# Coding Challenge 1 - Make Chocolate [Logic, mathematical]

We want make a package of **goal** kilos of chocolate. We have small bars (1 kilo each) and big bars (5 kilos each). Return the number of small bars to use, assuming we always use big bars before small bars. Return -1 if it can't be done.

public static int makeChocolate(int small, int big, int goal) {

}

makeChocolate(4, 1, 9) → 4  
makeChocolate(4, 1, 10) → -1  
makeChocolate(4, 1, 7) → 2

# Coding Challenge 2 – Arrowhead

This challenge involves the use of control statements. Prompt the user to provide the size of an arrowhead. The app will print an arrowhead in the console, line by line, for the # of lines given by the user… looking something like this:

>  
>>  
>>>  
>>  
>

Note: if the user enters an even # you’ll need to add a row to accommodate the middle row, the tip of the arrowhead.

# Coding Challenge 3 – First Last 6 [Arrays]

Given an array of ints, return true if 6 appears as either the first or last element in the array. The array will be length 1 or more.

firstLast6([1, 2, 6]) → true  
firstLast6([6, 1, 2, 3]) → true  
firstLast6([13, 6, 1, 2, 3]) → false

public boolean firstLast6(int[] nums) {

}

# Coding Challenge 4 – Rotate Left [Arrays]

Given an array of ints length 3, return an array with the elements "rotated left" so {1, 2, 3} yields {2, 3, 1}.

rotateLeft3([1, 2, 3]) → [2, 3, 1]  
rotateLeft3([5, 11, 9]) → [11, 9, 5]  
rotateLeft3([7, 0, 0]) → [0, 0, 7]

public int[] rotateLeft3(int[] nums) {

}

**\*BONUS\*** Modify your code so it accepts an array of any length, 3 or bigger

# Coding Challenge 5 – Replace Chars

Get a sentence from the user and return it with all the vowels replaced with underscores ( \_ ).

Enter a sentence: The fox jumped over the log.

Result: Th\_ f\_x j\_mp\_d \_v\_r th\_ l\_g.

public String replaceVowelsWithUnderscores(String s) {

}

# Coding Challenge 6 – String Bits (Working w/ Strings)

Given a string, return a new string made of every other char starting with the first, so "Hello" yields "Hlo".  
stringBits("Hello") → "Hlo"  
stringBits("Hi") → "H"  
stringBits("Heeololeo") → "Hello"

public String stringBits(String str) {

}

# Coding Challenge 7 – String Match

|  |
| --- |
| Given 2 strings, a and b, return the number of the positions where they contain the same length 2 substring. So "xxcaazz" and "xxbaaz" yields 3, since the "xx", "aa", and "az" substrings appear in both strings.  stringMatch("xxcaazz", "xxbaaz") → 3 stringMatch("abc", "abc") → 2 stringMatch("abc", "axc") → 0  public int stringMatch(String a, String b) {    } Coding Challenge 8 – StringSplosion Given a non-empty string like "Code" return a string like "CCoCodCode". stringSplosion("Code") → "CCoCodCode" stringSplosion("abc") → "aababc" stringSplosion("ab") → "aab"  public String stringSplosion(String str) {  } |

# Coding Challenge 9 – StopWatch (Date/Time)

Create an app that tracks time. You could start tracking time when the user enters the word ‘start’, then stop tracking when they enter ‘stop’.

Enter the word ‘start’ to start the timer.

Enter the word ‘stop’ to stop the timer.

That took 5 seconds!

HINT: There is a library available to do this. Google it and figure out how to implement.

# Coding Challenge 10 – Pig Dice

This is a dice roll game. A player rolls a dice, over and over again, until a 1 is rolled. A total/score is accumulated for each roll. When a 1 is rolled, the app should display the total score, number of rolls, and (bonus) any other interesting data you want to display.

Sample output:  
# of times to play?: 5  
Here we go….  
Played 5 games  
Highest Score is 82  
Again? N  
Bye!

Additional interesting stats to keep:

* Highest score
* Total rolls
* Highest rolls within a game
* Average rolls per game
* Average score per game
* Most common # rolled
* How long did we play?