This is the midterm examination for
CSC212: Data Structures
as taught by R. Jordan Crouser and Nicholas R. Howe in Spring 2022.

The following materials are **permitted** while taking this examination:

- a single 8.5x11 sheet of paper (double-sided) containing your own handwritten or typed notes
- blank scratch paper (provided at the end of this packet)

Honor code: no other resources are permitted during this exam.
This includes (but is not limited to): textbooks, online materials, tutors, teaching assistants, and other students.

If you encounter any issues while taking this exam,
the instructors can be reached on Slack:
`@Jordan Crouser` and `@Nicholas Howe`

NAME: _____

SCORE: _____ **out of 40**

Question 0. Getting in the Groove (0 points)

Note: This question is optional, but strongly recommended.

Educational research studies[1,2] have suggested that people perform better on tests when they spend a few minutes thinking about things they're good at before they begin.

In the space below, briefly tell us about a time when you were **really successful** at doing something challenging (it doesn't have to be related to this course). If you prefer, you can draw a picture instead of writing.

[1]Lang, Jonas WB, and Jessica Lang. "Priming competence diminishes the link between cognitive test anxiety and test performance: Implications for the interpretation of test scores." *Psychological Science* 21.6 (2010): 811-819.

[2]Barrows, Jennifer, Samantha Dunn, and Carrie A. Lloyd. "Anxiety, self-efficacy, and college exam grades." *Universal Journal of Educational Research* 1.3 (2013): 204-208.

Question 1. Vocabulary (6 points)

Word Bank:

abstract	call signature	declare	exception	generic	inheritance
initialize	instance	interface	iterator	method	overload
override	public	recursion	static	type	void

Fill in the blank with the term or concept that matches each of the definitions below:

(a) To _____ a variable means to introduce the name of the variable and specify its type, e.g. `int x;`

(b) The _____ of a method refers to the method's name combined with the description (i.e. type, number, and position) of its parameters.

(c) A(n) _____ is an event that occurs during the execution of a program that disrupts the normal flow of of the program's instructions. The term is shorthand for the phrase "exceptional event."

(d) The term _____ refers to the situation where one class (called a parent class or superclass) passes its attributes and behavior down to another class (called a child class or subclass). The child class may then either add new behavior, or change (override) some of the behavior of the parent class.

(e) In the Java programming language, the keyword _____ indicates that the particular member belongs to a type itself, rather than to an instance of that type. This means that only one instance of that member is created, which is then shared across all instances of the class.

(f) A(n) _____ is a contract that specifies a required set of methods that a class must implement. If a class `implements` one of these, then it guarantees that this contract is fulfilled.

Question 2. Abstract Data Structures (12 points)

For each scenario below, identify the abstract data structure (`Array`, `Hash Table / Map`, `Set`, `Linked List`, `Queue`, or `Stack`) that best supports the requirements described, and provide a brief justification of your choice.

(a) A pirate wants to keep track of where different pieces of treasure are buried. He'd like to be able to name a piece of treasure ("giant ruby") and quickly use this information to figure out the latitude and longitude where the item was buried. What data structure do you recommend?

(b) On planet Froobius, it snows different colors on different days. On days when it is not snowing, warm weather melts the top layer of snow instead, revealing the previous layer underneath. What data structure would be best for tracking the snow colors?

(c) Imagine you are playing a card game where players each have a collection of cards that varies in size during the game (for example, as they play some cards or draw new ones). The cards can combine in different ways, so players need to be able to reorder the cards in their hand at any time. What data structure would you recommend for representing a player's hand?

...this question continued on the next page.

(d) You are sending a large file over a computer network. Due to the size, the file has to be broken into pieces (called "packets"), and each packet gets sent separately. The transit time can vary, so the packets do not necessarily arrive in the order in which they were sent. However, each packet has a label indicating where it falls in the overall order (for example: #15 of 200). What data structure do you recommend for reassembling the file at the receiving end?

(e) A cook in a particular restaurant is responsible for making all the food that customers order. There are several people on the waitstaff who collect the orders. Customers don't like to wait a long time for their food, so the cook wants to fulfill the order of whichever customer has been waiting the longest. What data structure would you recommend for keeping track of orders?

(f) You are in charge of shopping for refreshments and decorations for a large event. You would like to be able to walk through the store, looking at each of the items on display, and quickly determine whether this is something you need to buy. Assuming that the number of items sold and the number of different items you are buying are both fairly large, what data structure would you recommend?

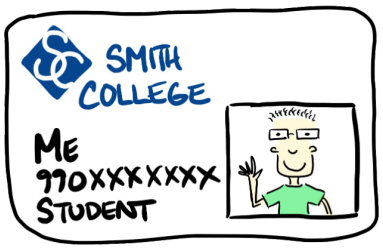
Question 3. Tracing Java Programs (4 points)

Consider the following program:

```
1  class NumberGame {
2
3    int startingNum;
4
5    public static void main(String[] args) {
6      NumberGame ng = new NumberGame(10);
7      ng.startGame();
8    }
9
10   public NumberGame(int n) {
11     startingNum = n;
12   }
13
14   public void startGame() {
15     play(startingNum);
16   }
17
18   private void play(int n) {
19     System.out.println(n);
20     if (n == 1) {
21         System.out.println("Done!");
22     } else if (n % 2 != 0) {
23         play(n*3+1);
24     } else {
25         play(n/2);
26     }
27   }
28 }
```

What output is printed to the console when we run this program?

Question 4. Hash Tables (4 points)



Early in this course we introduced an example of a `OneCard` class, which had the following fields:

- `String name`: to store a student's name
- `int id`: to store their 99-number
- `double balance`: to store the amount of money loaded onto the card

Suppose we now want to be able to look up additional data about a student given information we can get from their `OneCard`. We decide to do this using a **Map**, implemented as a `HashTable`.

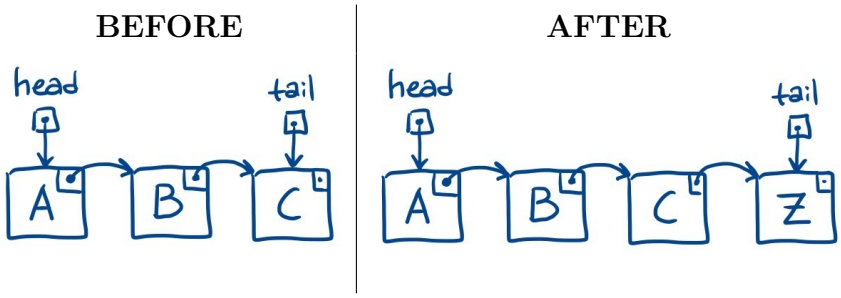
(a) Which of the three fields of OneCard (`name`, `id`, `balance`) would **not** be a good choice to use as a key? Explain why.

(b) We determine that we will need a table of size 10000, and we plan to use four digits from the `id` field as the key. Does it matter whether we take the first four or the last four digits? Why or why not?

Question 5. Operations on Data Structures (8 points)

In this question, we will present you with sketches various data structures **before** and **after** a single operation is performed. For each pair of images, determine the data structure and operation.

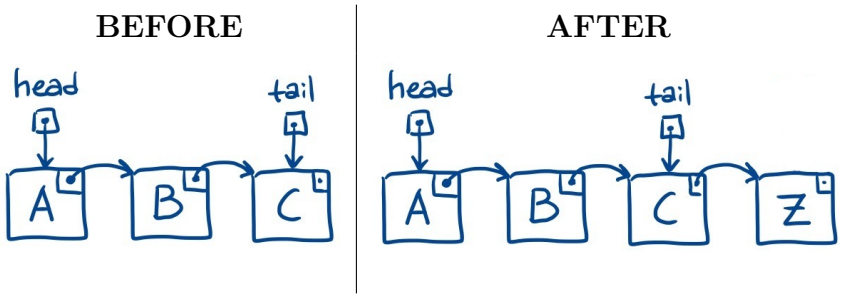
Example 1:



Sample response: this is a `LinkedList`, and the operation performed was `addLast(Z)`.

If the operation was not valid, describe what is wrong.

Example 2:

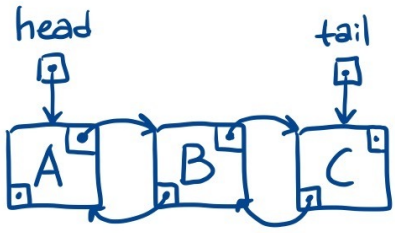


Sample response: this is a `LinkedList`, and it looks like they tried to `addLast(Z)` but forgot to update the `tail`.

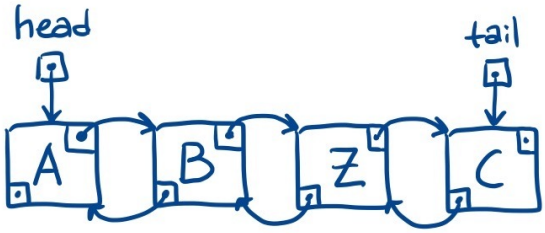
...this question continued on the next page.

(a)

BEFORE

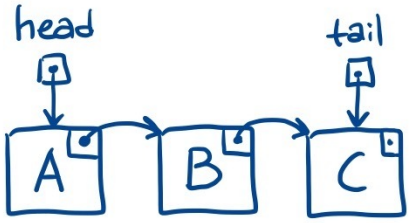


AFTER

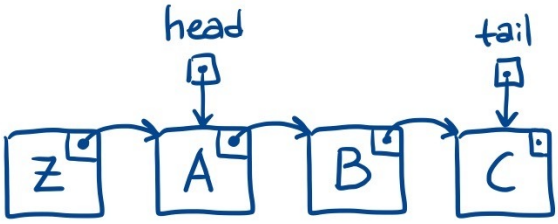


(b)

BEFORE



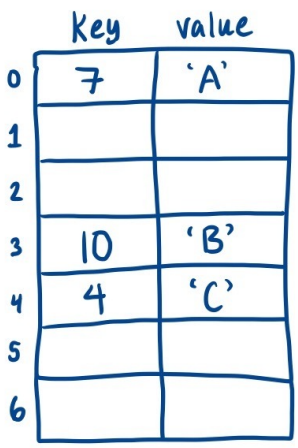
AFTER



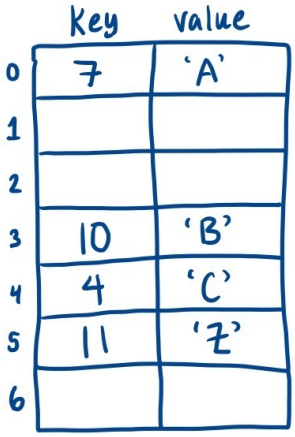
...this question continued on the next page.

(c)

BEFORE

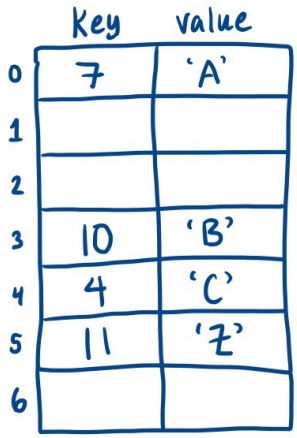


AFTER

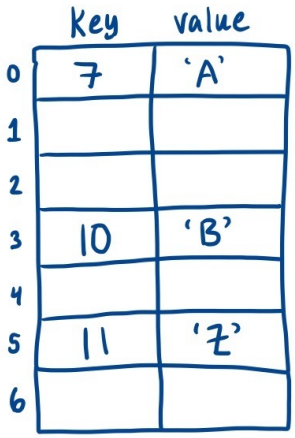


(d)

BEFORE



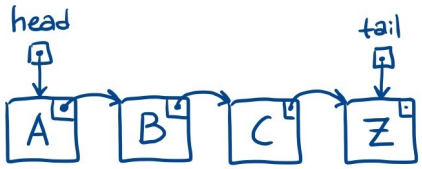
AFTER



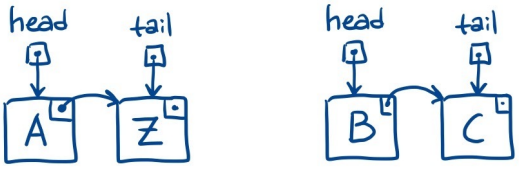
...this question continued on the next page.

(e)

BEFORE

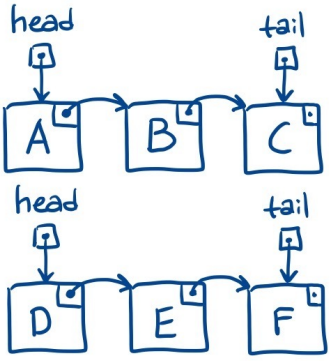


AFTER

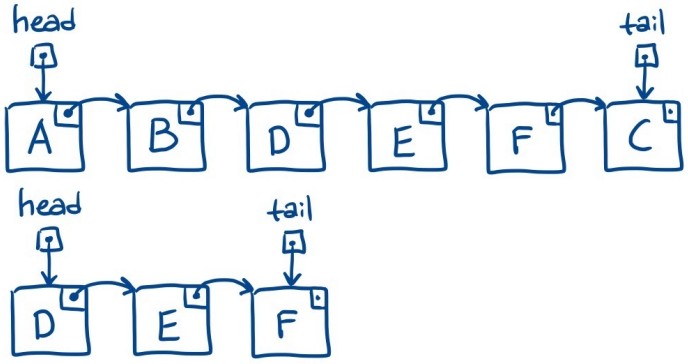


(f)

BEFORE



AFTER



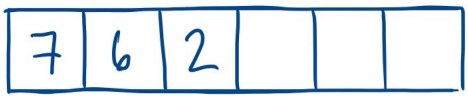
...this question continued on the next page.

(g)

BEFORE

top: 3

AFTER



top: 2

(h)

BEFORE

head: 3

numStored: 2

AFTER

head: 3

numStored: 3

Question 6. Efficiency - *Stretch Question* (6 points)

As we've explored various data structures in this course, there has often been an underlying question about how to perform certain operations **efficiently**. For each of the following operations, state how many elements in the sequence would be affected by the change:

(a) **inserting** an element in the middle of a 100-item sequence stored in an `Array`

(b) **inserting** an element at the middle of of a 100-item sequence stored in a `LinkedList`, given an iterator at the desired position

(c) **removing** the middle element of a 100-item sequence stored in an `Array`

(d) **removing** the middle element of a 100-item sequence stored in a `LinkedList`, given an iterator at the desired position

(e) **reading** the middle element of a 100-item sequence stored in an `Array`

(f) **reading** the middle element of a 100-item sequence stored in a `LinkedList`, without an iterator

This page intentionally left blank as scratch paper.

This page intentionally left blank as scratch paper.

This page intentionally left blank as scratch paper.