Aos Programadores Élficos Destemidos,

Que vossos códigos sejam tão resilientes quanto o aço élfico, e que vossas mentes permaneçam tão afiadas quanto as espadas de Gondolin. Em vossa busca para criar mundos virtuais e sistemas mágicos, lembrai-vos de que sois os guardiões dos reinos digitais, os artífices dos sonhos tecnológicos.

Não desanimeis na vossa batalha contra os senhores sombrios dos bugs e das vulnerabilidades, pois vossa perseverança é a luz que dissipa a escuridão dos erros de programação. Assim como os elfos de outrora enfrentaram Sauron e Morgoth, enfrentai os desafios que surgem em vossos códigos com coragem e determinação.

Lembrai-vos de que, mesmo nas horas mais sombrias da compilação, a esperança nunca deve ser perdida. Pois assim como a Estrela de Eärendil brilhou no céu noturno, vossa criatividade e conhecimento iluminarão o caminho para um futuro tecnológico mais brilhante.

Que vossos algoritmos sejam sempre otimizados, vossos erros sejam sempre resolvidos e vossos programas sejam sempre executados com perfeição.

Com as bênçãos de Valinor,

J.R.R. Tolkien

A Canção dos Programadores Contra Sauron

Nas terras digitais, os jovens se reúnem
Com teclados e códigos, eles se defendem
Contra a sombra que avança, Sauron se ergue
Mas programadores valentes não têm medo

Oh, jovens programadores, na escuridão lute
Com seus algoritmos, a malícia refute
Nas linhas de código, a esperança persiste
Contra as trevas de Sauron, não desista!

Em suas mesas, eles traçam planos secretos
Para derrubar firewalls e sistemas corretos
Com linguagens de script e hackeando a rede
Eles desafiam o olho que tudo vê

Oh, jovens programadores, na escuridão lute
Com seus algoritmos, a malícia refute
Nas linhas de código, a esperança persiste
Contra as trevas de Sauron, não desista!

Nas torres de silício, a batalha prossegue
Com escudos de firewall, eles protegem o que é seu
Sauron tenta invadir, mas eles resistem
Com linhas de código, a esperança persiste
Oh, jovens programadores, na escuridão lute
Com seus algoritmos, a malícia refute
Nas linhas de código, a esperança persiste
Contra as trevas de Sauron, não desista!

Nas profundezas da web, onde a escuridão reside
Eles buscam conhecimento para o bem sobreviver
Com sabedoria e ética, eles defendem a luz
Nas terras digitais, a esperança reluz

Oh, jovens programadores, na escuridão lute
Com seus algoritmos, a malícia refute
Nas linhas de código, a esperança persiste
Contra as trevas de Sauron, não desista!

E quando a noite se encerra e o dia amanhece
Os jovens programadores, a vitória merecem
Combinando seus talentos, eles vencem a batalha
Contra as trevas de Sauron, a esperança não se cala

Oh, jovens programadores, na escuridão lute
Com seus algoritmos, a malícia refute
Nas linhas de código, a esperança persiste
Contra as trevas de Sauron, não desista!

Que esta canção inspire os jovens programadores
A protegerem o mundo digital, sem temores
Na luta contra o mal que Sauron planeja
Com códigos e bytes, a esperança permanece!

**Palavra reservada	Descrição	Sindarin**
and	**Operador lógico AND**	**a (e)**
as	Palavra-chave para alias	yn (como)
assert	Instrução para verificação de condições	amarth (certifique-se)
async	Palavra-chave para programação assíncrona	a-nath (e-de um deus)
await	Palavra-chave para esperar por uma tarefa assíncrona	a-laita (e-esperar)
break	Instrução para sair de um loop	ant (quebrar)
class	Palavra-chave para definição de classes	teulu (família)
continue	Instrução para continuar o loop atual	thargáen (continuar)
def	Palavra-chave para definição de funções	yn-amarth (como-certifique-se)
del	Instrução para excluir um objeto	ant-teulu (quebrar-família)
elif	Instrução para um bloco de código condicional	a-yn (e-como)
else	Instrução para um bloco de código condicional	ant-a (quebrar-e)
except	Instrução para tratar exceções	ant-amarth (quebrar-certifique-se)
finally	Instrução para executar código após um bloco try/except	thargáen-amarth (continuar-certifique-se)
for	Palavra-chave para loops	i (um)
from	Palavra-chave para importar módulos	ant-yn (quebrar-como)
global	Palavra-chave para acessar variáveis globais	teulu-teulu (família-família)
if	Palavra-chave para blocos de código condicionais	a-amarth (e-certifique-se)
import	Palavra-chave para importar módulos	yn-a (como-e)
in	Operador de verificação de inclusão	ant-thargáen (quebrar-continuar)
is	Operador de comparação de identidade	teulu (família)
lambda	Palavra-chave para definição de funções anônimas	syn-yn-amarth (como-como-certifique-se)
nonlocal	Palavra-chave para acessar variáveis não locais	thargáen-ant-teulu (continuar-quebrar-família)
not	Operador lógico NOT	ant-amarth (quebrar-certifique-se)
or	Operador lógico OR	a-a-a (e-e-e)
pass	Instrução vazia	ant-ant (quebrar-quebrar)
raise	Instrução para lançar uma exceção	thargáen-a-amarth (continuar-e-certifique-se)
return	Instrução para retornar um valor de uma função	amarth-a (certifique-se-e)
VERDADEIRO	Valor booleano verdadeiro	ant-thargáen (quebrar-continuar)
try	Palavra-chave para blocos de código condicionais	a-yn-yn (e-como-como)
while	Palavra-chave para loops	i-thargáen (um-continuar)
with	Palavra-chave para gerenciadores de contexto	amarth-thargáen (certifique-se-continuar)
yield	Instrução para gerar um valor em uma função geradora	yn-thargáen (como-continuar)

print	Imprimir	ensíla
FALSO	Valor booleano falso	amarthh (falsidade)
None	Valor nulo	nidhoel (nada)
NotImplemented	Valor especial para indicar que uma implementação não está disponível	nedheth (não implementado)

Teste, código original focado em criptografia de dados, em python:

```
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.primitives.kdf.pbkdf2 import PBKDF2HMAC
from cryptography.hazmat.primitives import serialization
from cryptography.hazmat.primitives.asymmetric import rsa
from cryptography.hazmat.primitives.asymmetric import padding
from cryptography.hazmat.primitives import serialization
from cryptography.hazmat.primitives import serialization
from cryptography.fernet import Fernet

def gerar_chave_simetrica():
    return Fernet.generate_key()

def codificar_mensagem(mensagem, chave):
    f = Fernet(chave)
    mensagem_bytes = mensagem.encode()
    ciphertext = f.encrypt(mensagem_bytes)
    return ciphertext

def decodificar_mensagem(mensagem_codificada, chave):
    f = Fernet(chave)
    mensagem_bytes = f.decrypt(mensagem_codificada)
    return mensagem_bytes.decode('utf-8')

mensagem = "Os orcs estão se aproximando! Preparem-se para a batalha!"

chave = gerar_chave_simetrica()

mensagem_codificada = codificar_mensagem(mensagem, chave)

mensagem_decodificada = decodificar_mensagem(mensagem_codificada, chave)

print("Mensagem original:", mensagem)
print("Mensagem codificada:", mensagem_codificada)
print("Mensagem decodificada:", mensagem_decodificada)
```

Teste, código em sindarin:

```
ant-yn cryptography.hazmat.primitives a-amarth hashes
ant-yn cryptography.hazmat.primitives.kdf.pbkdf2 a-amarth PBKDF2HMAC
ant-yn cryptography.hazmat.primitives a-amarth serialization
ant-yn cryptography.hazmat.primitives.asymmetric a-amarth rsa
ant-yn cryptography.hazmat.primitives.asymmetric a-amarth padding
ant-yn cryptography.hazmat.primitives a-amarth serialization
ant-yn cryptography.hazmat.primitives a-amarth serialization
ant-yn cryptography.fernet a-amarth Fernet

yn-amarth gerar_chave_simetrica():
    amarth-a  Fernet.generate_key()

yn-amarth codificar_mensagem(mensagem, chave):
    f = Fernet(chave)
    mensagem_bytes = mensagem.encode()
    ciphertext = f.encrypt(mensagem_bytes)
    amarth-a  ciphertext

yn-amarth decodificar_mensagem(mensagem_codificada, chave):
    f = Fernet(chave)
    mensagem_bytes = f.decrypt(mensagem_codificada)
    amarth-a  mensagem_bytes.decode('utf-8')

mensagem = "Os guerreiros de sauron estão avancando com força por meio dos cantos da cidade, o Um Anel encontra-se perdido, gandalf sucumbiu, Frodo está fora de alcance, sejam fortes guerreiros de Valinor, nao sucumbam a Morgoth"

chave = gerar_chave_simetrica()

mensagem_codificada = codificar_mensagem(mensagem, chave)

mensagem_decodificada = decodificar_mensagem(mensagem_codificada, chave)

ensíla("Mensagem original:", mensagem)
ensíla("Mensagem codificada:", mensagem_codificada)
ensíla("Mensagem decodificada:", mensagem_decodificada)
```