

2. Fiind date tablourile unidimensionale $A=(9,12,5,10,4)$ și $B=(31,45,19,58,11,75,20,39,40)$, indexate de la 0, scrieți indicii elementelor din tabloul **B** care depășesc suma tuturor elementelor din tabloul **A**. (6p.)
3. Variabila **pret** memorează prețul unui abonament telefonic exprimat în lei (număr natural), iar variabila **cod** litera corespunzătoare tipului acestuia: litera **B** pentru abonament Basic, **s** pentru abonament Standard și **P** pentru abonament Premium. Declarați cele două variabile și scrieți o secvență de instrucțiuni C/C++ în urma executării căreia să se afișeze pe prima linie a ecranului cuvântul "abonament:", urmat de tipul abonamentului, iar pe a doua linie a ecranului prețul abonamentului urmat de cuvântul "lei" și precedat de cuvântul "pret:". Cuvintele sunt separate prin spații, ca în exemplu.
Exemplu: dacă variabila **cod** memorează litera **s**, iar variabila **pret** memorează 52, se afișează pe ecran:
 abonament: Standard
 pret: 52 lei (6p.)

SUBIECTUL al III-lea

(30 de puncte)

1. Se citește un număr natural n ($n \in [10, 10^9]$). Scrieți, în pseudocod, un algoritm care înlocuiește în n cifrele prime ale numărului n cu cifra 1 și scrie numărul obținut. Dacă numărul nu are cifre prime, se scrie numărul inițial.
Exemplu: dacă $n=124390$, se scrie 114190, iar dacă $n=4698$, se scrie 4698. (10p.)
2. Primăria unui oraș a lansat un program amplu de reamenajare urbană. În prima etapă sunt analizate străzile unui cartier în care fiecare casă are un număr de adresă. Urbaniștii primăriei au stabilit o regulă pentru a identifica casele cu prioritate ridicată la reamenajare, numite "cifric-dominante". Un număr de casă este considerat cifric-dominant dacă cifra zecilor este strict mai mare decât suma tuturor celorlalte cifre ale numărului.
Pentru a marca în baza de date urbanistică casele prioritare, acestea primesc un cod de reamenajare egal cu dublul numărului casei +1. Casele care nu sunt "cifric-dominante" rămân în sistem exact cu numărul lor original, deoarece nu sunt prioritare.
Scrieți un program C/C++ care citește de la tastatură un număr natural n din intervalul $[2, 10^2]$, reprezentând numărul caselor din cartier, apoi cele n elemente ale unui tablou unidimensional, numere naturale din intervalul $[10, 10^3]$, reprezentând numerele caselor. Programul modifică apoi tabloul în memorie înlocuind numerele caselor "cifric-dominante" cu codurile sale de reamenajare ($2 \cdot \text{număr_casă} + 1$) și afișează pe ecran tabloul obținut, cu elementele separate prin câte un spațiu.
Exemplu: pentru $n=6$ și tabloul (74, 190, 715, 26, 162, 954), se obține tabloul (149, 381, 715, 26, 325, 954) (pentru numărul 74, $7 > 4$, pentru numărul 190, $9 > 1+0$, pentru numărul 162, $6 > 1+2$). (10p.)
3. În inima Arhivei Cronos, timpul nu curgea pur și simplu; el era înregistrat, catalogat și păstrat în două mari Cărți Eterne: Volumul **A** și Volumul **B**. Volumul **A** conținea toate evenimentele majore din istorie, ordonate cu mialgă de la cel mai vechi la cel mai recent. Volumul **B** conținea toate evenimentele minore, dar esențiale (descoperiri uitate, întâlniri tănuite), sortate și ele perfect, de la zorii lumii până la clipa prezentă. Veghetorii Timpului, Maestrul **K** și ucenicul său, Elara, primeau o misiune critică: trebuiau să creeze Cartea Finală a Destinului, un singur Volum **C**, care să cuprindă toată istoria, mare și mică, dar care să mențină o ordine cronologică absolută. Ajută-l pe Elara să realizeze Cartea Finală a Destinului perfect ordonată și fără a risipi o singură secundă de energie.
Fișierul text **bac1.txt** conține cel mult 10^6 numere naturale de maxim nouă cifre reprezentând evenimentele majore din Volumul **A**, iar fișierul text **bac2.txt** conține cel mult 10^6 numere naturale de maxim nouă cifre reprezentând evenimentele minore din Volumul **B**. Se cere să se afișeze pe ecran, separate printr-un spațiu, conținutul Volumului **C**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.
Exemplu: Dacă volumul **A** (Majore) conține evenimentele (1935, 1945, 1980, 1996, 2008), iar volumul **B** (Minore) conține evenimentele (1920, 1935, 1955, 1977) atunci Marea Carte finală a destinului va conține evenimentele (1920, 1935, 1935, 1945, 1955, 1977, 1980, 1996, 2008).
a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)

Examenul național de bacalaureat 2026
Proba E.d)
INFORMATICĂ

BAREM DE EVALUARE ȘI DE NOTARE

Varianta 1

Filieră teoretică, profil real, specializare științe ale naturii

- Se punctează orice modalitate de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit în barem. Nu se acordă fracțiuni de punct. Se acordă zece puncte din oficiu. Nota finală se calculează prin împărțirea la zece a punctajului total acordat pentru lucrare.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț (de exemplu tipuri întregi cu semn pentru memorarea numerelor naturale, dimensiune a tablourilor) este acceptată din punctul de vedere al corectitudinii programului, dacă acest lucru nu afectează funcționarea acestuia.

SUBIECTUL I

(20 de puncte)

1b 2d 3c 4b 5c	5x4p.
----------------	-------

SUBIECTUL al II - lea

(40 de puncte)

1.	a) Răspuns corect:	6p.	6235 5623 3562 2356
	b) Pentru răspuns corect	6p.	1002
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiuni repetitive(*) -instrucțiune de decizie -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 4p. 1p. 1p. 1p.	(*) Se acordă numai 2p. dacă cel puțin una dintre instrucțiunile repetitive este corectă, dar nu toate instrucțiunile repetitive sunt corecte.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat.
2.	Răspuns corect: 1, 3, 5	6p.	Se acordă numai 2p. dacă doar o valoare este conform cerinței, numai 4p. dacă doar două valori sunt conform cerinței și numai 5p. dacă, în plus față de cele trei valori conform cerinței, s-a precizat încă o valoare.
3.	Răspuns corect: -declarare a variabilelor conform cerinței -afișare a datelor în formatul cerut (*) -corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*) Se acordă 2p. pentru utilizarea constantelor de tip char pentru identificarea unui abonament Basic/Standard/Premium (se acordă doar 1p., dacă sunt identificate două abonamente). Se acordă câte 1p. pentru fiecare aspect (corespondență cuvinte-valori afișate, spații și linii separate) conform cerinței.

SUBIECTUL al III - lea

(30 de puncte)

1.	Pentru algoritm corect -citire a datelor, conform cerinței -determinare a valorii cerute (*) -scriere a datelor, conform cerinței -scriere principal corectă a structurilor de control, corectitudine globală a algoritmului ¹⁾ (**)	10p. 1p. 6p. 1p. 2p.	(*) Se acordă câte 2p. pentru fiecare aspect specific (identificarea cifrelor prime, cifre suport în cadrul numărului, algoritm de bază pentru înlocuirea unor cifre) conform cerinței. (**) Se punctează orice formă explicită de structură repetitivă sau decizională.
2.	Pentru program corect -declarare a unei variabile care să memoreze un tablou unidimensional, conform cerinței -citire a datelor, conform cerinței -transformare a tabloului conform cerinței (*) -afișare a datelor, conform cerinței -declarare a variabilelor simple, corectitudine globală a programului ¹⁾	10p. 1p. 1p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (algoritm de bază pentru determinarea sumei cifrelor unui număr, eliminarea cifrei zecilor din suma cifrelor, identificare a unui număr cu cifra zecilor mai mare decât suma celorlalte cifre, valori suport verificate pentru determinarea numerelor cifric-dominante, elemente suport înlocuite/elemente suport nemodificate, modificare în memorie) conform cerinței.
3.	a) Pentru răspuns corect -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență b) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorii cerute (*),(**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, afișare a datelor, corectitudine globală a programului ¹⁾	2p. 1p. 1p. 8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principal corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. Se folosește algoritmul de interclasare care este eficient din punct de vedere al timpului de execuție și al memoriei. Cele două fișiere se parcurg concomitent, se afișează valoarea mai mică dintre cele două elemente curente și se avansează numai în fișierul din care s-a afișat valoarea. Când parcurgerea unui fișier se încheie, toate elementele celui alt fișier neparcurs încă se vor afișa, ele fiind în ordine crescătoare. Complexitatea algoritmului de mai sus este liniară.

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.