Hunter Poole
CSCI 155 HW5, Problem 1

Problem recap skipped due to length of problem.

Three Step Analysis:

A. Take any character as input. Return the entered character and its ASCII value.
B. Determine if the entered character is upper case, lower case, 0-9, or other. Store that info.
C. Return the next two characters
D. Loop until # is entered. Then return a table for the count of each character type.

INPUT	**OUTPUT**	**EQUATIONS**
Any char	The same char	**do…while** (Ch != '#')
# to quit	Char's ASCII value	**if** (Ch != '#')
	Next two characters (from input char)	**if** (Character.isUpperCase(Ch)) Uppercase++ **else if** (Character.isLowerCase(Ch)) Lowercase++ **else if** (Character.isDigit(Ch)) Digit++ **else if** (Ch != '#') Other++ **end if**
	Table of counts - display quantity of each type of char entered	

E. Limits / Constraints:
    a. Can only take one character at a time
    b. Cannot handle whitespace or null values.
        i. Only functions for ASCII characters within typeable range, excluding the space bar.
        ii. 33 - 126 (!, ~)

Hunter Poole
CSCI 155 HW5, Problem 1

```
char Ch, NextCh, NextCh2
int Ch_Value, Uppercase = 0, Lowercase = 0, Digit = 0, Other = 0

do
        write “Please enter your character: “
        read Ch
        Ch_Value = Ch

        if (Ch != ‘#’)
                write (Ch + “ “ + Ch_Value)
                NextCh = Ch
                NextCh2 = ++NextCh
                write ((NextCh++) + “ “ + (++NextCh2))
        end if

        if (Character.isUpperCase(Ch))
                Uppercase++
        else if (Character.isLowerCase(Ch))
                Lowercase++
        else if (Character.isDigit(Ch))
                Digit++
        else if (Ch != ‘#’)
                Other++
        end if

while (Ch != ‘#’)
end do-while

write (“Number of uppercase: “ + Uppercase + “Number of lowercase: “ + Lowercase + “Numbers: “ +
Digit + “Number of other characters: “ + Other)
```

src\Problem1.java

```
/*
Name: Hunter Poole
Date: 2/24/25
HW #: 5
Problem #: 1
Source Code: Problem1.java
Action: Takes a single char, returns char, ASCII value,
        and next two characters. Runs in a loop.
        Counts # of uppercase, lowercase, digits, and
        other characters entered. '#' to exit.
        Displays a table for counts of char types at exit.
*/

import java.util.Scanner;

public class Problem1
{
    public static void main(String[] args)
    {
        char Ch, NextCh, NextCh2;
        int Ch_Value, Uppercase = 0, Lowercase = 0, Digit = 0, Other = 0;

        Scanner Input = new Scanner(System.in);

        do
        {
            System.out.print("Please enter your character ---> ");
            Ch = Input.next().charAt(0);
            Ch_Value = Ch;

            if (Ch != '#')
            {
            System.out.printf("%n%s %c %n%s %c %s %d %n","You entered", Ch,
                              "The ASCII value of", Ch, "is", Ch_Value);

            NextCh = Ch;
            NextCh2 = ++NextCh;

            System.out.printf("%s %c %s %c %n%n", "The next two characters are:", NextCh++,
                           "and", ++NextCh2);
            }

            if (Character.isUpperCase(Ch))
            {
                Uppercase++;
            }
            else if (Character.isLowerCase(Ch))
```

```
            {
                Lowercase++;
            }
            else if (Character.isDigit(Ch))
            {
                Digit++;
            }
            else if (Ch != '#')
            {
                Other++;
            }

        }while (Ch != '#');

        System.out.printf("%n%s %d %n%s %d %n%s %d %n%s %d", "Number of uppercase--------->",
                        Uppercase,"Number of lowercase--------->", Lowercase,
                        "Number of numbers----------->", Digit,
                        "Number of other characters-->", Other);
    }
}

/*
Please enter your character ---> A

You entered A
The ASCII value of A is 65
The next two characters are: B and C

Please enter your character ---> v

You entered v
The ASCII value of v is 118
The next two characters are: w and x

Please enter your character ---> 5

You entered 5
The ASCII value of 5 is 53
The next two characters are: 6 and 7

Please enter your character ---> @

You entered @
The ASCII value of @ is 64
The next two characters are: A and B

Please enter your character ---> (

You entered (
```

```
The ASCII value of ( is 40
The next two characters are: ) and *

Please enter your character ---> #

Number of uppercase---------> 1
Number of lowercase---------> 1
Number of numbers-----------> 1
Number of other characters--> 2

    ////// Extremes \\\\\\

Please enter your character ---> !

You entered !
The ASCII value of ! is 33
The next two characters are: " and #

Please enter your character ---> ~

You entered ~
The ASCII value of ~ is 126
The next two characters are:  and ?

Please enter your character ---> #

Number of uppercase---------> 0
Number of lowercase---------> 0
Number of numbers-----------> 0
Number of other characters--> 2
 */
```

Hunter Poole
CSCI 155 HW5, Problem 2

2) Write a program that displays the first 40 Fibonacci numbers. A Fibonacci number is created by add the previous two, with the first two always being 0 and 1. A partial sequence is as follows: 0, 1, 1, 2, 3, 5, 8, 13, 21,.... Your table must display 6 numbers per row and use a spacing of 10 for each number. Don't forget to look at handout on formatting output and probably use "printf()".

Three Step Analysis:

A. Initialize two int variables at 0 & 1.
B. Add them, display result.
C. Write variable 2 into variable 1.
D. Write result into variable 2.
E. Add them, display result.
F. Repeat B-E for a total of 40 outputs.

INPUT	**OUTPUT**	**EQUATIONS**
	Fibonacci sequence, first 40 numbers.	**for** (i = 39; i > 0; i--)
	0, 1, 1, 2, 3, 5, 8, 13, 21,...	Num3 = Num1 + Num2
		if ((i + 1) % 6 == 1)

G. Limits / Constraints:
    a. Must have six numbers on a line
    b. Numbers must have spacing of 10
    c. Must go only to the 40th fibonacci number
        i. 102334155
        ii. Counted 0 as 0, 1 as 1, 2 as 1.
            1. https://planetmath.org/listoffibonaccinumbers

Hunter Poole
CSCI 155 HW5, Problem 2

```
int i, Num1 = 0, Num2 = 1, Num3

write Num1, Num2

for (i = 39; i > 0; i--)
        Num3 = Num1 + Num2
        write Num3

        Num1 = Num2
        Num2 = Num3

        if ((i + 1) % 6 == 1)
                write (%n)
        end if
end for
```

src\Problem2.java

```
/*
Name: Hunter Poole
Date: 2/24/25
HW #: 5
Problem #: 2
Source Code: Problem2.java
Action: Displays the first 40 fibonacci numbers in a table.
        Table to have 6 columns.
        Numbers in table have width of 10.
 */

public class Problem2
{
    public static void main(String[] args)
    {
        int i, Num1 = 0, Num2 = 1, Num3;

        System.out.printf("%-10d%-10d", Num1, Num2);

        for (i = 39; i > 0; i--)
        {
            Num3 = Num1 + Num2;
            System.out.printf("%-10d",Num3);

            Num1 = Num2;
            Num2 = Num3;

            if ((i + 1) % 6 == 1)
            {
                System.out.printf("%n");
            }
        }
    }
}

/*
0         1         1         2         3         5
8         13        21        34        55        89
144       233       377       610       987       1597
2584      4181      6765      10946     17711     28657
46368     75025     121393    196418    317811    514229
832040    1346269   2178309   3524578   5702887   9227465
14930352  24157817  39088169  63245986  102334155
 */


/*
```

```
Chose to count Fibonacci number 0 as 0, 1 as 1, 2 as 1, 3 as 2, etc
per https://planetmath.org/listoffibonaccinumbers
So, 102334155 is the 40th Fibonacci number with 0 and 1 counting as 0 and 1.

If wrong, update i = 39 --> i = 40, change if statement:
    if (i % 6 == 1)
Will print 165580141 (#41) as the last number, satisfying 6 num per row requirement.
*/
```

src\Problem3.java

```
/*
Name: Hunter Poole
Date: 2/24/25
HW #: 5
Problem #: 3
Source Code: Problem3.java
Action: Given a series of numbers, return how many of them are even.
        Exit program when "0" is entered.
*/

import java.util.Scanner;

public class Problem3
{

    public static void main(String[] args)
    {
        int Even_Count = 0;

        Scanner Input = new Scanner(System.in);
        System.out.print("Please provide your numbers one at a time. Enter 0 to exit: ");
        int Number = Input.nextInt();

        while (Number != 0)
        {
            if (Number % 2 == 0)
            {
                Even_Count++;
            }

            System.out.print("Next number: ");
            Number = Input.nextInt();
        }

        if (Number == 0)
        {
            System.out.print("You have entered " + Even_Count + " even numbers.");
        }

    }
}

/*
Please provide your numbers one at a time. Enter 0 to exit: 3
Next number: 56
Next number: 4
Next number: 13
```

```
Next number: 779
Next number: 46
Next number: 0
You have entered 3 even numbers.

Please provide your numbers one at a time. Enter 0 to exit: 905775
Next number: 13
Next number: 777
Next number: 346
Next number: 42
Next number: 12
Next number: 0
You have entered 3 even numbers.
*/
```

src\Problem4.java

```java
/*
Name: Hunter Poole
Date: 2/24/25
HW #: 5
Problem #: 4
Source Code: Problem4.java
Action: Given a whole number, displays multiples of 3 down to 3.
*/

import java.util.Scanner;

public class Problem4
{
    public static void main(String[] args)
    {
        Scanner Input = new Scanner(System.in);
        System.out.print("Provide your number: ");
        int Num = Input.nextInt();

        for (int Multiples = Num; Multiples >= 3; Multiples--)
        {
            if (Multiples % 3 == 0)
            {
                System.out.print(Multiples + "  ");
            }
        }
    }
}

/*
Provide your number: 16
15  12  9  6  3

Provide your number: 25
24  21  18  15  12  9  6  3

Provide your number: 70
69  66  63  60  57  54  51  48  45  42  39  36  33  30  27  24  21  18  15  12  9  6  3
 */
```