# Prueba de lógica

## Problema 1

Teniendo en cuenta la sucesión de Fibonacci:

**1,2,3,5,8,13,21...**

Encuentre la suma de los números impares de esta sucesión que no superen los 10 millones.

## Problema 2

Un número que se lee igual de izquierda a derecha y de derecha a izquierda se llama palíndromo. ¿Cuál es el número más grande con esta característica que resulta de la multiplicación de dos números de 3 dígitos?

## Problema 3

Encuentra la suma de todos los números primos que hay entre 2 millones y 3 millones, que además contengan el dígito 2 o el 7 pero no ambos dígitos al mismo tiempo

## Problema 4

¿Cuál es el valor de d menor a 1000 para el cual 1/d tiene el ciclo periódico más grande?

**Ejemplo:** Como 1/7=0.142857142857142857, el 1/7 tiene un ciclo periódico de 6 dígitos

## Problema 5

En la siguiente imagen se muestra una espiral de 5x5 en la que va incrementando cada número en una unidad.

**Calendario

Descripción generada automáticamente con confianza baja**

La suma de la diagonal de esta imagen da como resultado 101. ¿Cuánto da la suma de la diagonal de una espiral como esta de tamaño 1001x1001?

# Prueba de bases de datos

## Problema 1

Cree una vista donde se evidencie los bienes vendidos por país y año, ordenado de mayor a menor cantidad de ventas.

## Problema 2

Cree una función donde se evidencie los servicios vendidos por país y año, ordenado de mayor a menor cantidad de ventas.

## Problema 3

Indique cuales son las ventas totales de cada producto o servicio por país a lo largo de la historia.

# Prueba de optimización y solución de bugs

## Problema 1

Optimizar código

import numpy as np;

a = np.ones(50)

i = 0

t = 0

while i < 10:

i = i + 1

t = i \* 4

a[t] = t

## Problema 2

Optimizar código

import numpy as np

#Se tienen 6 graficas, se pueden comparar entre sí, teniendo en cuenta la matriz,

#el algoritmo debe recibir el id de la gráfica y devolver un vector que me indica que gráficas

#estan habilitadas y cuáles no.

m = [[1, 0, 1, 1, 0, 1],

[0, 1, 1, 0, 1, 1],

[1, 1, 1, 0, 1, 0],

[1, 0, 0, 1, 0, 1],

[0, 1, 1, 0, 1, 0],

[1, 1, 0, 1, 0, 1]]

print("Ingrese los valores, separados por comas")

values = input()

values = values.split(',')

r = []

s = []

for i in values:

r.append(m[int(i) - 1])

for i in range(0, 6):

b = 1

for j in range(0, len(values)):

b = r[j][i] \* b

s.append(b)

print(s)

## Problema 3

Optimizar código

#El algoritmo debe determinar si una palabra es palíndroma

# en lo posible usando una función recursiva

def isPalindrome(str):

p = "Es palindromo"

a = 0

b = len(str) - 1

while a < b:

if str[a] != str[b]:

p = "No " + p

break

a = a + 1

b = b - 1

print(p)

print("Ingrese la palabra")

word = input()

isPalindrome(word)

# Innovación y creatividad

NEU Energy nace bajo la visión de disrumpir en un mercado muy maduro donde la innovación era baja y estaba controlada por monopolios regionales, es por esto por lo que su concepción era ganar cuota de mercado con base a desarrollar mejoras innovadoras apoyadas de tecnología en las diferentes fases de la cadena. Esta mentalidad se fue esparciendo a través de la compañía, por lo cual las nuevas ideas siempre son bienvenidas.

Bajo esta primicia, se debe conformar equipos de hasta 4 personas que elaboren un proyecto tecnológico para resolver una necesidad del mercado, este proyecto deberá incluir:

Posible arquitectura de la solución.

Tecnologías para implementar.

Scketch a mano alzada de la UI de la solución.

Se deberá presentar en un periodo de no mas de 5 minutos ante el jurado.