

Tema 1

George-Alexandru Tudor, Farhad Ali-Irinel Gul, Stavăr Laurențiu-Cristian,
Andrei-Laurențiu Radu, Sândulache Mihnea, Costin-Alexandru Deonise
Coordonator: Mihai Nan

Data postării: 21.11.2023

Deadline: 16.12.2023 ora 23:59

1 Problema 1

1.1 Descriere

Bingo este un joc de noroc popular care implică extragerea aleatorie a unor numere și marcarea acestora pe carduri speciale. Fiecare card conține 25 de numere, așezate sub forma unei matrice cu 5 linii și 5 coloane. Fiecare coloană este etichetată cu o literă și conține un interval de numere restricționat după cum urmează:

- coloana 1 – **B** (conține doar numere naturale din intervalul $[1, 15]$)
- coloana 2 – **I** (conține doar numere naturale din intervalul $[16, 30]$)
- coloana 3 – **N** (conține doar numere naturale din intervalul $[31, 45]$)
- coloana 4 – **G** (conține doar numere naturale din intervalul $[46, 60]$)
- coloana 6 – **O** (conține doar numere naturale din intervalul $[61, 75]$)

Organizatorul extrage, pe rând, câte o bilă. Fiecare bilă conține o literă specifică unei coloane de pe card și un număr corespunzător. Toți jucătorii care au pe card numărul extras, la coloana indicată, vor colora, pe card, celula respectivă. Fiecare linie, coloană sau diagonală colorată complet valorează câte 10 puncte. De asemenea, organizatorul va extrage un anumit număr de bile, ceea ce înseamnă că nu întotdeauna vom avea un câștigător.

Jocul este câștigat dacă un jucător reușește să coloreze toate celulele de pe o linie, de pe o coloană sau de pe o diagonală. Pentru a oferi un ajutor jucătorului, după primirea cardului, acesta trebuie să înlocuiască numărul din centrul cardului cu simbolul # (colorează elementul din mijloc). Jocul se oprește când un jucător a atins o combinație și a strigat „BINGO”.

Noi vom face jocul puțin mai interesant. La un moment dat, în timpul jocului, organizatorul poate extrage bile mai speciale:

- o bilă de tip **SWAP-X-Y** are ca efect interschimbarea coloanelor X și Y , unde X și Y vor fi numere ce reprezintă coloanele.
- o bilă de tip **SHIFT-X-Y** va produce mutarea ciclică în jos a numerelor de pe coloana X cu Y poziții în jos, pe aceeași coloană (unde X și Y vor fi numere).
- o bilă de tip **ASC-X** va ordona crescător numerele de pe coloana X (unde X este un număr ce reprezintă indexul coloanei).

- o bilă de tip **DSC-Y** va ordona descrescător numerele de pe coloana Y (unde Y este un număr ce reprezintă indexul coloanei).
- o bilă de tip **SHOW-I** va forța jucătorul I să își afișeze cardul (unde I este un număr).

Multă lume s-a înscris la jocul nostru și, pentru a valida cardurile câștigătoare, avem nevoie de ajutorul vostru. Veți primi N carduri pe care dorim să le verificați. Cardurile se află în starea inițială, înainte de începerea jocului.

1.2 Cerințe

Voi trebuie să implementați un program care să realizeze următoarele:

1. Citirea cardurilor de joc. Pentru a înțelege formatul în care vor fi furnizate aceste carduri, analizați secțiunea 1.4.

Important

După ce a fost citit un card, colorăm elementul aflat în mijlocul fiecărei matrice. Vom considera ca fiecare card va avea asociat un ID egal cu poziția sa în input. Considerăm indexarea cardurilor de la 0.

Observație

Pentru a putea stoca toate cardurile de joc, puteți folosi un tablou tridimensional. Spre exemplu, `int carduri[10][5][5]`; În acest caz, `carduri` poate fi privit drept un vector cu 10 elemente de tip matrice 5×5 . `carduri[0]` o să fie matricea în care stocăm numerele existente în cardul jucătorului cu **ID-ul 0**. `carduri[1][2][3]` o să fie numărul din cardul jucătorului 1 aflat pe linia 2 și coloana 3.

Hint! Puteți utiliza un tablou suplimentar în care să marcați celulele colorate.

2. Implementarea unei funcții pentru fiecare tip de operație:
 - (a) o funcție care interschimbă coloanele dintr-un card;
 - (b) o funcție care produce o mutare ciclică pe coloana X a cardului;
 - (c) o funcție care sortează crescător numerele de pe coloana X ;
 - (d) o funcție care sortează descrescător numerele de pe coloana Y ;
 - (e) o funcție care *colorează* o anumită celulă de pe card dacă aceasta conține valoarea indicată;
 - (f) o funcție care afișează conținutul unui card;
 - (g) o funcție care verifică dacă un card conține o combinație finală (conține cel puțin o linie, coloană sau diagonală colorată complet).

Important

Puteți alege ce antet doriți pentru aceste funcții și puteți adăuga funcții suplimentare dacă veți considera necesar, dar este obligatoriu să aveți definită cel puțin o funcție pentru fiecare operație.

3. Citirea și aplicarea comenzilor una câte una. Pentru a înțelege formatul în care vor fi furnizate aceste comenzi, analizați secțiunea 1.4.

Important

Dacă după aplicarea unei comenzi va exista un card ce conține o combinație finală:

- vom afișa indicele cardului câștigător;
- vom afișa cardul câștigător (celulele colorate vor fi marcate cu #);
- vom afișa numărul de puncte acumulate;
- vom termina execuția programului, ignorând restul comenzilor.

Dacă după aplicarea unei comenzi vom găsi mai multe carduri câștigătoare, îl vom desemna câștigător pe cel ce colectează mai multe puncte. În cazul în care există mai multe carduri ce au același punctaj maxim, alegem cardul cu ID-ul minim.

Dacă după aplicarea tuturor comenzilor niciun card nu a ieșit câștigător, vom afișa mesajul NO WINNER.

1.3 Restricții și precizări

- Vor exista două tipuri de teste:
 1. Teste cu un singur card.
 2. Teste cu mai multe carduri.

Observație

În cazul în care doriți să rezolvați problema doar pentru cazul cu un singur card, atunci acest lucru este posibil și puteți primi punctaj doar pentru prima categorie de teste (cele cu un singur card).

- Vor exista teste simple care vor verifica individual fiecare tip de bilă. Astfel, puteți obține punctaj parțial dacă alegeți să implementați doar o parte din funcționalități.

1.4 Date de intrare

- Prima linie conține N , numărul de carduri.
- Urmează N matrice de dimensiune 5×5 . Fiecare matrice conține numere de la 1 la 75, așa cum am descris mai sus.
- După fiecare matrice este o linie goală.
- Pe următoarea linie este numărul de bile care vor fi extrase de organizator, M .
- Urmează M linii, fiecare conținând un text de forma COLOANĂ-NUMĂR sau SPECIAL (bile speciale, așa cum am descris mai sus).

1.5 Date de ieşire

- Pentru fiecare bilă de tip **SHOW**, vom afişa cardul corespunzător jucătorului specificat. Celule colorate vor fi marcate cu # în output.
- Dacă există un câştigător:
 - vom afişa pe o linie nouă indicele cardului câştigător;
 - pe următoarele linii vom afişa cardul câştigătorului;
 - pe ultima linie vom afişa punctajul câştigătorului.
- Dacă niciun card nu a ieşit câştigător, se va afişa pe o linie nouă textul **NO WINNER**.

1.6 Exemple

1.6.1 Exemplul 1

INPUT

14 30 41 53 74
4 28 35 57 68
12 23 34 46 75
8 16 43 48 70
10 27 39 56 66

5

B-1

O-61

I-30

G-47

N-31

OUTPUT

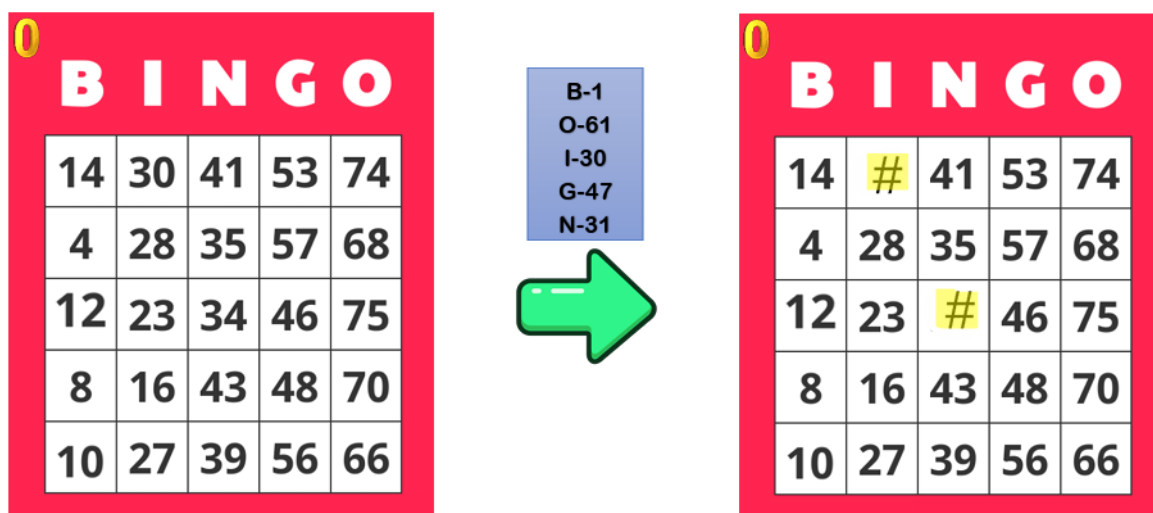
NO WINNER

Explicații

Primul lucru pe care va trebui să-l facem după ce am citit cardul jucătorului constă în colorarea celulei din mijloc.

Jucătorul cardului cu ID 0 nu a reuşit să coloreze complet nicio coloană, linie sau diagonală.

Fiind un singur card, necâştigător, s-a afişat mesajul **NO WINNER**.



1.6.2 Exemplul 2

INPUT

```
2
14 30 41 53 74
4 28 35 57 68
12 23 34 46 75
8 16 43 48 70
10 27 39 56 66

7 28 43 54 69
6 27 40 46 67
3 17 36 55 65
11 16 37 49 73
10 23 45 57 75

20
B-6
I-29
B-11
O-69
B-3
G-46
O-74
N-40
B-14
G-57
B-8
B-4
B-7
B-12
I-16
B-10
ASC-2
SHIFT-1-8
O-63
G-60
```

OUTPUT

```
0
# 30 41 53 #
# 28 35 # 68
# 23 # # 75
# # 43 48 70
# 27 39 56 66
20
```

Explicații

La fiecare operație se înlocuiește celula corespunzătoare bilei cu #, în toate cardurile în care se găsește.

Când se ajunge la bila B-10, se observă că ambele carduri devin câștigătoare.

Ambele carduri au acumulat 20 de puncte. În acest caz se va afișa cardul cu ID-ul cel mai mic dintre cele cu punctaj maxim, acela fiind cardul 0.

Execuția programului se termină, având în vedere că s-a declarat un câștigător. Prin urmare, toate operațiile după bila B-10 nu vor mai fi realizate.

0

B I N G O

14	30	41	53	74
4	28	35	57	68
12	23	34	46	75
8	16	43	48	70
10	27	39	56	66

B-6
I-29
B-11
O-69
B-3
G-46
O-74
N-40
B-14
G-57
B-8
B-4
B-7
B-12
I-16



0

B I N G O

#	30	41	53	#
#	28	35	#	68
#	23	#	#	75
#	#	43	48	70
10	27	39	56	66

1

B I N G O

7	28	43	54	69
6	27	40	46	67
3	17	36	55	65
11	16	37	49	73
10	23	45	57	75

1

B I N G O

#	28	43	54	#
#	27	#	#	67
#	17	#	55	65
#	#	37	49	73
10	23	45	#	75

0

B I N G O

#	30	41	53	#
#	28	35	#	68
#	23	#	#	75
#	#	43	48	70
10	27	39	56	66

B-10



0

B I N G O

#	30	41	53	#
#	28	35	#	68
#	23	#	#	75
#	#	43	48	70
#	27	39	56	66

1

B I N G O

#	28	43	54	#
#	27	#	#	67
#	17	#	55	65
#	#	37	49	73
10	23	45	#	75

1

B I N G O

#	28	43	54	#
#	27	#	#	67
#	17	#	55	65
#	#	37	49	73
#	23	45	#	75

1.6.3 Exemplul 3

INPUT

```
1
14 30 41 53 74
4 28 35 57 68
12 23 34 46 75
8 16 43 48 70
10 27 39 56 66
```

5

SWAP-1-2

SHIFT-4-3

DSC-1

ASC-0

SHOW-0

OUTPUT

```
4 43 30 53 75
8 41 28 57 70
10 39 23 46 66
12 35 16 48 70
14 # 27 56 66
NO WINNER
```

Explicații

Prima bilă extrasă presupune interschimbarea coloanelor 1 (I) și 2 (N). De asemenea, se ține cont de faptul că elementul din mijloc (34) este colorat de la început.

Bila SHIFT-4-3 va muta toate numerele de pe coloana 4 (O) cu 3 poziții în jos. Se observă că mutarea este ciclică, iar, de exemplu, elementul de pe linia 4, va ajunge pe linia 2.

La următoarele bile, se sortează descrescător coloana 1 (N) și crescător coloana 0 (B). Trebuie să fiți atenți la faptul că #-ul ține locul numărului 34, care este cel mai mic număr de pe coloană.

După afișarea cardului cu ID 0, jocul se termină (nu mai sunt alte bile) și niciun card nu a ieșit câștigător, afișându-se mesajul corespunzător – NO WINNER.

⁰ **B I N G O**

14	30	41	53	74
4	28	35	57	68
12	23	34	46	75
8	16	43	48	70
10	27	39	56	66

SWAP-1-2



⁰ **B N I G O**

14	41	30	53	74
4	35	28	57	68
12	#	23	46	75
8	43	16	48	70
10	39	27	56	66

⁰ **B N I G O**

14	41	30	53	74
4	35	28	57	68
12	#	23	46	75
8	43	16	48	70
10	39	27	56	66

SHIFT-4-3



⁰ **B N I G O**

14	41	30	53	75
4	35	28	57	70
12	#	23	46	66
8	43	16	48	74
10	39	27	56	68

⁰ **B N I G O**

14	41	30	53	75
4	35	28	57	70
12	#	23	46	66
8	43	16	48	74
10	39	27	56	68

DSC-1
ASC-0



⁰ **B N I G O**

4	43	30	53	75
8	41	28	57	70
10	39	23	46	66
12	35	16	48	74
14	#	27	56	68

2 Problema 2

2.1 Descriere

O bibliotecă are probleme cu gestionarea cărților și managerul a decis să vă ceară ajutorul pentru a implementa un sistem de gestiune care să permită găsirea celor mai potrivite cărți.

O carte conține următoarele detalii:

- ID carte – un număr natural;
- titlul cărții – șir de maximum 40 caractere;
- autorul cărții – șir de maximum 20 caractere;
- anul apariției – un număr natural;
- descrierea – un text pentru care trebuie să alocați dinamic **exact cât este nevoie**.

Observație

Vom considera că descrierea unei cărți este un text ce conține un număr de maximum 500 de caractere care se termină cu newline ('`\n`'). Pentru eficiență, în implementare, veți folosi alocare dinamică și veți aloca doar cât aveți nevoie.

Important

În implementare trebuie să folosiți o structură pentru reținerea informațiilor aferente unei cărți. Pentru a reține toate cărțile din bibliotecă, vom folosi un vector de structuri ce este alocat dinamic și realocat atunci când este cazul.

2.2 Cerințe

Bibliotecarii au nevoie de un program care să îi ajute să țină evidența cărților. Astfel, va trebui să îi ajutați prin implementarea unui program care să realizeze următoarele:

1. Citirea cărților care există în bibliotecă. Pentru a înțelege formatul în care vor fi furnizate aceste cărți, analizați secțiunea 2.3.
2. Implementarea unei funcții pentru fiecare dintre următoarele comenzi:
 - Comanda **SEARCH** **<ELEMENT>**, unde **<ELEMENT>** poate fi un număr sau un șir de caractere. Dacă argumentul comenzii **SEARCH** este un număr, atunci va trebui să găsim și să afișăm toate cărțile ce au anul apariției egal cu acel număr. Dacă argumentul comenzii **SEARCH** este un șir de caractere, atunci va trebui să găsim și să afișăm toate cărțile pentru care șirul de caractere furnizat reprezintă titlul cărții sau autorul cărții.

Important

Operația de căutare este **case-insensitive**, iar dacă avem mai multe cărți ce îndeplinesc condiția căutării, va trebui să le afișăm în ordinea crescătoare a ID-ului.

- Comanda `REPLACE <WORD1> <WORD2>`, unde `<WORD1>` și `<WORD2>` pot fi șiruri de caractere. Această comandă va căuta în toate descrierile cărților cuvântul `<WORD1>` și îi va înlocui toate aparițiile cu `<WORD2>`.

Important

Este garantat că `<WORD1>` și `<WORD2>` nu vor conține spații.

- Comanda `ENCODE <ID>`, unde `<ID>` este un număr natural reprezentând ID-ul unei cărți. Această comandă va codifica descrierea cărții prin aplicarea următorului algoritm:

Presupunem că descrierea este un șir de caractere `S` ce conține `N` caractere. Pentru fiecare cuvânt din acest șir de caractere, aplicăm următoarele operații:

- se înlocuiește fiecare subsir de caractere identice cu acest caracter urmat de numărul de apariții;
- numărul de apariții este convertit din baza 10 în baza 16;
- șirul obținut este inversat și afișat drept output la *ieșirea standard*.

Important

Semnele de punctuație din descriere vor fi ignorate. Considerăm cuvintele drept șiruri de caractere ce conțin doar litere din alfabetul englez despărțite prin spațiu. Dacă apar cifre în descriere acestea vor fi ignorate în codificare. Dacă nu există o carte cu ID-ul furnizat, atunci ignorăm comanda `ENCODE`.

- Comanda `ADD_BOOK` care o să fie urmată de 5 linii ce vor conține următoarele:
 - ID carte
 - titlul cărții
 - autorul cărții
 - anul apariției
 - descrierea

Important

Adăugarea se va face la finalul vectorului de cărți. Dacă nu mai aveți suficient spațiu în vectorul de cărți pentru noua carte atunci îl veți realoca dinamic.

3. Citirea și aplicarea de diverse comenzi furnizate ca intrare de la tastatură. Pentru a înțelege formatul în care vor fi furnizate aceste comenzi, analizați secțiunea 2.3.
4. Printarea informațiilor despre cărțile din bibliotecă obținute după aplicarea comenzilor anterioare. Consultați secțiunea 2.4 pentru a înțelege formatul în care va trebui să le afișați.

2.3 Date de intrare

- Prima linie conține `N`, numărul de cărți din bibliotecă.
- Pentru fiecare din cele `N` cărți vor exista câte 5 linii:

- o linie ce conține un număr natural ce reprezintă ID-ul cărții;
 - o linie ce conține un șir de maximum 20 de caractere ce reprezintă titlul cărții;
 - o linie ce conține un șir de maximum 20 de caractere ce reprezintă autorul cărții;
 - o linie ce conține un număr natural ce reprezintă anul de apariție;
 - o linie ce conține un șir de maximum 500 de caractere ce reprezintă descrierea cărții.
- După cele $N \times 5$ linii ce conțin informațiile despre cărți, urmează o linie pe care se află un număr natural M ce reprezintă numărul total de comenzi pe care le veți avea de executat.
 - În continuare, vor fi furnizate comenzile (fiecare comandă începe pe o linie nouă).

2.4 Date de ieșire

- Pentru fiecare comandă de tip **SEARCH** vom afișa toate cărțile care îndeplinesc condițiile de căutare. O carte o să fie afișată folosind același format utilizat la citire.
- Dacă pentru o comandă de tip **SEARCH** nu există un rezultat, vom afișa mesajul
NOT FOUND
- După fiecare afișare produsă de o comandă de tip **SEARCH** se va afișa un rând liber.
- După ce am executat toate comenzile, vom afișa pe un rând nou numărul total de cărți din bibliotecă după care pentru fiecare carte vom afișa următoarele:
 - o linie ce conține un număr natural ce reprezintă ID-ul cărții;
 - o linie ce conține un șir de maximum 20 de caractere ce reprezintă titlul cărții;
 - o linie ce conține un șir de maximum 20 de caractere ce reprezintă autorul cărții;
 - o linie ce conține un număr natural ce reprezintă anul de apariție;
 - o linie ce conține un șir de maximum 500 de caractere ce reprezintă descrierea cărții.

2.5 Exemple

2.6 Exemplul 1

INPUT	OUTPUT	Explicații
3	2	Se citesc datele a 3 cărți și
1	Carte2	4 comenzi.
Carte1	Autor2	
Autor1	2002	Comanda "SEARCH 2002" afișează
2001	Aceasta este cartea2	datele cărților care au anul
Aceasta este cartea1		apariției 2002, urmate de o
2	1	linie liberă.
Carte2	Carte1	
Autor2	Autor1	Comanda "SEARCH Autor1"
2002	2001	afișează datele cărților care
Aceasta este cartea2	Aceasta este cartea1	îl au ca autor sau ca titlu pe
3		Autor1, urmate de o linie
Carte3	3	liberă.
Autor3	Carte3	
2003	Autor3	Comanda "SEARCH Carte3"
Aceasta este cartea3	2003	îl au ca autor sau ca titlu pe
4	Aceasta este cartea3	Carte3, urmate de o linie
SEARCH 2002		liberă.
SEARCH Autor1	NOT FOUND	
SEARCH Carte3		Comanda "SEARCH Carte4" nu
SEARCH Carte4	3	găsește nicio carte care să îl
	1	aibă ca autor sau ca titlu pe
	Carte1	Carte4, așa că afișează NOT
	Autor1	FOUND, urmat de linie liberă.
	2001	
	Aceasta este cartea1	Urmează afișarea numărului de
	2	cărți și a datelor celor 3
	Carte2	cărți primite la început.
	Autor2	
	2002	
	Aceasta este cartea2	
	3	
	Carte3	
	Autor3	
	2003	
	Aceasta este cartea3	

2.7 Exemplul 2

INPUT	OUTPUT	Explicații
3	3	Se citesc datele a 3 cărți și o comandă.
1	1	
Carte1	Carte1	Comanda "REPLACE este are" înlocuiește fiecare apariție din descrierea cărților a cuvântului "este" cu "are". De exemplu, descrierea "Aceasta este cartea1" devine "Aceasta are cartea1".
Autor1	Autor1	
2001	2001	
Aceasta este cartea1	Aceasta are cartea1	
2	2	Urmează afișarea datelor celor 3 cărți primite la început, după modificarea descrierilor.
Carte2	Carte2	
Autor2	Autor2	
2002	2002	
Aceasta este cartea2	Aceasta are cartea2	
3	3	
Carte3	Carte3	
Autor3	Autor3	
2003	2003	
Aceasta este cartea3	Aceasta are cartea3	
1		
REPLACE este are		

2.8 Exemplul 3

INPUT	OUTPUT
3	3
1	1
Carte1	Carte1
Autor1	Autor1
2001	2001
Aceasta este cartea1	Aceasta este cartea1
2	2
Carte2	Carte2
Autor2	Autor2
2002	2002
Aceasta este cartea2	21a1e1t1r1a1c 1e1t1s1e 1a1t1s1a1e1c1A
3	3
Carte3	Carte3
Autor3	Autor3
2003	2003
Aceasta este cartea3	Aceasta este cartea3
2	
ENCODE 2	
ENCODE 4	

Explicații

Se citesc datele a 3 cărți și 2 comenzi.

Pentru prima comandă, se va înlocui descrierea cărții cu ID 2, cu un șir de caractere format astfel:

1. La fiecare caracter se adaugă numărul de apariții consecutive:

Aceasta este cartea2 -> A1c1e1a1s1t1a1 e1s1t1e1 c1a1r1t1e1a12

Se observă că spațiile și cifrele sunt ignorate în codificare

2. Se transformă numărul de apariții din baza 10 în baza 16. În exemplul curent, toate caracterele apar o singură dată consecutiv, astfel $1_{10} \rightarrow 1_{16}$.

3. Se inversează șirul obținut.

La comanda "ENCODE 4" nu se produce nimic deoarece nu există o carte cu ID-ul 4

2.9 Exemplul 4

INPUT	OUTPUT
3	2
1	Carte2
Carte1	Autor2
Autor1	2002
2001	Aceasta este cartea2
Aceasta este cartea1	
2	2
Carte2	Carte2
Autor2	Autor2
2002	2002
Aceasta este cartea2	21a1e1t1r1a1c 1e1t1s1e 1a1t1s1a1e1c1A
3	
Carte3	4
Autor3	Carte4
2003	Autor2
Aceasta este cartea3	2004
5	Aceasta este cartea4
SEARCH 2002	
ENCODE 2	4
REPLACE este are	1
ADD_BOOK	Carte1
4	Autor1
Carte4	2001
Autor2	Aceasta are cartea1
2004	2
Aceasta este cartea4	Carte2
SEARCH Autor2	Autor2
	2002
	21a1e1t1r1a1c 1e1t1s1e 1a1t1s1a1e1c1A
	3
	Carte3
	Autor3
	2003
	Aceasta are cartea3
	4
	Carte4
	Autor2
	2004
	Aceasta este cartea4

Explicații

Comanda "SEARCH 2002" afișează datele cărților care au anul apariției 2002, urmate de o linie liberă.

Comanda "ENCODE 2" transformă descrierile cărților cu id-ul 2 conform algoritmului.

Comanda "REPLACE este are" înlocuiește fiecare apariție din descrierea cărților a cuvântului "este" cu "are".

Comanda "ADD_BOOK", urmată de 5 linii (reprezentând datele cărții), adaugă în listă nouă carte.

Comanda "SEARCH Autor2" afișează datele cărților care au anul apariției 2002, urmate de o linie liberă.

Se afișează numărul de cărți și nouă lista de cărți (cele 3 primite la început și cea adăugată ulterior).

2.10 Exemplul 5

INPUT	OUTPUT	Explicații
3	4	Se citesc datele a 3 cărți și 1 comandă.
1	1	
Carte1	Carte1	Comanda "ADD_BOOK", urmată de 5 linii, (reprezentând datele cărții), adaugă în listă nouă carte.
Autor1	Autor1	
2001	2001	
Aceasta este cartea1	Aceasta este cartea1	
2	2	
Carte2	Carte2	În listă sunt acum 4 cărți.
Autor2	Autor2	
2002	2002	
Aceasta este cartea2	Aceasta este cartea2	Se afișează numărul de cărți și noua listă de cărți (care conține acum și noua carte).
3	3	
Carte3	Carte3	
Autor3	Autor3	
2003	2003	
Aceasta este cartea3	Aceasta este cartea3	
1	4	
ADD_BOOK	Carte4	
4	Autor4	
Carte4	2004	
Autor4	Aceasta este cartea4	
2004		
Aceasta este cartea4		

3 Precizări

- Separați logica programelor în mai multe funcții.
- **NU** se vor puncta sursele în care tot programul este scris în `main`!
- Este interzisă folosirea variabilelor globale!
- Soluția temei va fi scrisă în ANSI C! Nu folosiți sintaxă sau instrucțiuni specifice limbajului C++.
- Fiecare problemă va fi scrisă într-un fișier sursă separat, numit așa cum s-a precizat!
- În README precizați cât timp v-a luat implementarea cerințelor și explicați, pe scurt în câteva fraze, implementarea temei (comentariile din cod vor documenta mai amănunțit rezolvarea).
- Este recomandat ca liniile de cod și cele din fișierul README să nu depășească 80 de caractere.
- Temele sunt strict individuale. Copierea temelor va fi sancționată cu punctaj 0 pentru toți cei care au porțiuni de cod identice!
- Persoanele cu porțiuni de cod identice **NU** vor primi niciun punctaj pe temă.
- **NU** copiați cod de pe Internet! Se poate ajunge la situația în care doi studenți să aibă același cod, preluat de pe Internet, caz în care ambele teme vor fi punctate în TOTALITATE cu 0, deși studenții nu au colaborat direct între ei.
- **NU** folosiți tool-uri care generează automat cod pentru rezolvarea temei. Temele vor fi comparate cu soluțiile generate de astfel de tool-uri.
- Temele trimise după deadline **NU** vor fi luate în considerare.
- Datele de intrare sunt descrise pentru fiecare cerință individual și vor fi citite de la tastatură.
- Toate citirile se fac de la tastatură în codul final al temei! Folosiți fișiere sau/și redirectarea intrării/ieșirii doar pentru testele voastre intermediare!

Operatorul `<` poate fi folosit pentru a redirecționa conținutul unui fișier către intrarea standard a unui program. Spre exemplu, considerăm că avem executabilul `bingo` (care în mod normal citește de la tastatură) și un fișier numit `input.txt`. Putem utiliza următoarea comandă pentru a redirecționa conținutul fișierului `input.txt` către intrarea executabilului `bingo`:

```
./bingo < input.txt
```

De asemenea, putem redirecționa intrarea unui program și să salvăm ieșirea într-un fișier folosind operatorii `<` și `>` în același timp:

```
./bingo < input.txt > output.txt
```

4 Coding Style

Folosiți un coding style astfel încât codul să fie ușor de citit și înțeles. De exemplu:

- Dați nume corespunzătoare variabilelor și funcțiilor.
- Nu adăugați prea multe linii libere sau alte spații goale unde nu este necesar:
 - nu terminați liniile în spații libere, trailing whitespaces;
 - nu adăugați prea multe linii libere între instrucțiuni sau la sfârșitul fișierului.
- Principalul scop al spațiilor este indentarea.
- Fiți consecvenți în coding style-ul ales.
- Vă recomandăm să parcurgeți această resursă: Coding style
- Există programe sau extensii pentru editoare text care vă pot formata codul. Este permisă utilizarea lor.
- Deși vă pot ajuta destul de mult, ar fi ideal să încercați să respectați coding style-ul pe măsură ce scrieți codul.

5 Trimiterea temei

Veți trimite o arhivă **ZIP** cu numele **EXACT** acesta: GRUPA_Nume_Prenume_Tema1.zip

De exemplu: 311CC_Popescu_Maria_Tema1.zip

Arhiva va conține următoarele fișiere:

1. README
2. Makefile
3. bingo.c
4. biblioteca.c

Atenție!

Veți fi depunctați complet pentru formatarea incorectă a arhivei (alt nume, alt tip, alte nume pentru fișiere, alte fișiere în plus sau în minus etc.) - **0 puncte pe temă.**

- Pentru întrebări legate de temă se va folosi în mod exclusiv **FORUM-ul** temei, pe care vă recomandăm să îl vizitați chiar și dacă nu aveți întrebări, întrucât este posibil să aflați informații noi din întrebările puse de colegii voștri, respectiv din răspunsurile date de noi.
- Compilarea nu ar trebui să producă warning-uri (verificați prin adăugarea flagului -Wall la gcc).
- Temele trebuie să fie încărcate pe **vmchecker**. NU se acceptă teme trimise pe e-mail sau altfel decât prin intermediul **vmchecker-ului**.
- Link vmchecker: <https://vmchecker.cs.pub.ro/ui/>

6 Punctaj

O temă perfectă valorează **125 de puncte**. Vor exista atât teste simple, care verifică o operație specifică, dar și teste complexe, în care există majoritatea operațiilor.

Punctajul pe probleme este următorul:

Cerința	Punctaj
Problema 1	60 puncte
Problema 2	65 puncte

Atenție!

Depunctările care se pot aplica:

- **-10 puncte** pentru lipsa explicațiilor din README
- **-10 puncte** pentru lipsă comentarii din cod (atenție! nu trebuie comentată fiecare linie, doar ceea ce este esențial pentru a putea fi ușor de înțeles rezolvarea – detalii la curs!)
- **-20 puncte** pentru Coding Style necorespunzător
- **Depunctare totală pentru formatarea incorectă a arhivei** (alt nume, alt tip, alte nume pentru fișiere, alte fișiere în plus sau în minus) - **0 puncte pe temă**

O temă care NU compilează va fi punctată cu 0.

Vor exista mai multe teste pentru fiecare problemă în parte. Punctele pe teste sunt independente, punctajul pe un anumit test nefiind condiționat de alte teste.

În fișierul README va trebui să descrieți pe scurt soluția pe care ați ales-o pentru fiecare problemă și alte lucruri pe care le considerați utile de menționat

7 Checker

- Arhiva se va trimite pe vmchecker, unde tema se va testa automat.
- Pentru testarea locală, aveți disponibil un set de teste și un checker local.
- Punctajul acordat pe rularea testelor este cel de pe vmchecker.
- Corectorii își rezervă dreptul de a scădea puncte pentru orice problemă găsită în implementare, dacă vor considera acest lucru necesar.