Tema 1

Responsabil: Irina Mocanu

Tabele Hash

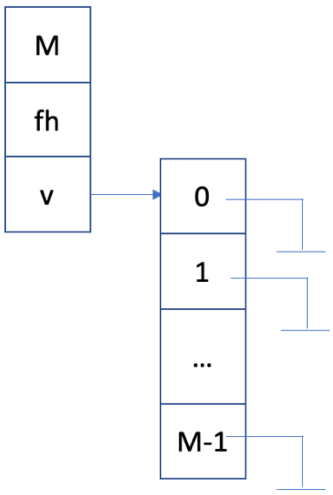
Deadline: 10.04.2022 (ora 23:59)

Se dorește realizarea unei statistici pentru analiza unui fișier text. De exemplu dorim sa aflam care sunt frecventele cuvintelor din text, frecventa unor cuvinte cu o anumita lungime, etc.

In acest scop este necesar sa construi o tabela hash in care sa păstram cuvintele întâlnite. Se vor considera cuvinte, șirurile de caractere (de lungime >= 3) formate din litere mici/mari, ,-,.

Exemplu:

Tabela hash construita va avea următoarea structura (Figură 1):



Figură 1. Tabela hash vidă

1. Cerință:

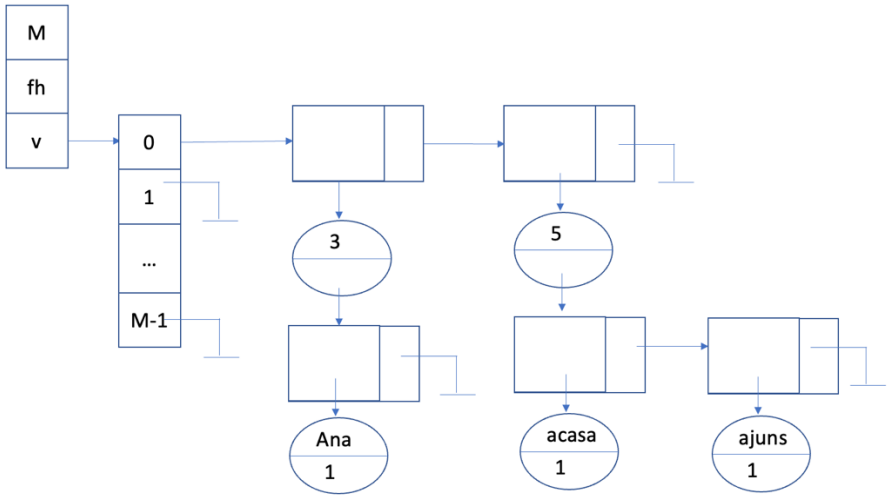
Tema presupune să realizați un program care să implementeze următoarele funcționalități:

- Inserare cuvinte: **insert text**
  In care text este format dintr-o succesiune de cuvinte.

  Exemplu: se considera o tabela hash vida in care se va insera următorul text:

  insert Ana a ajuns acasa.

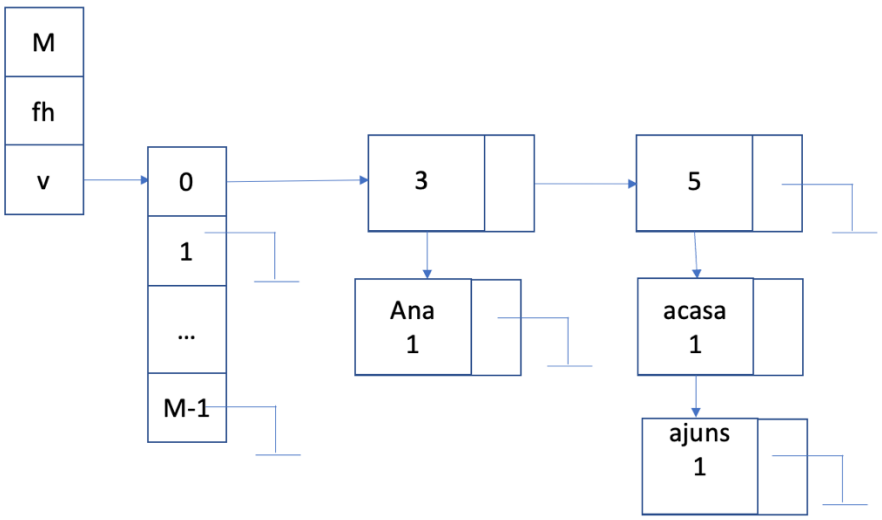
  Tabela hash va avea structura din Figură 2:



Figură 2. Tabela hash dupa inserare

Fiecare element din vectorul v (din tabela hash) este o lista simplu înlănțuită generica, care va conține cuvintele care încep cu caracterul din alfabet aflat pe poziția asociata intrării din vector. Aceasta lista va conține cuvintele împătrite in subliste pe baza lungimilor acestora. Pentru exemplul dat prima celula din lista conține cuvintele de lungime 3 (in acest caz „Ana”), iar a doua celula de lista va conține lista cuvintelor de lungime 5 (in acest caz: „acasa”, „ajuns”). Lista este ordonata crescător in funcție de lungimile cuvintelelor stocate. In sublistele asociate sunt păstrate cuvintele, împreună cu numărul de apariții – sublistele sunt sortate descrescător in functie de numărul de apariții a cuvintelor; in cazul frecventelor egale, sortarea se va realiza lexicografic (se folosește rezultatul întors de funcția strcmp). Implementarea sublistelor va fi realizata tot sub forma unei liste simplu înlănțuite generice.

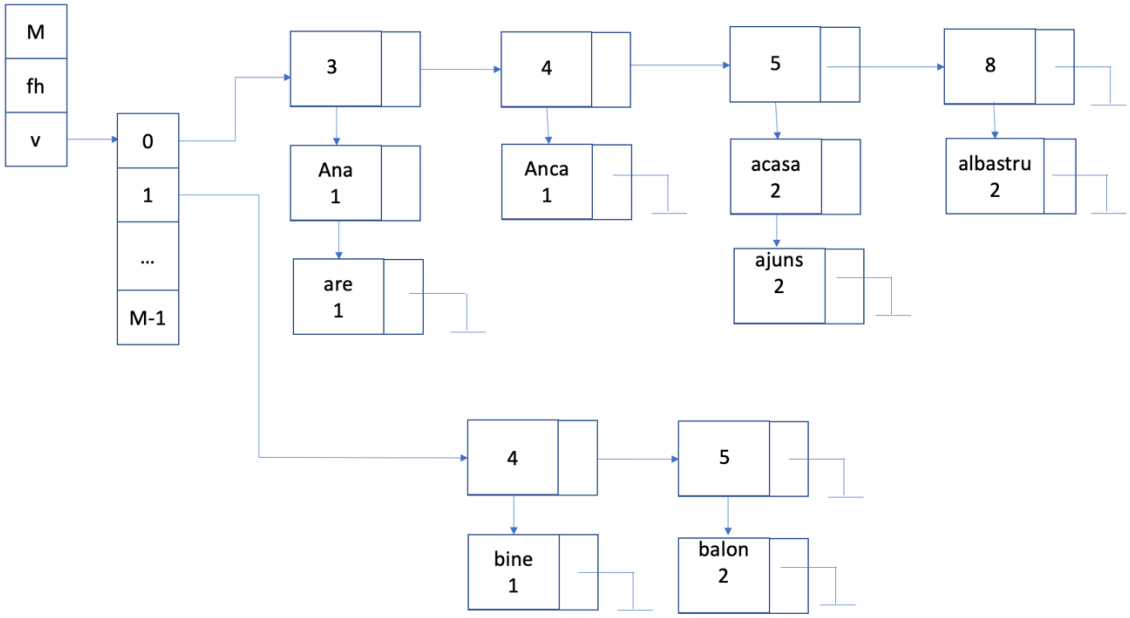
In continuare vom considera reprezentarea elementelor din tabela hash (Figură 3):



Figură 3. Reprezentare tabela hash

In tabela hash din Figură 3 se vor considera următoarele comenzi de inserare:

insert E un balon albastru.
insert Anca are un balon albastru.
insert acasa e bine
Tabela hash rezultata este prezentata in Figură 4.



Figură 4. Tabela hash dupa comenzile de inserare

- afișare tabela hash: **print**

  se afișează conținutul întregii tabele hash.

  Exemplu: **print** (aplicat pe tabela hash din Figură 4):

  pos0: (3:Ana/1, are/1) (4:Anca/1) (5:acasa/2, ajuns/2) (8:albastru/2)
  pos1: (4:bine/1) (5:balon/2)

  Nu se afișează pozițiile din vectorul v pentru care listele sunt vide.

- afișare cuvinte care încep cu o anumita litera si au o anumita lungime: **print c n**, unde c reprezintă prima litera a cuvintelor, iar n lungimea acestora

  Exemplu: **print a 5** (aplicata pentru tabela hash din Figură 4):
  (5:acasa/2, ajuns/2)

- afișare cuvine care apar de maxim n ori: **print n**, unde n reprezintă numărul maxim posibil de apariții

Exemplu: **print 1** (aplicat pe tabela hash din Figură 4):

pos0: (3:Ana/1, are/1) (4:Anca/1)
pos1: (4:bine/1)

2. Notare

- **85 puncte** obținute pe testele de pe vmchecker
- **10 puncte: coding style**, codul trebuie să fie comentat, consistent și ușor de citit. De exemplu, tema nu trebuie sa conțină (alte reguli găsiți in [1]):
    - Warning-uri la compilare
    - linii mai lungi de 80 de caractere
    - tab-uri amestecate cu spații
    - denumire neadecvată a funcțiilor sau a variabilelor
    - folosirea incorectă de pointeri, neverificarea codurilor de eroare
    - utilizarea unor metode ce consumă resurse în mod inutil (alocare de memorie)
    - neeliberarea resurselor folosite (eliberare memoriei alocate, ștergerea fișierelor temporare, închiderea fișierelor)
    - alte situații nespecificate aici, dar considerate inadecvate
- **5 puncte: README** – va conține detalii despre implementarea temei, precum și punctajul obținut la teste (la rularea pe calculatorul propriu)
- **Bonus: 20 puncte** pentru soluțiile ce nu au memory leak-uri (bonusul se va considera numai în cazul în care a fost obținut punctajul aferent testului)
- **Temele care nu compilează, nu rulează sau obțin punctaj 0 la teste, indiferent de motive, vor primi punctaj 0**
- **Temele neprezentate la laborator (la o data ce va fi ulterior anunțata) vor fi notate cu 0 puncte (indiferent de punctajul obținut pe vmchecker).**

3. Reguli de trimitere a temelor

Temele vor fi încărcate pe vmchecker (în secțiunea Structuri de Date seria CB: SD-CB), dar și pe cs.curs.pub.ro, în secțiunea destinată assignment-ului "Tema1".

Arhiva finala a temei rezolvate trebuie sa conțină:

- fișierele sursă
    - Fiecare fișier sursă creat sau modificat trebuie să înceapă cu un comentariu de forma: **/* NUME Prenume - grupa */**
- fișierul README în care va fi detaliat modul de implementare al rezolvării
- fișierul Makefile cu trei reguli (build și clean)
    - fișierul trebuie obligatoriu denumit **Makefile** și trebuie sa conțină cele 2 reguli menționate
    - Regula **build** va compila sursele și va crea executabilul numit **tema1**
    - Regula run - rulează tema conform cerințelor din enunțul temei
    - Regula **clean** care va șterge executabilele create

Arhiva va conține numai fișierele menționate mai sus (nu se acceptă fișiere executabile sau obiect).

Dacă arhiva nu respectă aceste specificații, aceasta nu va fi acceptată la upload și implicit tema nu va fi luată în considerare.

4. Reguli împotriva copierii temelor

- se consideră copiate doua teme care seamănă suficient de mult pentru a putea trage aceasta concluzie;
- modificarea unei alte teme, asemănarea mai mult sau mai puțin evidentă a implementării, bucăți de cod identice etc. duc la considerarea temelor în cauză ca fiind copiate;
- pentru prima temă copiată: atât sursei cât și destinației li se va anula punctajul pentru tema respectivă şi ambii studenţi vor primi mustrare scrisă, fără discuții relative la cine a copiat de la cine și a cui e vina;
- la a doua temă copiată: atât sursei cât și destinației li se va anula punctajul pentru toate temele (ceea ce va duce la repetarea materiei) şi ambii studenţi vor primi mustrare scrisă, fără discuții relative la cine a copiat de la cine și a cui e vina.

5. Referințe

[1] https://ocw.cs.pub.ro/courses/programare/coding-style