Algoritmo de Hill

Integrantes: Nathan Moreno, Erick Toledo, Adrián Valverde

**Código:**

import numpy as np  
  
  
# Función para convertir texto en una lista de números  
def texto\_a\_numeros(texto):  
 numeros = []  
 for letra in texto.replace(' ', '').upper():  
 numeros.append(ord(letra) - ord('A'))  
 return numeros  
  
# Función para convertir una lista de números en texto  
def numeros\_a\_texto(numeros):  
 texto = ''  
 for numero in numeros:  
 texto += chr((numero % 26) + ord('A'))  
 return texto  
  
# Función para multiplicar un bloque por la matriz de clave  
def multiplicar\_matriz\_bloque(clave, bloque):  
 resultado = np.dot(clave, bloque) % 26  
 return resultado  
  
# Función para cifrar el texto  
def cifrar(texto, clave):  
 n = clave.shape[0]  
 texto\_numeros = texto\_a\_numeros(texto)  
  
 # Añadir ceros al final del texto si no es divisible por n  
 while len(texto\_numeros) % n != 0:  
 texto\_numeros.append(0)  
  
 texto\_cifrado = []  
 for i in range(0, len(texto\_numeros), n):  
 bloque = np.array(texto\_numeros[i:i + n])  
 bloque\_cifrado = multiplicar\_matriz\_bloque(clave, bloque)  
 texto\_cifrado.extend(bloque\_cifrado)  
  
 return numeros\_a\_texto(texto\_cifrado)  
  
# Función para descifrar el texto  
def descifrar(texto\_cifrado, clave):  
 n = clave.shape[0]  
 texto\_cifrado\_numeros = texto\_a\_numeros(texto\_cifrado)  
  
 # Calcular la inversa de la clave en módulo 26  
 clave\_inversa = np.linalg.inv(clave)  
 clave\_inversa = np.round(clave\_inversa).astype(int) % 26  
  
 texto\_descifrado = []  
 for i in range(0, len(texto\_cifrado\_numeros), n):  
 bloque = np.array(texto\_cifrado\_numeros[i:i + n])  
 bloque\_descifrado = multiplicar\_matriz\_bloque(clave\_inversa, bloque)  
 texto\_descifrado.extend(bloque\_descifrado)  
  
 return numeros\_a\_texto(texto\_descifrado)  
  
clave = np.array([[6, 24, 1],  
 [13, 16, 10],  
 [20, 17, 15]])  
  
texto = 'BIENVENIDOS A ALGEBRA LINEAL'  
# Cifrar y descifrar el texto  
texto\_cifrado = cifrar(texto, clave)  
texto\_descifrado = descifrar(texto\_cifrado, clave)  
  
# Mostrar resultados  
print("Texto original:", texto)  
print("Texto cifrado:", texto\_cifrado)  
print("Texto descifrado:", texto\_descifrado)

C