

## Part 1

For this assignment, the overall problem will be creating an array of characters, and having a method which receives a user input and compares it to an element of the array, and if the elements matches the input, the next element of the array will be compared to the next input. For this assignment, I will use a main method, which will populate an array of characters using a loop, open a scanner object and receive the user input, and depending on the user input it will call 1 of two methods, a method that iterates through the array of characters starting at the first index, or a method that starts at the last index. I will use a loop in the main method that will ask for a user input and check if the user input is either 's' or 'e', signalling if the user would like to start at the start or end of the array. It will look something like this:

```
While (*input is not s or e) {  
    System.print("enter an input");  
    userInput = scanner.nextLine();  
    if ( userInput == s ) {  
        AlphabetGameStart();  
    }  
    If (userInput == e ) {  
        AlphabetGameEnd();  
    }  
    Else {  
        System.print( "invalid input");  
    }  
    Scanner.close();
```

The two methods for the actual alphabet game themselves will look somewhat similar and will behave nearly identically, except the one that starts at the first index will start at the first index of the array and will iterate towards the end of the array. If there is an incorrect answer, the loop will not iterate to the next element and and the counter (presumably 'i') will be  $i = i-1$ ; for everytime there is an incorrect answer so as that the loop will ask to input the same requested value again. There will be a variable that will store the time when the method begins running, and another variable that stores the time when the method stops running. The time elapsed will be the difference between the two. This method will run until it reaches the end of the array. The other method that starts at the last index of the array will behave the same except the array will start at the last element and will iterate towards 0 (the first index).

## Part 2

```
import java.util.Scanner;  
public class Main {  
  
    public static void main(String args[]) {  
        int n = 26; //length of alphabet  
        char[] alphabet = new char[n];  
        for (int i = 0; i < n; i++) {  
            alphabet[i] = (char) ('a' + i); //populates the alphabet array
```

```

by using ASCII values that change through a loop
    }
    boolean choice = false; //boolean that represents a correct input
(s or e)
    Scanner scanner = new Scanner(System.in); //scanner is created
    while (!choice) { //the program will continue to ask for an input
as long as e or s has not been selected
        System.out.println("\nChoose to start from the start (type s) or
the end (type e)");
        String userInput = scanner.next();
        // user input is stored. equals method is used to compare the
user's input and decide if the alphabet is to be used from the start or the
end
        if ("s".equalsIgnoreCase(userInput.trim())) {
            choice = true; // boolean is updated and loop ends
            alphabetGameS(alphabet, n, scanner);
            break;
        }
        if ("e".equalsIgnoreCase(userInput.trim())) {
            choice = true; // boolean is updated and loop ends
            alphabetGameE(alphabet, scanner);
            break;
        } else {
            System.out.println("\nInvalid input. Valid inputs are 's'
or 'e'");
        }
    }
    scanner.close(); //scanner is closed
}

public static void alphabetGameS(char[] alphabet, int n, Scanner
scanner) {
    System.out.println("\nFirst letter is a");
    long startTime = System.currentTimeMillis(); //timer is started
    for (int k = 0; k < n; k++) {
        String answer = scanner.next();
        if(answer.equalsIgnoreCase(String.valueOf(alphabet[k])) && k <
25) { //checks if answer is correct and that the alphabet array hasn't
reached the end
            System.out.println("\nCorrect! Next letter is: " +
alphabet[k + 1]);
        } else if
(!answer.equalsIgnoreCase(String.valueOf(alphabet[k]))) {
            System.out.println("\nIncorrect! Try again");
            k -= 1; //this removes that answer as a guess as it wasn't
correct
        }
        if(answer.equalsIgnoreCase(String.valueOf(alphabet[n - 1])) &&
k == n - 1) { //once the last letter has been typed in and the array is on
its last element, the loop ends
            System.out.println("\nSuccess! Game Over");
            long endTime = System.currentTimeMillis(); //end time
            double totalTime = (endTime - startTime) / 1000.0; //total
time elapsed, in seconds
            System.out.println("\nTotal time taken: " + totalTime +
"seconds.");
            break; //game is finished
        }
    }
}
// this method works much the same as the other one, except the array is

```

```

used starting from the last element.
public static void alphabetGameE(char[] alphabet, Scanner scanner) {
    System.out.println("\nFirst letter is z");
    long startTime = System.currentTimeMillis();
    for (int k = 25; k >= 0; k--) { //loop instead starts at the end of
the array, and iterates towards zero, the first element
        String answer = scanner.next();
        if(answer.equalsIgnoreCase(String.valueOf(alphabet[k]))) && k >
0) {
            System.out.println("\nCorrect! Next letter is: " +
alphabet[k - 1]);
        } else if
(!answer.equalsIgnoreCase(String.valueOf(alphabet[k]))) {
            System.out.println("\nIncorrect! Try again");
            k += 1; // guess is removed as was incorrect
        }
        if(answer.equalsIgnoreCase(String.valueOf(alphabet[0]))) && k ==
0) { //once the last letter has been typed in and the array is on it's
first element, the loop ends
            System.out.println("\nSuccess! Game Over");
            long endTime = System.currentTimeMillis();
            double totalTime = (endTime - startTime) / 1000.0; //total
time elapsed, in seconds
            System.out.println("\nTotal time taken: " + totalTime + "
seconds.");
            break;
        }
    }
}

```

## Part 3

The screenshot shows the IntelliJ IDEA interface with the Main.java file open. The code is a simple Java program that prompts the user to choose between 's' (start) or 'e' (end). If an invalid input is provided, it prints an error message. The terminal window below shows the execution of the program, where the user types 'q', resulting in the output: "Invalid input. Valid inputs are 's' or 'e'".

```
19     break;
20 }
21 if ("e".equalsIgnoreCase(userInput.trim())) {
22     choice = true; // boolean is updated and loop ends
23     alphabetGame(alphabet, scanner);
24     break;
25 } else {
26     System.out.println("\nInvalid input. Valid inputs are 's' or 'e'"); //loop will continue to run as s or e hasn't been selected
27 }
28 }
29 scanner.close(); //scanner is closed
30 }
31 }

1 usage
```

```
Run Main x
Run Main x
@ : /Users/ciarangray/Library/Java/JavaVirtualMachines/openjdk-21.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=55958:/Ap
↑ ↓
Chose to start from the start (type s) or the end (type e)
q
Invalid input. Valid inputs are 's' or 'e'
Chose to start from the start (type s) or the end (type e)
```

The screenshot shows the IntelliJ IDEA interface with the Main.java file open. The code is identical to the first screenshot. The terminal window below shows the execution of the program, where the user types a sequence of letters: 'c', 'd', 'e', 'f', 'g', and 'a'. For each letter, the program prints "Correct! Next letter is: [letter]". After 'g', the user types 'a', which is followed by the message "Incorrect! Try again".

```
19     break;
20 }
21 if ("e".equalsIgnoreCase(userInput.trim())) {
22     choice = true; // boolean is updated and loop ends
23     alphabetGame(alphabet, scanner);
24     break;
25 } else {
26     System.out.println("\nInvalid input. Valid inputs are 's' or 'e'"); //loop will continue to run as s or e hasn't been selected
27 }
28 }
29 scanner.close(); //scanner is closed
30 }
31 }

1 usage
```

```
Run Main x
Run Main x
@ : Correct! Next letter is: c
c
Correct! Next letter is: d
d
Correct! Next letter is: e
e
Correct! Next letter is: f
f
Correct! Next letter is: g
g
a
Incorrect! Try again
```

The screenshot shows the IntelliJ IDEA interface with the Main.java file open. The code implements a game where the user inputs a letter, and the program prints the next letter in the alphabet. If the user inputs 'e', the loop ends. If the user inputs 's', the program prints "Success! Game Over". The run output window shows the following interaction:

```
a  
↓  
Incorrect! Try again  
s  
↓  
Incorrect! Try again  
y  
Correct! Next letter is: z  
z  
Correct! Next letter is: x  
x  
Success! Game Over  
Total time taken: 36.952 seconds.  
Process finished with exit code 0
```

The screenshot shows a terminal window with the command:

```
/Users/ciarangray/Library/Java/JavaVirtualMachines/openjdk-21.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=56507:/Application
```

The terminal output shows the game loop starting with 'e' input:

```
Chose to start from the start (type s) or the end (type e)  
e  
First letter is z  
z  
Correct! Next letter is: y  
y  
Correct! Next letter is: x  
x  
Correct! Next letter is: w  
w  
Correct! Next letter is: v  
v  
Correct! Next letter is: u  
u  
Correct! Next letter is: t  
t  
Correct! Next letter is: s  
s  
Correct! Next letter is: r  
r  
Correct! Next letter is: q  
q  
Correct! Next letter is: p
```

```
Run Main ×
Run Main ×
Correct! Next letter is: j
j
Correct! Next letter is: i
i
Correct! Next letter is: h
h
Correct! Next letter is: g
g
Correct! Next letter is: f
f
Correct! Next letter is: e
e
Correct! Next letter is: d
d
Correct! Next letter is: c
c
Correct! Next letter is: b
b
Correct! Next letter is: a
a
Success! Game Over
Total time taken: 30.85 seconds.
Process finished with exit code 0
```

Expected Output	Actual Output
Iterating through the alphabet array and displaying what the next letter will be	Encountered an error when I reached the last letter of the array and couldn't display the next letter
Comparing the current letter of the array to the user's input	Made the common mistake of using '==' instead of .equals to compare the characters