

Trabajo practico integración N°1

Alumnos: Ayala, Nahuel // Arminchiardi, Marcela // Aramayo, Marisabel Viviana // Archiria, Facundo Miguel.

Tema elegido: Conversión de números.

Consigna: Desarrollen un programa que convierta números decimales a binarios y, de forma opcional, también de binario a decimal.

Extensión: Validar la entrada y mostrar mensajes de error ante datos incorrectos.

Introducción

Para la realización de este programa trabajamos con los conceptos matemáticos de los sistemas numéricos decimal y binario. Estos conceptos nos ayudaron a entender la estructura de números de estas características para poder implementar la conversión de binario a decimal y viceversa.

Se utilizaron conceptos vistos en programación 1:

- Declaración y asignación de variables.
- Operaciones aritméticas.
Transformación de número a cadena y viceversa.
- Estructura de control de flujo (if, elif, else)
- Bucles (for, while)

Uso de Inteligencia Artificial

En este proyecto hemos utilizado la IA para que nos ayude a realizar la validación de que el número ingresado sea un binario, recurrimos a esta herramienta al no encontrar un método que busque coincidencias dentro de un número.

A continuación documentamos a través de capturas de pantalla la conversación con la IA (ChatGPT), a la que le solicitamos las opciones existentes para realizar este tipo de validaciones, para continuar con la opción que nos parecía mas adecuada para este programa, también le solicitamos la explicación de conceptos de programación que utilizó y no logramos comprender.

Preciso realizar una validación de entrada de un número en python, saber si es un número binario.
¿Existe un método que me deje buscar coincidencia dentro de un número, ya sea del modulo math o nativo de python?

Sí, en Python puedes validar si un número es binario (es decir, que solo contiene dígitos 0 y 1) de varias maneras, aunque no hay un método específico en el módulo `math` para esto. Lo más práctico es usar métodos nativos de Python como:

Opción 1: Convertir el número a cadena y verificar sus caracteres

```
python Copiar Editar

def es_binario(n):
    return all(c in '01' for c in str(n))
```

Opción 2: Usar expresiones regulares (`re` es parte de la librería estándar)

```
python Copiar Editar

import re

def es_binario(n):
    return bool(re.fullmatch(r'[01]+', str(n)))
```

Opción 3: Verificar aritméticamente (menos común)

```
python ↓ Copiar Editar
```

Vamos a utilizar la opción 1 de pasarlo a cadena y luego buscar la coincidencia:
¿De la siguiente forma es correcta? se realizó dentro de un if, podrías corregirlo si verificas un error o una mejora a realizar:

```
#Transformar int a string
binario_string = str(binario)

if '01' in binario_string:
```



Estás en el camino correcto al convertir el número a string, pero la condición `if '01' in binario_string:` no valida si el número es binario, sino si existe la subcadena "01" en algún lugar del número.

Por ejemplo:

```
python                                                                    Copiar Editar

binario = 12301
binario_string = str(binario)
print('01' in binario_string) # Esto da True, aunque no es un número binario
```

✓ Corrección

Si lo que quieres es verificar que todos los caracteres del número sean 0 o 1, deberías usar:

```
python                                                                    Copiar Editar

if all(c in '01' for c in binario_str)
```



all, ¿Es un método que busca todas las coincidencias?
Y la variable c, ¿es un ejemplo utilizado o es una variable que se le asigna '01' con in?

Muy buenas preguntas, vamos a explicarlo con claridad:

♦ ¿Qué hace `all()` ?

`all()` es una función **nativa de Python** que:

- Recibe un iterable (por ejemplo, una lista o una expresión generadora).
- Devuelve `True` si todos los elementos del iterable son verdaderos.
- Devuelve `False` si al menos uno es falso.

Ejemplo simple:

python

 Copiar


 Editar

```
all([True, True, True]) # → True  
all([True, False, True]) # → False
```

◆ ¿Qué es `c in '01'`?

En el ejemplo:

python

 Copiar


 Editar

```
all(c in '01' for c in binario_string)
```

- Aquí `c` es una **variable temporal** que representa **cada carácter** dentro de la cadena `binario_string`.
- El bucle `for c in binario_string` recorre uno por uno los caracteres de la cadena.
- Luego, `c in '01'` evalúa si ese carácter `c` está dentro de la cadena `'01'`, es decir, si es `'0'` o `'1'`.

Entonces la expresión completa:

python

 Copiar

 Editar

```
all(c in '01' for c in binario_string)
```

Significa: "devuelve True solo si todos los caracteres del string están en `'01'` (es decir, son ceros o unos)."