**Java Refresher**

**Variables, Input, Output**

**Question 1**

Make a program to has three integers (d, h, and m), and display the number of seconds in d days, h hours, and m minutes (assume d, h, and m have values of 1 , 17 , 34 respectively).

**Question 2**

Make a program to swap the values of variables x and y.

**Question 3**

Create a program that takes an integer from the user and outputs just the last digit in the input value

**If Statements**

**Question 1**

Make a program to input an integer representing a month number and output the name of the month. If the integer is not a month number, a suitable reproach message should be displayed.

**Question 2**

Take three numbers from the user and print the greatest number.

**Question 3**

Write a program to find the number of days in a month by entering the month number and year.

**Question 4**

Write a program that takes a year from user and print whether that year is a leap year or not. Year is divisible by 4 or 400 but not 100

**Question 5**

Write program to allow the user to input their age. Then the program will show if the person is eligible to vote. A person who is eligible to vote must be older than or equal to 18 years old.

**Question 6**

Write a program to determine whether an input number is an even number.

**Question 7**

Write a program to take in 2 integer numbers and determine whether the first number is multiple of the second.

**Question 8**

Design a program to input three positive integers representing day, month and year, and output ‘yes’ or ‘no’ according to whether they form a legitimate date.

For Example:

Input 🡺 day = 10

month = 5

year = 1912

Output 🡺 yes

Input 🡺 day = 30

month = 2

year = 1958

Output 🡺 no

**Loop Statements**

**Question 1**

Create a program that takes 3 int numbers from the user (s,e,and i). Then create a loop that starts at s and counts to e in increments of i. Your program should print the numbers

Input s=2, e=11, I = 3

Output = 2, 5 ,8, 11

**Question 2**

Create a program that takes 1 number from the user and checks if it is prime. (You will need to use % to check for even division)

**Question 3**

Adapt Q2 to find all prime numbers up to the user input.

**Question 4**

Create a guessing game program. Your program should generate a random number between 1 and 100 (inclusive) and then ask the user for a guess. The game will keep asking for a guess until one of the following conditions are matched:

* The user guesses the number correctly (Win state)
* The user excesses 20 incorrect guesses (Lose state)
* The user enters a negative number (Quit state)

In addition, your program should do the following:

1. at the end, the computer prints the number of guesses made by the user,
2. each time the user guesses incorrectly, the computer says “too high” or “too low”

**Characters and Strings Manipulation**

**Question 1**

Create a program to compare 3 letter variables and output the largest.

**Question 2**

Create a program to increase the letter in your name by 1. E.g. Eoin = Fpjo

**Question 3**

Create a program take a String from the user. If the length is greater than 10 then print “Double” to the screen, otherwise print “Single”.

**Question 4**

Design a function **distance** that takes four fractional number parameters x0, y0, x1, and y1, and returns the distance between points (x0, y0) and (x1, y1). (Look up the formula if you can’t remember it!)

**Question 5**

Design a procedure **printString** to input an integer n and a string s, and print n copies of (the value of) n on the screen, two copies per line (except, possibly, the last line).

If the user enters a negative value for n, the computer should print a complaint.

**Question 6**

Design a function **numTxt** that takes an integer n as parameter and returns a string.

The value of n is assumed to be in the range 0…9, and the value returned is its equivalent textual form. For example, the value returned by **numTxt**(4) should be "four".

**Question 7**

Design a method (called **encode**) to accept a string ***s***, composed exclusively of

lowercase alphabetic characters and spaces from the keyboard. The method should also accept a positive integer ***n***. Your code should then apply a Caesar Cipher to the string and print the encoded string.

You should also create a test program to verify the method you have created.

**Tips:** A Caesar Cipher is applied individually to each letter in the string. Each letter must be shifted forward **n** steps in the alphabet. If a letter is shifted off the end of the alphabet (i.e. past ‘z’), then it is shifted all the way back around to the beginning of the alphabet (to ‘a’). Spaces are left unchanged during the encoding process.

**Examples:** encode(“hello world”,1) will return “idmmp xpsme”

encode(“hello world”,2) will return “jennq yqtnf” encode(“hello world”,3) will return “kfoor zruog”

encode(“hello world”,4) will return “lgpps asvph”

**Additional Questions**

|  |
| --- |
| **Question 1**  Complete the exercise below: FizzBuzz |
|  |

|  |
| --- |
| \* a) Print out the numbers 1 to 100 (inclusive) each on a new line. |
|  |

|  |
| --- |
| \* b) If the number to be printed is divisible evenly by 3, print Fizz instead |
|  |

|  |
| --- |
| \* c) If the number to be printed is divisible evenly by 5, print Buzz instead |
|  |

\* d) If the number to be printed is divisible evenly by both 3 and 5, print FizzBuzz instead

**Question 2**

|  |
| --- |
| Complete the exercise below: Stars |
|  |

|  |
| --- |
| a) As the user to enter a number |
|  |

|  |
| --- |
| b) Print that number of stars in a row |
|  |

|  |
| --- |
| c) on the next line, print the stars -1 and continue to print a line subtracting 1 each time |
|  |

|  |
| --- |
| d) When there are no more stars, print the pattern in reverse, increasing by 1 each time |
|  |

|  |
| --- |
| Example: |
|  |

|  |
| --- |
| Enter the number of stars: 5 |
|  |

|  |
| --- |
| \*\*\*\*\* |
|  |

|  |
| --- |
| \*\*\*\* |
|  |

|  |
| --- |
| \*\*\* |
|  |

|  |
| --- |
| \*\* |
|  |

|  |
| --- |
| \* |
|  |

|  |
| --- |
| \*\* |
|  |

|  |
| --- |
| \*\*\* |
|  |

|  |
| --- |
| \*\*\*\* |
|  |

|  |
| --- |
| \*\*\*\*\* |
|  |

Note only a single star in the centre...