
A continuación se presentan dos problemas: uno para codificar una cadena, y el otro para decodificarlo.

Adjuntamos un proyecto GOLD con un codificador y un decodificador. Tanto el codificador como el decodificador que aparecen en dicho proyecto, solo copian la cadena leída. Ustedes deben modificar el codificador y el decodificador para resolver los problemas planteados.

Entreguen un zip con este proyecto. El zip debe nombrarse con los logins de los integrantes del grupo.

1 [60%] Defina e implemente en GOLD:

Un automata con respuestas que recibe cadenas formadas por cadenas de letras minúsculas separadas por un único blanco y que terminan en punto:

$$\omega_1_ \omega_2_ \dots _ \omega_n.$$

(Donde $_$ es un único blanco). El autómata produce una cadena de la forma:

$$\beta_1 d_1 \beta_2 d_2 \dots \beta_n d_n.$$

Donde:

- $d_i = (\text{longitud}(\omega_i) + i) \% 5$
- β_i se obtiene de ω_i así: Si $\omega_i = \sigma_1 \dots \sigma_m$ entonces $\beta_i = \sigma_1 \rho_2 \dots \rho_m \sigma_1$ donde

$$\rho_i = \begin{cases} (j \% 3) & \text{si } \sigma_j = \sigma_1 \\ \sigma_j & \text{de lo contrario} \end{cases}$$

El autómata debe rechazar cadenas que comiencen con un blanco, tengan más de un blanco entre subcadenas o antes del punto. Así como cadenas que tengan símbolos después del punto.

Al final encuentra ejemplos de la ejecución del codificador.

2 [40%] Defina e implemente en GOLD:

Un autómata con respuestas que decodifique las cadenas codificadas con el autómata del punto anterior. Deben verificar que la codificación sea correcta.

- la primera la letra no debe aparecer en la mitad.
- la primera letra debe aparecer al final.
- ρ_j debe tener el valor correcto según la posición en la subcadena
- d_i debe tener el valor correcto.
- Los números deben aparecer en las posiciones de adecuadas.

EJEMPLOS

```
,
1 Input String empty string to exit):
2 exercises for codingdecoding are easy using exlipsegold.
3 The string was accepted!. Output:
4 ex0rcis2se0forf0codingde0odingc2area2easye4usingu1exlips1golde3.
5 -----
6 Input String empty string to exit):
7 concocting examples to test this project is not that easy.
8 The string was accepted!. Output:
9 con1o0tingc1exampl1se0tot0tes1t3thist4projectp3isi4notn1tha1t3easye4.
10 -----
11 Input String empty string to exit):
12 aaaaaa bbbbbbbb ccccc.
13 The string was accepted!. Output:
14 a2012a1b201201b4c2012c3.
15 -----
```