2019 NORTHERN INDIANA CODING COMPETITION

April 13, 2019

COMPUTER SCIENCE AND INFORMATICS
COLLEGE OF LIBERAL ARTS AND SCIENCES
INDIANA UNIVERSITY SOUTH BEND

Round One

Problem 1. A normal year has 365 days. A Leap Year has 366 days (the extra day is the 29th of February). Here is the general algorithm to determine leap year.

- Leap Years are any year that can be exactly divided by 4 (such as 2012, 2016)
  - except if it can be exactly divided by 100, then it isn't (such as 2100, 2200)
    - except if it can be exactly divided by 400, then it is (such as 2000, 2400)

Write a program that takes a **four-digit** (1000-9999) year as input and outputs **yes** or **no** to indicate whether the input is a leap year.

Sample input (red color) and output (blue color):

Input: 1999
Output: no

Input: 2012
Output: yes

Input: 2100
Output: no

Input: 2400
Output: yes

Problem 2. A word palindrome is a word whose characters read the same backward as forward, such as *madam*, *racecar*, *redder*, or *civic*. A sentence palindromes may be written when allowances are made to ignore letter cases, punctuation, and word dividers, such as "A man, a plan, a canal, Panama!", "Was it a car or a cat I saw?"

Write a program that takes **a sentence** (1 to 200 characters) as input and outputs **yes** or **no** to indicate whether the sentence (word) is palindrome.

Sample input (red color) and output (blue color):

Input: Refer
Output: yes

Input: call
Output: no

Input: redder
Output: yes

Input: Was it a car or a cat I saw?
Output: yes

Input: Is it a car or a cat?
Output: no

Problem 3. The factorial of an integer n is defined as the product of all the numbers from 1 to n: n! = 1*2*...*n. It represents the number of ways we can arrange n objects, also known as permutations.

Given an integer number n, how many trailing 0s can be found on the right side of n!? For 5 it is 1, because 5! = 120. For 7! there is one 0, because it contains 5, which is divisible by 5, and 2, divisible by 2. However, for 26! the answer is 6: one zero contributed by 5, 10, 15, and 20, and two of them contributed by 25.

Write a program that takes **a positive number, n** (1 to 1000) as input and outputs **the number of trailing 0s of n!**.

Sample input (red color) and output (blue color):

Input: 3
Output: 0

Input: 7
Output: 1

Input: 12
Output: 2

Input: 78
Output: 18

Problem 4. Given **a string** (1-100 characters) composed of open and closed parentheses, and maybe some spaces in between, find out if the parentheses are well matched. This means that each open parenthesis must have a matching closed one to its right and that each closed parenthesis closing an opening parenthesis that has not been matched yet. The answer must be **yes** or **no**.

Sample input (red color, there might be spaces between parentheses) and output (blue color):

Input: ((( ) (( ))) ( ))
Output: yes

Input: (( )) (( ) ( ))

Output: yes

Input: ((( )) (( )
Output: no

Input: (( ))) (( )
Output: no

Problem 5. Mastermind is a game of two players. The first player chooses a secret code of 4 tokens selected from a set of 6 tokens of different colors. The second player must guess this sequence in 10 attempts. The first player will provide hints after each attempted guess, consisting of the number of tokens for which both the color and the position are correct (or an **exact** match), and number of tokens for which the color is correct but not the position (a **color** match).

Write a program that accepts a secret code and a guess, and returns the hint that will be provided for this guess.

The input will be a sequence of **4 non-repeating numbers** between 1 and 6, and a guess, which is a sequence **of the same type**. The output will consist of two numbers: count of **exact** matches, and count of **color (no position)** matches.

Sample input (red color, numbers are separated by a space) and output (blue color):

Input: 1 2 3 4
2 5 1 4
Output: 1 2

Input: 3 6 1 4
3 2 1 5
Output: 2 0

Input: 2 1 5 3
2 5 3 4
Output: 1 2

Problem 6. Given coordinates of **a source point** (x1, y1) and **a destination point** (x2, y2), determine if it is possible to reach from the source to the destination. At any point (x, y), there are only two types of valid movements: (x, x + y) and (x + y, y). Output **yes** if it is possible or **no** if it is not possible.

Examples:

For source point: (2, 10) and destination point: (26, 12), the answer is yes.
Because (2, 10)→(2, 12)→(14, 12)→(26, 12) is a valid path.

For source point: (20, 10) and destination point: (6, 12), the answer is no.
Because there is no such path.

Note: All coordinates are **positive integers less than 200**.

Sample input (red color, numbers are separated by a space) and output (blue color):

Input (x1, y1, x2, y2): 2 10 26 12
Output: yes

Input (x1, y1, x2, y2): 1 1 8 9
Output: yes

Input (x1, y1, x2, y2):11 10 20 21
Output: no