

Proba scrisă a examenului de licență 2026
Informatică Română
VARIANTA 2

NOTĂ

- Toate subiectele sunt obligatorii. La toate subiectele se cer rezolvări cu soluții complete.
- Nota minimă ce asigură promovarea este 5.00.
- Timpul efectiv de lucru este de 3 ore.

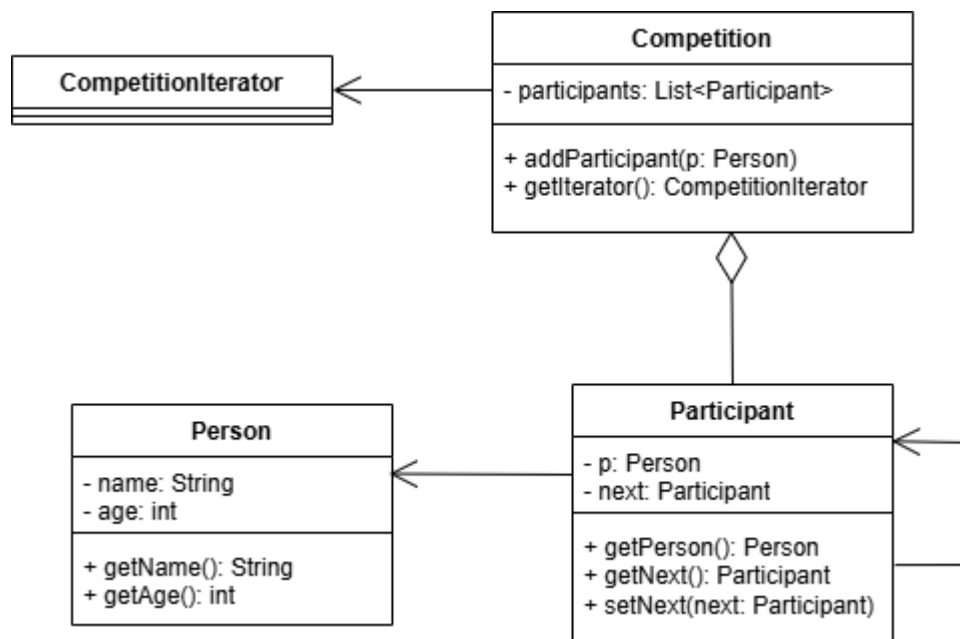
SUBIECT Algoritmica și programare

- Se va indica limbajul de programare folosit.
- În implementările cerute nu este necesară dealocarea zonelor de memorie alocate dinamic.
- Lipsa unui stil de programare adecvat (denumiri sugestive pentru variabile, indentarea codului, comentarii, dacă sunt necesare, lizibilitatea codului) duce la pierderea a 10% din punctajul aferent subiectului.
- Nu adăugați alte atribute, metode, în afara celor menționate în enunț, cu excepția constructorilor și a destructorilor, dacă e cazul. Nu modificați vizibilitatea atributelor specificate în enunț.
- Nu se vor folosi containere sortate, operații de sortare și căutare predefinite.
- Pentru tipurile de date puteți folosi biblioteci existente (Python, C++, Java, C#).

La o competiție se înscriu participanți din 7 categorii de vârstă: c_0 (5-9 ani), c_1 (10-19 ani), c_2 (20-29 ani), c_3 (30-39 ani), c_4 (40-49 ani), c_5 (50-59 ani) și c_6 (60-69 ani). Participanții vor fi memorați pe categorii de vârstă, în ordinea $c_0, c_1, \dots, c_6$, în fiecare categorie de vârstă participanții fiind memorați în ordine alfabetică.

1. (2.5 puncte) Scrieți o funcție într-unul din limbajele de programare **Python**, **C++**, **Java** sau **C#**, care primește ca parametri o listă *lst* de 7 liste de șiruri de caractere ordonate alfabetic (pe poziția i în lista *lst* sunt memorate, în ordine alfabetică, numele participanților de la categoria de vârstă c_i), un număr natural *varsta* și un șir de caractere *nume* (reprezentând vârsta și numele unui participant) și inserează numele participantului în lista corespunzătoare categoriei de vârstă, astfel încât această listă să rămână ordonată alfabetic. Nu se vor folosi funcții de inserare predefinite în biblioteci pentru inserarea unui element pe o poziție în interiorul unei liste.
2. (1 punct) Precizați complexitatea timp în cazurile *favorabil*, *mediu* și *defavorabil* pentru funcția de la punctul 1. Justificați răspunsul.
3. (5.5 puncte) Se dă următoarea diagramă UML conținând clasele **Competition** (Competiție), **Participant** (Participant), **Person** (Persoană), **CompetitionIterator** (Iterator Competiție).

Notă Constructorii claselor nu sunt indicați pe diagramă.



- Clasa **Competition** are un atribut *participants* care este o listă având dimensiunea egală cu numărul de categorii de vârstă. Pe poziția i în lista *participants* sunt memorați participanții de la categoria de vârstă c_i . Clasa are următoarele metode:
 - o **addParticipant(p: Person)** care adaugă persoana **p** în lista asociată categoriei sale de vârstă.
 - o **getIterator()** care returnează un iterator pentru parcurgerea (iterarea) participanților la competiție.
- Clasa **Participant** reprezintă o persoană dintr-o listă de persoane care participă în competiție într-o anumită categorie de vârstă. Participanții din fiecare categorie de vârstă sunt memorați în ordine alfabetică a numelor acestora.
- Clasa **Person** memorează numele (*name*) și vârsta (*age*) unei persoane care participă la competiție. Clasa are metode pentru returnarea atributelor. Vârsta persoanelor este un număr natural din intervalul [5-69].
- Clasa **CompetitionIterator** definește un iterator de persoane care participă la competiție, în ordinea categoriilor de vârstă ale acestora ($c_0, c_1, \dots, c_6$), iar în cadrul unei categorii de vârstă în ordine alfabetică a numelor.

Scrieți un program într-unul din limbajele de programare **C++**, **Java**, sau **C#**, cu următoarele cerințe:

- a) Declarați clasele **Competition**, **Person**, **Participant**, attributele și metodele acestora conform diagramei de mai sus. Implementați doar următoarele metode:
 - a1) Constructorul clasei **Competition**.
 - a2) Metoda **setNext(p: Participant)** din clasa **Participant**.
- b) Definiți clasa **CompetitionIterator**, incluzând attributele și metodele specifice unui iterator. Implementați toate metodele clasei **CompetitionIterator**. Atributele și metodele implementate vor folosi spațiu de memorie $\theta(1)$. Metodele vor avea complexitate timp $\theta(1)$.
- c) Definiți o funcție care primește ca parametru un obiect **c** de tip **Competition** și, folosind clasa definită la b) tipărește participanții la competiție, în ordinea categoriilor de vârstă ale acestora, iar în cadrul unei categorii de vârstă în ordine alfabetică a numelor.
- d) Definiți o funcție care primește ca parametri un obiect **c** de tip **Competition** și un șir de caractere *name* și returnează categoria de vârstă la care participă în competiție persoana având numele *name*, sau șirul de caractere vid în cazul în care nu există niciun participant având numele *name*. Dacă există mai mulți participanți cu același nume *name*, se va returna prima categorie de vârstă în care se găsește un participant cu numele dat.
- e) Construiți un obiect **o** de tip **Competition** în care adăugați 2 categorii de vârstă și 4 participanți (câte 2 în fiecare categorie). Apelați apoi funcția de la punctul c). Pentru una dintre persoanele participante apelați funcția de la d).

Problema 1. (4 puncte)

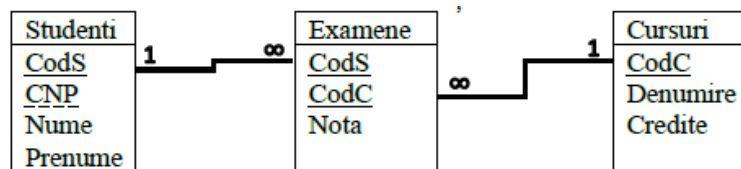
O companie media utilizează o bază de date relațională pentru a-și gestiona activitatea.

- Fiecare ziar are un cod unic de identificare, un nume (unic pentru fiecare ziar), un an de fondare și o frecvență de publicare (poate fi un atribut simplu, șir de caractere; de exemplu: *zilnic*, *săptămânal*, *lunar*); fiecare ziar este gestionat de un singur birou central, dar un birou central poate gestiona mai multe ziare.
- Un birou central are un cod de identificare și un număr de telefon (unic per birou central).
- Fiecare jurnalist are un cod de identificare, nume, prenume, data nașterii, gen și an de debut. Un jurnalist poate avea contracte de colaborare cu mai multe ziare. Un ziar poate colabora cu mai mulți jurnaliști. Un jurnalist poate avea un singur contract de colaborare cu un anumit ziar. Pentru fiecare contract de colaborare, se rețin: ziarul, jurnalistul, data de început a contractului, durata contractului (număr întreg reprezentând numărul de luni) și remunerația.
- Un articol are un cod de identificare, un titlu și este scris de un jurnalist pentru unul din ziarele cu care colaborează.
- O temă are un cod de identificare, o denumire (unică per temă; de exemplu, *Economie*, *Sport*) și o descriere. Un articol poate trata mai multe teme, iar o temă se poate regăsi în mai multe articole.

Realizați o schemă relațională BCNF pentru baza de date, evidențiind riguros cheile primare, cheile candidat și cheile externe. Realizați schema într-una din manierele indicate în exemplul de mai jos:

* exemplu pentru tabelele Studenți, Cursuri și Examene:

V1. Diagramă cu tabele, chei primare subliniate cu linie continuă, chei candidat subliniate cu linie întreruptă, legături trasate direct între cheile externe și cheile primare / candidat corespunzătoare (de exemplu, legătură trasată între coloana CodS din Examene și coloana CodS din Studenți).



V2.

Studenți[CodS, CNP, Nume, Prenume]

Cursuri[CodC, Denumire, Credite]

Examene[CodS, CodC, Nota]

Cheile primare sunt subliniate cu linie continuă, iar cheile candidat sunt subliniate cu linie întreruptă.

{CodS} este cheie externă în Examene și face referire la {CodS} din Studenți. {CodC} este cheie externă în Examene și face referire la {CodC} din Cursuri.

Problema 2. (5 puncte)

Se dau următoarele scheme relaționale:

CreatoriContinut[CodCreator, Nume, Tara, AnulNasterii]

Videoclipuri[CodVideo, Titlu, Categorie, Durata, *CodCreator*]

Platforme[CodPlatforma, Denumire]

VideoclipuriPlatforme[CodVideo, CodPlatforma, Monetizare, AnPublicare]

Cheile primare sunt subliniate. Cheile externe sunt scrise cursiv și au aceeași denumire cu coloanele la care fac referire.

a. Scrieți o interogare SQL care returnează codul și numele fiecărui creator de conținut care satisface următoarele două condiții:

1. a publicat cel puțin 2 videoclipuri în același an și
2. are cel puțin un videoclip publicat pe platforma “StreamNow”.

Un videoclip este *publicat* dacă apare pe o platformă.

Exemplu pentru condiția 1:

- creatorul C1 a publicat videoclipurile: V1 în 2020, V2 în 2020, V3 în 2021; C1 satisface condiția 1.
- creatorul C2 a publicat videoclipurile: V4 în 2020, V5 în 2021, V6 în 2022; C2 nu satisface condiția 1.

b. Se dau instanțele următoare pentru relațiile CreatoriContinut, Videoclipuri, Platforme și VideoclipuriPlatforme:

CreatoriContinut

Cod Creator	Nume	Tara	AnulNasterii
1	n1	Romania	2005
2	n2	Romania	2005
3	n3	Franta	2005

Platforme

Cod Platforma	Denumire
1	p1
2	p2
3	p3

Videoclipuri

Cod Video	Titlu	Categorie	Durata	Cod Creator
1	v1	c1	3	1
2	v2	c2	3	1
3	v3	c1	4	1
4	v4	c1	2	2
5	v5	c1	3	2
6	v6	c2	4	1
7	v7	c2	2	1

VideoclipuriPlatforme

Cod Video	Cod Platforma	Monetizare	An Publicare
1	2	T	2020
2	2	F	2020
3	3	T	2021
4	3	T	2020
5	3	T	2021
6	1	F	2020
7	1	F	2021

b1. Precizați rezultatul evaluării interogării de mai jos pe instanțele date. Menționați strict valorile tuplului / tuplurilor și denumirile coloanelor din rezultat, fără a prezenta toți pașii evaluării interogării.

```
SELECT vp.CodPlatforma, MIN(vp.AnPublicare) AnMin
FROM CreatoriContinut c INNER JOIN Videoclipuri v ON c.CodCreator = v.CodCreator
     INNER JOIN VideoclipuriPlatforme vp ON v.CodVideo = vp.CodVideo
WHERE c.Tara = 'Romania'
GROUP BY vp.CodPlatforma
HAVING COUNT(vp.CodVideo) >
  (SELECT COUNT(*)
   FROM VideoclipuriPlatforme vp2 INNER JOIN Platforme p2
    ON vp2.CodPlatforma = p2.CodPlatforma
   WHERE p2.Denumire = 'p2')
```

b2. Explicați dacă următoarele dependențe funcționale sunt satisfăcute sau nu de datele din instanța Videoclipuri:

- $\{Categorie\} \rightarrow \{Titlu\}$
- $\{CodCreator\} \rightarrow \{CodVideo\}$

SUBIECT Sisteme de operare

Problema 1. (5 puncte) Răspundeți la următoarele întrebări despre execuția programului `./a.out` compilat din codul sursă de mai jos, considerând că toate include-urile necesare sunt prezente și că apelurile sistem `fork` și `wait` se execută cu succes. Menționăm că macro-ul `WEXITSTATUS(st)` extrage codul de ieșire transmis de fiu prin `exit` (un întreg pe 8 biți, în intervalul 0–255).

<pre>1 int main(int argc, 2 char** argv){ 3 int i, st, total = 0; 4 for (i=1; i<argc; i++) { 5 if (fork() == 0) { 6 exit(strlen(argv[i])); 7 } 8 wait(&st); 9 total += WEXITSTATUS(st); 10 } 11 printf("%d\n", total); 12 return 0; 13 }</pre>	a) Ce va afișa execuția <code>./a.out ab cde f</code> ? Justificați răspunsul. b) Câte procese fiu creează <code>./a.out ab cde f</code> ? Dar <code>./a.out</code> (fără argumente), și ce afișează în acest caz? Justificați răspunsul. c) Explicați în detaliu funcționarea liniei 6 și cum ajunge valoarea transmisă de <code>exit</code> la procesul părinte? d) Explicați în detaliu funcționarea liniilor 8–9. e) Ce va afișa <code>./a.out</code> apelat cu un argument format din 260 de caractere? Justificați răspunsul.
--	--

Problema 2. (4 puncte) Răspundeți la următoarele întrebări despre execuția scriptului Shell UNIX `./a.sh` de mai jos, considerând fișierul `d.txt` dat. Notă: numerotarea liniilor NU aparține de conținutul fișierelor.

<pre>1 #!/bin/bash 2 while read L; do 3 N=0; 4 for X in `echo "\$L"   sed -E "s/(.)\\1 /g"`; do 5 N=`expr \$N + 1` 6 done 7 if [\$N -ge 4]; then 8 echo "\$L" sed -E "s/^(..)*\$/\\1/g" 9 else 10 echo "\$N" 11 fi 12 done < d.txt</pre>	<table><tr><th colspan="2">d.txt</th></tr><tr><td>1</td><td>hello</td></tr><tr><td>2</td><td>ab</td></tr><tr><td>3</td><td>sun</td></tr><tr><td>4</td><td>code</td></tr></table>	d.txt		1	hello	2	ab	3	sun	4	code
d.txt											
1	hello										
2	ab										
3	sun										
4	code										

- a) Specificați ce afișează scriptul pentru linia 1 respectiv 2 a fișierului `d.txt`. Justificați răspunsul.
- b) Explicați funcționarea comenzii `sed` din rândul 8.
- c) Specificați ce valori numerice se afișează. Justificați răspunsul.
- d) Explicați în detaliu funcționarea liniilor 3–6.

BAREM INFORMATICĂ

VARIANTA 2

Subiect Algoritmică și Programare

Oficiu – **1p**

Cerința 1. – **2.5p**

- semnatura – 0.2p
- implementare – **2.3p** din care
 - căutarea listei corespunzătoare vârstei *varsta* – 0.3p
 - adaugarea numelui *nume* în lista identificată – 2p

Cerința 2. – **1p**

- caz favorabil **0.2p** din care
 - complexitate – 0.1
 - justificare – 0.1
- caz mediu **0.4p** din care
 - complexitate – 0.2
 - justificare – 0.2
- caz defavorabil **0.4p** din care
 - complexitate – 0.2
 - justificare – 0.2

Cerința 3.a) – **1p**

- Definirea clasei Competition – **0.4p** din care
 - atribut – 0.1
 - constructor (a1) – 0.2p
 - metode **addParticipant**, **getNext** – 0.1
- Definirea clasei Participant – **0.3p** din care
 - atribute – 0.1
 - metode **getPerson**, **getNext** – 0.1
 - metoda setNext (a2) – 0.1p
- Definirea clasei Person – **0.3p** din care
 - atribute – 0.1
 - constructor – 0.1
 - metode **getName**, **getAge** – 0.1

Cerința 3.b) – **2.5p**

- Definirea clasei CompetitionIterator
 - atribute – 0.5
 - constructor – 0.5
 - metode specifice - 1.5

Funcția 3.c) – **0.7p**

- semnatura – **0.1p**
- parcurgere – **0.6p**

Funcția 3.d) – **1p**

- semnatura – **0.1p**
- implementare funcție – **0.8p**
- returnare rezultat – **0.1p**

Funcția principală 3.e) – **0.3p**

- construire obiect **o** – 0.1p
- adăugare 2 categorii de vârstă și 4 participanți – 0.1p
- apel funcții c) și d) – 0.1p

Problema 1. (4 puncte)

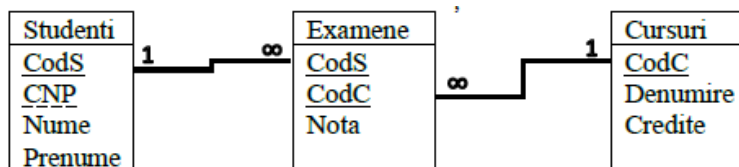
O companie media utilizează o bază de date relațională pentru a-și gestiona activitatea.

- Fiecare ziar are un cod unic de identificare, un nume (unic pentru fiecare ziar), un an de fondare și o frecvență de publicare (poate fi un atribut simplu, șir de caractere; de exemplu: *zilnic*, *săptămânal*, *lunar*); fiecare ziar este gestionat de un singur birou central, dar un birou central poate gestiona mai multe ziare.
- Un birou central are un cod de identificare și un număr de telefon (unic per birou central).
- Fiecare jurnalist are un cod de identificare, nume, prenume, data nașterii, gen și an de debut. Un jurnalist poate avea contracte de colaborare cu mai multe ziare. Un ziar poate colabora cu mai mulți jurnaliști. Un jurnalist poate avea un singur contract de colaborare cu un anumit ziar. Pentru fiecare contract de colaborare, se rețin: ziarul, jurnalistul, data de început a contractului, durata contractului (număr întreg reprezentând numărul de luni) și remunerația.
- Un articol are un cod de identificare, un titlu și este scris de un jurnalist pentru unul din ziarele cu care colaborează.
- O temă are un cod de identificare, o denumire (unică per temă; de exemplu, *Economie*, *Sport*) și o descriere. Un articol poate trata mai multe teme, iar o temă se poate regăsi în mai multe articole.

Realizați o schemă relațională BCNF pentru baza de date, evidențiind riguros cheile primare, cheile candidat și cheile externe. Realizați schema într-una din manierele indicate în exemplul de mai jos:

* exemplu pentru tabelele Studenți, Cursuri și Examene:

V1. Diagramă cu tabele, chei primare subliniate cu linie continuă, chei candidat subliniate cu linie întreruptă, legături trasate direct între cheile externe și cheile primare / candidat corespunzătoare (de exemplu, legătură trasată între coloana CodS din Examene și coloana CodS din Studenți).



V2.

Studenți[CodS, CNP, Nume, Prenume]

Cursuri[CodC, Denumire, Credite]

Examene[CodS, CodC, Nota]

Cheile primare sunt subliniate cu linie continuă, iar cheile candidat sunt subliniate cu linie întreruptă.

{CodS} este cheie externă în Examene și face referire la {CodS} din Studenți. {CodC} este cheie externă în Examene și face referire la {CodC} din Cursuri.

Problema 2. (5 puncte)

Se dau următoarele scheme relaționale:

CreatoriContinut[CodCreator, Nume, Tara, AnulNasterii]

Videoclipuri[CodVideo, Titlu, Categorie, Durata, *CodCreator*]

Platforme[CodPlatforma, Denumire]

VideoclipuriPlatforme[CodVideo, CodPlatforma, Monetizare, AnPublicare]

Cheile primare sunt subliniate. Cheile externe sunt scrise cursiv și au aceeași denumire cu coloanele la care fac referire.

a. Scrieți o interogare SQL care returnează codul și numele fiecărui creator de conținut care satisface următoarele două condiții:

1. a publicat cel puțin 2 videoclipuri în același an și
2. are cel puțin un videoclip publicat pe platforma “StreamNow”.

Un videoclip este *publicat* dacă apare pe o platformă.

Exemplu pentru condiția 1:

- creatorul C1 a publicat videoclipurile: V1 în 2020, V2 în 2020, V3 în 2021; C1 satisface condiția 1.
- creatorul C2 a publicat videoclipurile: V4 în 2020, V5 în 2021, V6 în 2022; C2 nu satisface condiția 1.

b. Se dau instanțele următoare pentru relațiile CreatoriContinut, Videoclipuri, Platforme și VideoclipuriPlatforme:

CreatoriContinut

Cod Creator	Nume	Tara	AnulNasterii
1	n1	Romania	2005
2	n2	Romania	2005
3	n3	Franta	2005

Platforme

Cod Platforma	Denumire
1	p1
2	p2
3	p3

Videoclipuri

Cod Video	Titlu	Categorie	Durata	Cod Creator
1	v1	c1	3	1
2	v2	c2	3	1
3	v3	c1	4	1
4	v4	c1	2	2
5	v5	c1	3	2
6	v6	c2	4	1
7	v7	c2	2	1

VideoclipuriPlatforme

Cod Video	Cod Platforma	Monetizare	An Publicare
1	2	T	2020
2	2	F	2020
3	3	T	2021
4	3	T	2020
5	3	T	2021
6	1	F	2020
7	1	F	2021

b1. Precizați rezultatul evaluării interogării de mai jos pe instanțele date. Menționați strict valorile tuplului / tuplurilor și denumirile coloanelor din rezultat, fără a prezenta toți pașii evaluării interogării.

```
SELECT vp.CodPlatforma, MIN(vp.AnPublicare) AnMin
FROM CreatoriContinut c INNER JOIN Videoclipuri v ON c.CodCreator = v.CodCreator
     INNER JOIN VideoclipuriPlatforme vp ON v.CodVideo = vp.CodVideo
WHERE c.Tara = 'Romania'
GROUP BY vp.CodPlatforma
HAVING COUNT(vp.CodVideo) >
  (SELECT COUNT(*)
   FROM VideoclipuriPlatforme vp2 INNER JOIN Platforme p2
    ON vp2.CodPlatforma = p2.CodPlatforma
   WHERE p2.Denumire = 'p2')
```

b2. Explicați dacă următoarele dependențe funcționale sunt satisfăcute sau nu de datele din instanța Videoclipuri:

- $\{Categorie\} \rightarrow \{Titlu\}$
- $\{CodCreator\} \rightarrow \{CodVideo\}$

BAREM INFORMATICA

VARIANTA 2

Sisteme de Operare

1p – oficiu

Problema 1 (5p)

1p – a) $6 = \text{strlen}(\text{"ab"}) + \text{strlen}(\text{"cde"}) + \text{strlen}(\text{"f"}) = 2 + 3 + 1$

1p – b) 3 fii pentru ./a.out ab cde f; 0 fii pentru ./a.out fără argumente, și se afișează 0.

1p – c) Linia 5 termină procesul fiu cu un cod de ieșire egal cu lungimea argumentului `argv[i]`; valoarea ajunge la părinte prin starea de terminare citită de `wait`.

1p – d) `wait(&st)` așteaptă terminarea fiului curent și pune starea lui în `st`; `WEXITSTATUS(st)` extrage codul de ieșire (lungimea), care se adună în `total`.

1p – e) Afișează 4; nu va afișa 260, deoarece codul de ieșire ocupă doar 8 biți (0–255);

Problema 2 (4p)

1p – a) (linia 1: `hello`) → `he`, (linia 2: `ab`) → `2. ...` + justificare

1p – b) Comanda `sed` înlocuiește fiecare linie (aici vom avea una singură) cu primele două caractere ale ei, ignorând restul textului.

1p – c) se va afișa 2 și 3 (pentru liniile 2 și 3 din `d.txt`)

1p – d) Liniile 3-6 calculează în `N` **lungimea** liniei curente