

Práctica 2

Conversión a "Números Glíficos" Antiguos

En una civilización antigua olvidada, Aethelgard, se utilizaba un sistema numérico peculiar basado en un conjunto limitado de "glifos" y reglas de combinación muy específicas. Su sistema no era puramente posicional ni puramente aditivo/sustractivo como el romano, sino una mezcla compleja que generaba una representación única para cada número.

Su tarea es escribir un programa que convierta un número entero positivo en su representación "glífica" de Aethelgard, con la dificultad adicional de que existen reglas de "compactación" que buscan la representación más corta posible, similar a cómo el romano usa IV en lugar de IIII.

Reglas del Sistema Glífico de Aethelgard:

1. Glifos Fundamentales y sus Valores:

- Σ : 1
- Λ : 5
- Ω : 10
- Δ : 50
- Φ : 100
- Ψ : 500
- Ξ : 1000

2. Reglas de Combinación Básicas (Aditivas):

- Los glifos se leen de izquierda a derecha.
- Un glifo de mayor valor puede preceder a uno de menor valor, y sus valores se suman.
 - Ejemplo: $\Omega\Sigma = 10+1=11$
 - Ejemplo: $\Psi\Phi\Lambda = 500+100+5=605$

3. Reglas de Compactación (Sustractivas y Multiplicativas Implícitas): Esta es la parte compleja que busca la representación más corta y única. Aethelgard no repetía glifos más de 3 veces consecutivas, y además, usaban ciertas combinaciones especiales:

- **"Glifo Elevado"**: Un glifo de menor valor colocado *inmediatamente antes* de un glifo de mayor valor significa una resta del menor al mayor. **Esta regla solo aplica para los siguientes pares exactos, y se prioriza sobre cualquier otra combinación aditiva si resulta en una cadena más corta:**
 - $\Sigma\Lambda = 4$ ($5-1$)
 - $\Sigma\Omega = 9$ ($10-1$)
 - $\Lambda\Omega = 5$ ($10-5$) (Nota: esto es para evitar " $\Sigma\Sigma\Sigma\Sigma$ " que no existe, Λ es la forma estándar de 5. Pero la regla de glifo elevado permite esta compactación " $\Lambda\Omega$ ")
 - $\Omega\Delta = 40$ ($50-10$)
 - $\Omega\Phi = 90$ ($100-10$)

- $\Delta\Phi = 50$ (100–50)
- $\Phi\Psi = 400$ (500–100)
- $\Phi\Xi = 900$ (1000–100)
- **Restricción de Repetición:** Ningún glifo puede repetirse más de 3 veces consecutivas. Si al generar una secuencia se repite 4 veces, se debe buscar la forma compactada.
 - Ejemplo: 4 no es $\Sigma\Sigma\Sigma\Sigma$, sino $\Sigma\Lambda$.
 - Ejemplo: 9 no es $\Sigma\Sigma\Sigma\Sigma\Sigma\Sigma\Sigma$, ni $\Lambda\Sigma\Sigma\Sigma\Sigma$, sino $\Sigma\Omega$.
- **Glifos de '5' no se repiten:** Los glifos Λ , Δ , Ψ (los equivalentes a 5, 50, 500) **nunca** se repiten. Se usa el siguiente glifo de mayor valor o una combinación sustractiva.
 - Ejemplo: 10 no es $\Lambda\Lambda$, sino Ω .
 - Ejemplo: 100 no es $\Delta\Delta$, sino Φ .

Objetivo Adicional (Máxima Dificultad): La Representación *Canónica*

Para un número dado, puede haber varias combinaciones de glifos que sumen el mismo valor (ej. 15 podría ser $\Omega\Lambda\Lambda\Lambda\Lambda$ (inválido) o $\Omega\Lambda\Sigma\Sigma\Sigma\Sigma$ (válido pero largo) o $\Omega\Lambda\Sigma\Sigma\Sigma$ o $\Omega\Lambda\Lambda\Lambda\Lambda$ (Inválido) o incluso $\Omega\Lambda$. La meta es encontrar la representación **canónica**, que es la secuencia de glifos más corta posible que representa el número, siguiendo estrictamente las reglas de compactación. Si hay múltiples secuencias de la misma longitud mínima, se prefiere aquella que utilice glifos de mayor valor primero.

Entrada:

El archivo de texto `glifos.txt` consistirá en una serie de líneas, cada línea conteniendo un entero x . La entrada terminará cuando la línea contenga un 0. Los números a convertir serán positivos y menores a 4000.

Validación de Entrada:

- El programa debe validar que cada línea contenga un número entero.
- El número debe estar en el rango de 1 a 3999 (inclusive).
- Si una línea es inválida (no numérica o fuera de rango y no es 0 de terminación), debe registrar un error en el archivo de salida y omitir la conversión.

Salida:

El archivo de texto `salidaglifos.txt` contendrá una línea por cada entrada:

- Si la entrada es válida, imprima el número en decimal y su representación glífica canónica.
- Si la entrada es inválida, imprima el contenido original de la línea de entrada seguido de un mensaje de error claro.

Ejemplo de Entrada (`glifos.txt`):

```
3
8
172
4
```

```
1990
9
49
3999
1
4000
abc
0
```

Ejemplo de Salida (salidaglifos.txt):

```
3 $\Sigma\Sigma\Sigma$
8 $\Lambda\Sigma\Sigma\Sigma$
172 $\Phi\Lambda\Omega\Sigma\Sigma$
4 $\Sigma\Lambda$
1990 $\Xi\Phi\Omega\Phi$ (o $\Xi\Phi\Omega\Phi$ si se permite M C M X
C) -> Para este problema buscaremos la forma compacta mas corta M C M
X C (1000 + 900 + 90) = 1990. En este caso es $\Xi\Phi\Xi\Omega\Phi$
9 $\Sigma\Omega$
49 $\Omega\Lambda\Sigma\Lambda$ (O $\Omega\Lambda\Sigma\Lambda$ - No
$\Omega\Sigma\Lambda$ - No $\Omega\Lambda\Sigma\Lambda$) -> 49 es
$\Omega\Lambda\Sigma\Lambda$
3999 $\Xi\Xi\Xi\Phi\Xi\Omega\Phi\Sigma\Lambda$
1 $\Sigma$
4000 ERROR: Número fuera de rango (1-3999).
abc ERROR: Formato de entrada inválido. No es un número.
```

Este problema requiere no solo la lógica de conversión, sino también una cuidadosa construcción de la tabla de mapeo para garantizar la forma más corta y única, que es un desafío algorítmico en sí mismo y si para hacerlo más interesante no tenemos archivos precargados Quitar los archivos precargados de configuración y generación de paquetes eleva la dificultad al requerir que toda la interacción sea en **tiempo real. ¡EXITOS!**