Challenge 1:

Get the nth number in the fibonacci sequence given n

Alternatively given a number F, print out whether it's a fibonacci number and what the closest index n in the

fibonacci sequence is.

Java;

Fn of Fibonacci numbers:

F{n} = F{n-1} + F{n-2} with the base value

F{0} =0 AND F{1}=1

class Main

{

public static int fib(int n)

{

If (n <= 1) {

return n;

}

return fib(n-1) + fib(n-2);

public static void main(String[] args)

{

Int n=8;

System.out.println(“F(n) =” + fib(n));

}

}

Output:21

Challenge 2:

Write a solution to find the character that has the highest number of occurrences within a certain string, ignoring

case. If there is more than one character with equal highest occurrences, return the character that appeared first

within the string.

For example:

Input: "Character"

Output: c

Java:

Import java.util.\*;

Public class Main {

Static final int N=256;

Static char MaxOccuringChar(String str1){

Int ctrl[] = new int[N];

Int l =str1.length();

For (int I = 0; I < l; i++)

Ctr[str1.charAt(i)]++;

Int max = -1;

Char result = ‘ ‘;

For (int i = 0;I < l;i++) {

If (max <ctr[str1.charAt(i)]) {

Max = ctr[str1.charAt(i)];

Result = str1.CharAt(i);

}

}

Public static void main(String[] args){

String str1 = “character”;

System.out.println(“The given string is: “ + str1);

System.out.println(“Max occurring character in the given string is: “ + MaxOccuringChar(str1));

}

}

Output:

The given string is: Character

one character with equal highest occurrences, return the character that appeared first

:C