/*******************************

Name: Noah Buchanan
Username: ua203
Problem Set: PS6
Due Date: July 14, 2020
*******************************/

Data Structures

July 14, 2020

1. (a) An array implementation has a set size that is set at the very beginning but also requires a resize() method if the initial size is surpassed. Memory is accumulated during runtime for a linked list as well while nodes are added onto the stack, as for array it is set amount of memory until a resize is needed.

   (b) Both implementations methods have a constant runtime O(1) except for push() method for an array in the case that the array must be resized the resize() function is O(n) which is also called in the push() function.

   (c) An array implementation would be useful if you needed to access a location not at the top but it also takes up more initial space and needs a correct size unless you want to call a resize() function which is the only method that does not run on O(n) so it would run slower in the worse case. A linked list implementation accumulates memory as it moves and all methods run on O(1) so it is useful when you dont know how large the end result will be as there will be no resize needed.

2. (use case 1): looking for matches in an array or string or etc. for example looking for opening and closing parentheses and brackets to make sure its a valid expression.

   (use case 2): ctrl z and ctrl y are also a real world implementation of stacks, as you press ctrl z more it loads more items onto the stack and removes them as you go back through them.

(use case 3): flipping strings or arrays is also very simple with stacks as the LIFO method is reversing the order regardless, you push it onto the stack and as you take them off the stack and insert them it will be in reversed order.

1 UAStackArray

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

/*******************************
Name: Noah Buchanan
Username: ua203
Problem Set: PS6
Due Date: July 12, 2020
*******************************/

public class UAStackArray {

        int top = -1;
        int size;
        UAStudent[] array;
        int i = 0;

        public UAStackArray(int size) {

                array = new UAStudent[size];
        }

        public void push(UAStudent s) {
                if(top == (size-1)) {
                        resize();
                }
                if(s != null) {

                        top++;
                        array[top] = s;
                }
                size++;
        }

        public UAStudent pop() {

                UAStudent temp = array[top];
        if(top != 0) {
                array[top] = null;
                top--;
        }
        size--;
        return temp;
}

public int size() {
        return size;
}

public boolean isStackEmpty() {
        return size == 0;
}

public void resize() {
        UAStudent[] newArray = new UAStudent[(int) Math.ceil(array.length * 1.5)]
        for(int i = 0; i < array.length; i++) {
                newArray[i] = array[i];
        }
        array = newArray;
        size = array.length;
}



public static class UAStudent {

        private int studentId;
        private String firstName;
        private String lastName;

        public UAStudent(String input) {
                String[] x = input.split(",");

                this.firstName = x[0];
                this.lastName = x[1];
                this.studentId = Integer.parseInt(x[2]);

        }

        public int getStudentId() {
                return studentId;
        }
        public void setStudentId(int studentId) {
                this.studentId = studentId;
        }

        public String getFirstName() {
                return firstName;
        }

        public void setFirstName(String firstName) {
                this.firstName = firstName;
        }

        public String getLastName() {
                return lastName;
        }

        public void setLastName(String lastName) {
                this.lastName = lastName;
        }

        public String toString() {
                return "Student: \t" + lastName + ", " + firstName;
        }

}


public static void main(String[] args) throws IOException {

        UAStackArray st = new UAStackArray(100);

        BufferedReader br = new BufferedReader(new FileReader("Records.txt"));

        String line = "";

        while((line = br.readLine()) != null) {



                st.push(new UAStudent(line));
        }

        System.out.println("=== Start =============================");
        System.out.println("Size of Stack:  " + st.size() );
```

```
                System.out.println("\n\n");
                System.out.println("=== Inserts ==============================");

                UAStudent a = new UAStudent("Mackey,Andrew,44444");
                st.push( a );
                st.push( new UAStudent("Mackey,Andrew,55555") );
                st.push( new UAStudent("Mackey,Andrew,99999") );

                System.out.println("Size of Stack:  " + st.size() );

                System.out.println("\n\n");
                System.out.println("=== Deletes ==============================");
                System.out.println(st.pop());
                System.out.println(st.pop());

                System.out.println("Size of Stack:  " + st.size() );

                System.out.println(st.pop());

                System.out.println("Size of Stack:  " + st.size() );

                br.close();
        }



}
```

2 UALinkedList

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

/********************************
Name: Noah Buchanan
Username: ua203
Problem Set: PS6
Due Date: July 12, 2020
********************************/
public class UAStackLinkedList {

        UAStudent top;
        int size;
```

```
public void push(UAStudent s) {

        if(s != null) {
                s.next = top;
                top = s;
                size++;
        }
}

public UAStudent pop() {
        UAStudent temp = top;
        if(top != null) {
                top = top.next;
                temp.next = null;
                size--;
        }
        return temp;
}

public int size() {
        return size;
}

public boolean isStackEmpty() {
        return size == 0;
}

public static class UAStudent {

        private int studentId;
        private String firstName;
        private String lastName;
        UAStudent next;

        public UAStudent(String input) {
                String[] x = input.split(",");

                this.firstName = x[0];
                this.lastName = x[1];
                this.studentId = Integer.parseInt(x[2]);
        }

        public int getStudentId() {
                return studentId;
        }
```

```java
        public void setStudentId(int studentId) {
                this.studentId = studentId;
        }

        public String getFirstName() {
                return firstName;
        }

        public void setFirstName(String firstName) {
                this.firstName = firstName;
        }

        public String getLastName() {
                return lastName;
        }

        public void setLastName(String lastName) {
                this.lastName = lastName;
        }

        public String toString() {
                return "Student: \t" + lastName + ", " + firstName;
        }

}

public static void main(String[] args)throws IOException {

        UAStackLinkedList st = new UAStackLinkedList();

        BufferedReader br = new BufferedReader(new FileReader("Records.txt"));

        String line = "";

        while ((line = br.readLine()) != null) {

                st.push(new UAStudent(line));
        }

        System.out.println("=== Start ==============================");
        System.out.println("Size of Stack:  " + st.size());

        System.out.println("\n\n");
        System.out.println("=== Inserts ==============================");
```

```
                UAStudent a = new UAStudent("Mackey,Andrew,44444");
                st.push(a);
                st.push(new UAStudent("Mackey,Andrew,55555"));
                st.push(new UAStudent("Mackey,Andrew,99999"));

                System.out.println("Size of Stack:  " + st.size());

                System.out.println("\n\n");
                System.out.println("=== Deletes ==============================");
                System.out.println(st.pop());
                System.out.println(st.pop());

                System.out.println("Size of Stack:  " + st.size());

                System.out.println(st.pop());

                System.out.println("Size of Stack:  " + st.size());

                br.close();

        }

}
```