charAt(0));
4 }
```

- A
- B
- C
- D
- E

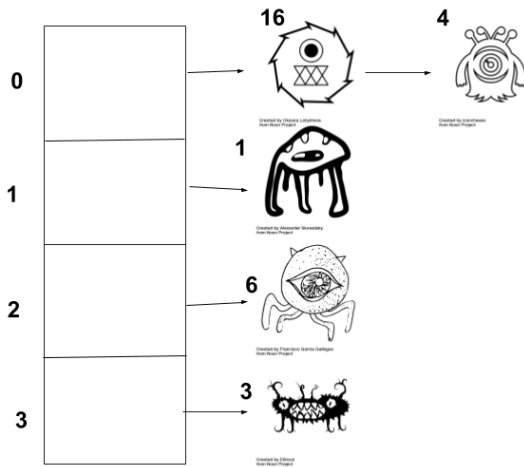
(b) We have the below external chaining HashSet.



If the load factor is 1.25, how many more insertions can we make before we will resize? **Do not** include the insertion that will begin with the resize.

- 1
- 2
- 3
- 4
- 5
- 6

(c) Suppose we have the following MonsterHashTable.



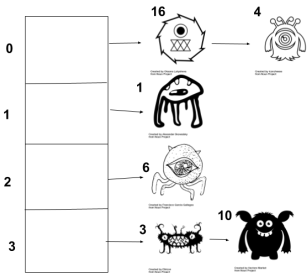
The number to the upper left of the monster is their hashCode.

Part One:

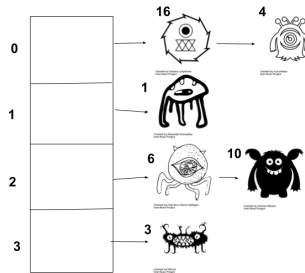
Suppose we want to insert the element:



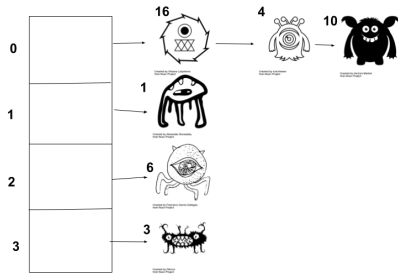
Which of the following would correctly mirror the state after inserting the above element?



A)



B)



C)

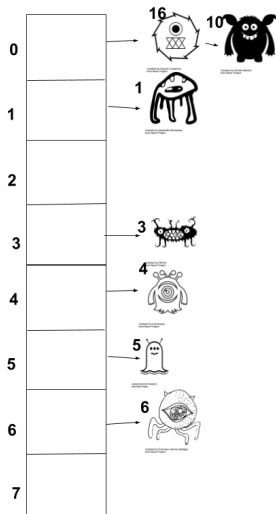
- A
- B
- C

Part Two:

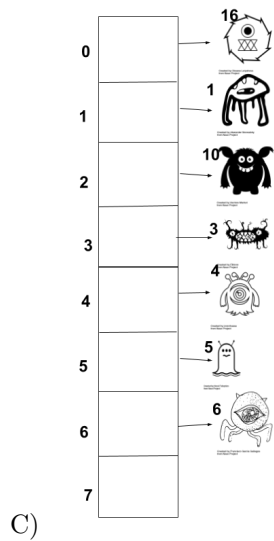
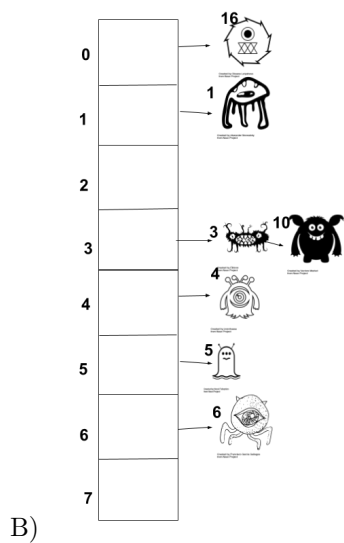
Now, after inserting that element, we want to insert a new element (shown below) **and** resize.



Which of the following correctly mirrors the new state?



A)



- A
- B
- C

2 Unexpected Hashing

Suppose we have the `Lamp` class below:

```

1  class Lamp {
2      int brightness;
3
4      Lamp(int brightness) {
5          this.brightness = brightness;
6      }
7
8      @Override
9      public int hashCode() {
10         return brightness;
11     }
12
13     @Override
14     public boolean equals(Object o) {
15         return ((Lamp) o).brightness == brightness;
16     }
17 }

```

Assume the `HashMap` is implemented with external chaining. Assume the size of the internal array of the `HashMap` is 2 and doesn't resize. Determine the output of each print line below:

```

1  Lamp a = new Lamp(1);
2  Lamp b = new Lamp(2);
3
4  HashMap<Lamp, Integer> map = new HashMap<>();
5
6  map.put(b, 0);
7  map.put(a, 1);
8  map.put(a, 2);
9
10 System.out.println(map.get(a)); // print statement 1
11 System.out.println(map.get(b)); // print statement 2
12
13 map.put(b, 3);
14 a.brightness = 2;
15 map.put(b, 4);
16
17 System.out.println(map.get(a)); // print statement 3
18 System.out.println(map.get(b)); // print statement 4
19 System.out.println(map.get(new Lamp(1))); // print statement 5

```

Print Statement 1:

- 0
- 1
- 2
- 3
- 4
- null

Print Statement 2:

- 0
- 1
- 2
- 3
- 4
- null

Print Statement 3:

- 0
- 1
- 2
- 3
- 4
- null

Print Statement 4:

- 0
- 1
- 2
- 3
- 4
- null

Print Statement 5:

- 0
- 1
- 2
- 3
- 4
- null