

Utilitar text cu compresie RLE

Scopul acestui exercițiu este să construiți o aplicație web în JavaScript care implementează un utilitar de compresie și decompresie text folosind algoritmul RLE (Run-Length Encoding). Aplicația trebuie să aibă o interfață simplă (HTML + CSS) în care utilizatorul poate introduce text, îl poate comprima sau decompresa și poate vedea informații despre raportul de compresie, timpul de execuție și validarea compresiei.

Obiective

- Înțelegerea și implementarea algoritmului de compresie RLE.
- Lucrul cu DOM, evenimente și manipularea textului în JavaScript.
- Utilizarea unor principii de curățenie a codului și modularizare.
- Lucrul cu fișiere, versionarea codului și debugging.
- Documentarea codului și instrucțiuni de folosire a aplicației.

Conceptul RLE

Algoritmul **Run-Length Encoding** transformă grupurile de caractere consecutive identice într-o pereche caracter–număr.

Exemple:

aaabbccc → a3b2c3

abc → a1b1c1

Decompresia este operația inversă: reconstruiește textul inițial din forma comprimată.

Cerinte

Modul easy

Implementați funcțiile de compresie și decompresie care:

- primesc un șir de caractere și returnează forma procesată a textului;
 - compresie: (ex: "aaabb" → "a3b2")
 - decompresie: (ex: "a3b2" → "aaabb")
- caracterele de input pot conține și cifre nu doar litere (aaaabbbb222cccc)

Interfață grafică:

- Se cere o interfață grafică (folosind HTML, CSS & JS) care să conțină următoarele elemente:
 - O zonă de text pentru input;
 - O zonă de text read-only pentru output
 - Două butoane: **Comprimă**, **Decomprimă**;
 - Afisarea unor metrice:
 - lungimii inputului și outputului;
 - raportului de compresie (output length / input length);
 - timpul de execuție al procesării;
 - mesaj de status („Compresie reușită”, „Eroare în format” etc.);

Documentație

- Adăugați documentație în cod despre algoritmul făcut
 - comentarii explicative acolo unde credeți că este necesar
- Adăugați un fișier README.md care să conțină informații despre cum se folosește/pornește/testează aplicația;

Restricții

- Nu se folosește nicio bibliotecă externă;
- Nu se folosește AI pentru generarea codului;

Modul Hard

- Pentru partea de UI se va folosi ReactJS.
 - Compresia si decompresia trebuie sa suporte caractere speciale (emojis, etc)
 - Posibilitatea de încărcare a textului prin upload de fișier
- Implementați funcțiile de compresie si decompresie (aceleasi ca la modul Easy)

Interfață grafică folosind React JS:

- Se cere o interfata grafica care sa conțină următoarele elemente:
 - Două zone textarea (input și output read-only)
 - Butoane funcționale „Comprimă” și „Decomprimă”
 - Afisarea unor metrice (aceleasi ca la modul Easy)
- Aplicația trebuie sa fie impartite pe componente, folosind elemente specifice React (states, hooks etc)

Functionalitate extinsa:

- Aplicația trebuie sa suporte și caractere speciale (emoji-uri, caractere asiatice etc)
- Posibilitatea de a incarca un fișier text (care contine textul propus spre a fi comprimat/decomprimat)
 - rezultatul va apărea în același textarea rezervat output-ului
- Posibilitatea de a încărcă fișierul text prin drag & drop

Restricții

- Nu se folosește nicio librărie externă (doar pentru UI: React, jQuery, Bootstrap, Material UI, etc);
- Nu se folosește AI pentru generarea codului;

Barem

Easy — 100 puncte (max. 110 cu bonus)

Funcționalitate de bază (40p total)

- Funcția `compress()` funcțională – 20p
- Funcția `decompress()` funcțională – 20p

Interfață grafică (30p total)

- Două zone textarea (input și output read-only) – 15p
- Butoane funcționale „Comprimă” și „Decomprimă” – 15p

Metrici și validări (20p total)

- Calcul și afișare lungime input/output – 5p
- Calcul raport de compresie – 5p
- Calcul timp de execuție – 5p
- Afișare mesaj de status („Compresie reușită”, „Eroare în compresie/decompresie” etc.) – 5p

Code quality (10p)

- Documentație proiect +5p
- Cod modular, reutilizabil (funcții separate, module) – 5p

Bonus (10p)

- Compresie < 0.5 – 10p
- Compresie < 0.6 – 5p

Hard — 200 puncte (max. 220 cu bonus)

Funcționalitate de bază (30p total)

- Funcția `compress()` funcțională (ex: „aaabb” → „a3b2”) – 15
- Funcția `decompress()` funcțională (ex: „a3b2” → „aaabb”) – 15

Interfață grafică folosind React JS (60p total)

- Două zone textarea (input și output read-only) – 15p
- Butoane funcționale „Comprimă” și „Decomprimă” – 15p
- Impartirea aplicației pe componente 30p

Metriци și validări (20p total)

- Calcul și afișare lungime input/output – 5p
- Calcul raport de compresie – 5p
- Calcul timp de execuție – 5p
- Afișare mesaj de status („Compresie reușită”, „Eroare în compresie/decompresie” etc.) – 5p

Code quality (10p)

- Documentație proiect +5p
- Cod modular, reutilizabil (funcții separate, module) – 5p

Funcționalitate extinsă (80p total)

- Suport pentru caractere speciale (Emoji, etc.) – 30
- File upload - 30
- Drag and drop - 20p

Bonus (20p)

- Compresie <0.5 - 20p
- Compresie <0.6 - 10p