

2. Utilizând metoda backtracking, se generează toate buchetele formate din câte trei flori distincte din mulțimea ordonată astfel: {trandafir, crin, gerbera, iris, eustoma, orhidee, zambilă}; în același buchet crinul nu poate fi împreună cu eustoma sau cu zambila. Două buchete diferă prin cel puțin o floare. Primele cinci soluții sunt, în ordinea generării: (trandafir, crin, gerbera), (trandafir, crin, iris), (trandafir, crin, orhidee), (trandafir, gerbera, iris), (trandafir, gerbera, eustoma). Scrieți două soluții, una generată imediat înainte, iar cealaltă generată imediat după (crin, gerbera, orhidee). (6p.)
3. Gama muzicală conține, în această ordine, notele **do, re, mi, fa, sol, la, si**. Două note formează un interval muzical, iar acesta este numit **terță** dacă, în ordinea din gamă, între cele două note există o singură altă notă muzicală. În notația englezescă se folosesc unele litere mari ale alfabetului pentru notele muzicale, astfel: litera **A** pentru nota **la**, litera **B** pentru nota **si**, apoi, în ordine, litera **C** pentru nota **do**, litera **D** pentru nota **re**, litera **E** pentru nota **mi**, litera **F** pentru nota **fa** și litera **G** pentru nota **sol**. Variabila **m**, declarată alăturat, memorează literele corespunzătoare notației englezești pentru două note care formează un interval muzical, date în ordinea din gamă. Știind că în câmpul **nota1** se memorează una dintre literele **C, D, E, F** sau **G**, scrieți o secvență de instrucțiuni C/C++ în urma executării căreia să se afișeze pe ecran mesajul **TERTA**, dacă intervalul respectiv formează o terță, sau mesajul **NU**, în caz contrar.
- ```
struct interval
{ char nota1;
 char nota2;
}m;
```
- Exemplu:** dacă în câmpul **nota1** se memorează litera **G** (pentru nota **sol**), iar în câmpul **nota2** se memorează litera **B** (pentru nota **si**), se afișează mesajul **TERTA**. (6p.)

### SUBIECTUL al III-lea

(30 de puncte)

1. Un număr natural nenul, **n**, se numește **moderat** dacă este egal cu produsul a două numere prime, iar acestea sunt consecutive în șirul numerelor prime (2, 3, 5, 7, 11, 13, 17...). Subprogramul **moderat** are un singur parametru, **n**, prin care primește un număr natural ( $n \in [1, 10^5]$ ). Subprogramul returnează valoarea 1, dacă **n** este un număr moderat, sau valoarea 0, în caz contrar. Scrieți definiția completă a subprogramului.
- Exemplu:** dacă **n=35**, subprogramul returnează 1 ( $35=5 \cdot 7$ ), iar dacă **n=18** sau **n=55** sau **n=4**, subprogramul returnează 0. (10p.)
2. O grădină este împărțită în parcele pătrate identice, dispuse pe rânduri succesive. Toate parcelele de pe un rând conțin același tip de plante (doar flori sau doar gazon) și oricare două rânduri alăturate conțin tipuri diferite de plante, **pe primul rând fiind gazon**. Pentru fiecare parcelă se memorează înălțimea acesteia, egală cu înălțimea medie a plantelor conținute, exprimată în decimetri. Pentru a pune în valoare florile, fiecare parcelă cu flori trebuie să fie mai înaltă cu cel puțin un decimetru decât oricare dintre parcelele cu gazon de pe rândul vecin anterior. S-a hotărât tunderea parcelelor cu gazon prea înalte, astfel încât ele să fie aduse la înălțimea maximă permisă. Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul  $[2, 10^2]$ , **m** și **n**, apoi elementele unui tablou bidimensional cu **m** linii și **n** coloane, numere naturale din intervalul  $[2, 10]$ , reprezentând înălțimile parcelelor, în ordinea dispunerii acestora în grădină, rând după rând, și pe fiecare rând de la stânga la dreapta. Programul modifică apoi tabloul în memorie corespunzător situației terenului după tunderea gazonului și afișează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele de pe aceeași linie fiind separate prin câte un spațiu.
- Exemplu:**  
pentru **m=4, n=6** și tabloul
- |   |   |   |   |   |   |
|---|---|---|---|---|---|
| 6 | 2 | 3 | 7 | 9 | 4 |
| 7 | 8 | 5 | 9 | 8 | 5 |
| 8 | 7 | 2 | 6 | 2 | 9 |
| 7 | 2 | 5 | 9 | 3 | 8 |
- se obține tabloul
- |   |   |   |   |   |   |
|---|---|---|---|---|---|
| 4 | 2 | 3 | 4 | 4 | 4 |
| 7 | 8 | 5 | 9 | 8 | 5 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 2 | 5 | 9 | 3 | 8 |
- (10p.)
3. Șirul 0, 0, 1, 4, 13, 38, 105, 280, 729 . . . este definit astfel:  $f_1=f_2=0, f_3=1, f_n=4 \cdot f_{n-1}-3 \cdot f_{n-2}-2 \cdot f_{n-3}$  (unde **n** este un număr natural  $n \geq 4$ ). Se citesc de la tastatură trei numere naturale **x, y** și **z** ( $x \leq y < z \leq 10^9$ ), valorile a trei termeni aflați pe poziții consecutive în șirul dat, și se cere să se scrie în fișierul **bac.txt**, în ordine descrescătoare, separați prin câte un spațiu, toți termenii șirului care sunt mai mici sau egali cu **z**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
- Exemplu:** dacă se citesc numerele 38 105 280 fișierul conține numerele 280 105 38 13 4 1 0 0
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)  
b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)



2. Utilizând metoda backtracking, se generează toate buchetele formate din câte trei flori distincte din mulțimea ordonată astfel: {trandafir, crin, gerbera, iris, eustoma, orhidee, zambilă}; în același buchet crinul nu poate fi împreună cu eustoma sau cu zambila. Două buchete diferă prin cel puțin o floare. Primele cinci soluții sunt, în ordinea generării: (trandafir, crin, gerbera), (trandafir, crin, iris), (trandafir, crin, orhidee), (trandafir, gerbera, iris), (trandafir, gerbera, eustoma). Scrieți două soluții, una generată imediat înainte, iar cealaltă generată imediat după (crin, gerbera, orhidee). (6p.)
3. Gama muzicală conține, în această ordine, notele **do, re, mi, fa, sol, la, si**. Două note formează un interval muzical, iar acesta este numit **terță** dacă, în ordinea din gamă, între cele două note există o singură altă notă muzicală. În notația englezescă se folosesc unele litere mari ale alfabetului pentru notele muzicale, astfel: litera **A** pentru nota **la**, litera **B** pentru nota **si**, apoi, în ordine, litera **C** pentru nota **do**, litera **D** pentru nota **re**, litera **E** pentru nota **mi**, litera **F** pentru nota **fa** și litera **G** pentru nota **sol**. Variabila **m**, declarată alăturat, memorează literele corespunzătoare notației englezești pentru două note care formează un interval muzical, date în ordinea din gamă. Știind că în câmpul **nota1** se memorează una dintre literele **C, D, E, F** sau **G**, scrieți o secvență de instrucțiuni Pascal în urma executării căreia să se afișeze pe ecran mesajul **TERTA**, dacă intervalul respectiv formează o terță, sau mesajul **NU**, în caz contrar.
- ```
type interval=record
    nota1, nota2:char
end;
var m: interval;
```
- Exemplu:** dacă în câmpul **nota1** se memorează litera **G** (pentru nota **sol**), iar în câmpul **nota2** se memorează litera **B** (pentru nota **si**), se afișează mesajul **TERTA**. (6p.)

SUBIECTUL al III-lea

(30 de puncte)

1. Un număr natural nenul, **n**, se numește **moderat** dacă este egal cu produsul a două numere prime, iar acestea sunt consecutive în șirul numerelor prime (2, 3, 5, 7, 11, 13, 17...). Subprogramul **moderat** are un singur parametru, **n**, prin care primește un număr natural ($n \in [1, 10^5]$). Subprogramul returnează valoarea 1, dacă **n** este un număr moderat, sau valoarea 0, în caz contrar. Scrieți definiția completă a subprogramului.
- Exemplu:** dacă **n=35**, subprogramul returnează 1 ($35=5 \cdot 7$), iar dacă **n=18** sau **n=55** sau **n=4**, subprogramul returnează 0. (10p.)
2. O grădină este împărțită în parcele pătrate identice, dispuse pe rânduri succesive. Toate parcelele de pe un rând conțin același tip de plante (doar flori sau doar gazon) și oricare două rânduri alăturate conțin tipuri diferite de plante, **pe primul rând fiind gazon**. Pentru fiecare parcelă se memorează înălțimea acesteia, egală cu înălțimea medie a plantelor conținute, exprimată în decimetri. Pentru a pune în valoare florile, fiecare parcelă cu flori trebuie să fie mai înaltă cu cel puțin un decimetru decât oricare dintre parcelele cu gazon de pe rândul vecin anterior. S-a hotărât tunderea parcelelor cu gazon prea înalte, astfel încât ele să fie aduse la înălțimea maximă permisă. Scrieți un program Pascal care citește de la tastatură două numere naturale din intervalul $[2, 10^2]$, **m** și **n**, apoi elementele unui tablou bidimensional cu **m** linii și **n** coloane, numere naturale din intervalul $[2, 10]$, reprezentând înălțimile parcelelor, în ordinea dispunerii acestora în grădină, rând după rând, și pe fiecare rând de la stânga la dreapta. Programul modifică apoi tabloul în memorie corespunzător situației terenului după tunderea gazonului și afișează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele de pe aceeași linie fiind separate prin câte un spațiu.
- Exemplu:** pentru **m=4**, **n=6** și tabloul
- | | | | | | |
|---|---|---|---|---|---|
| 6 | 2 | 3 | 7 | 9 | 4 |
| 7 | 8 | 5 | 9 | 8 | 5 |
| 8 | 7 | 2 | 6 | 2 | 9 |
| 7 | 2 | 5 | 9 | 3 | 8 |
- se obține tabloul
- | | | | | | |
|---|---|---|---|---|---|
| 4 | 2 | 3 | 4 | 4 | 4 |
| 7 | 8 | 5 | 9 | 8 | 5 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 2 | 5 | 9 | 3 | 8 |
- (10p.)
3. Șirul 0, 0, 1, 4, 13, 38, 105, 280, 729 . . . este definit astfel: $f_1=f_2=0$, $f_3=1$, $f_n=4 \cdot f_{n-1}-3 \cdot f_{n-2}-2 \cdot f_{n-3}$ (unde **n** este un număr natural $n \geq 4$). Se citesc de la tastatură trei numere naturale **x, y** și **z** ($x \leq y < z \leq 10^9$), valorile a trei termeni aflați pe poziții consecutive în șirul dat, și se cere să se scrie în fișierul **bac.txt**, în ordine descrescătoare, separați prin câte un spațiu, toți termenii șirului care sunt mai mici sau egali cu **z**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
- Exemplu:** dacă se citesc numerele 38 105 280 fișierul conține numerele 280 105 38 13 4 1 0 0
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
- b. Scrieți programul Pascal corespunzător algoritmului proiectat. (8p.)

Examenul național de bacalaureat 2024
Proba E. d)
INFORMATICĂ

BAREM DE EVALUARE ȘI DE NOTARE
(comun pentru limbajele C/C++ și Pascal)

Varianta 8

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică
Filieră vocațională, profil militar, specializare matematică-informatică

- 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 sa.

SUBIECTUL I (20 de puncte)

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

SUBIECTUL al II - lea (40 de puncte)

1.	a. Răspuns corect: 43141	6p.	
	b. Pentru răspuns corect	6p.	Se acordă câte 3p. pentru fiecare dintre cele două numere conform cerinței (oricare dintre numerele 111, 1111, 11011) și se acordă numai câte 2p. pentru fiecare dintre cele două numere care conduce la rezultatul cerut, dar care nu aparține intervalului indicat.
	c. Pentru program corect -declarare a variabilelor -citire a datelor -afișare a datelor -instrucțiuni de decizie (*) -instrucțiune repetitivă -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 3p. 2p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile de decizie este conform cerinței.
	d. Pentru algoritm pseudocod corect -utilizare a unei structuri repetitive de tipul indicat (*) -aspecte specifice ale secvenței obținute prin înlocuire, conform cerinței (**) -algoritm complet, corectitudine globală a algoritmului ¹⁾	6p. 2p. 3p. 1p.	(*) Se acordă punctajul pentru orice formă de structură repetitivă de tipul cerut (repetă...până când, repetă...cât timp, execută...până când, execută...cât timp, do ... while etc.). (**) Se acordă numai 2p. dacă doar un aspect specific (expresie de continuare, echivalență pentru cazul $n \leq 9$) este conform cerinței.
2.	Răspuns corect (crin, gerbera, iris) (crin, iris, orhidee)	6p.	Se acordă câte 3p. pentru fiecare dintre cele două soluții conform cerinței.
3.	Pentru rezolvare corectă -expresie de acces la un câmp al structurii/înregistrării -afișare a datelor conform cerinței (*) -corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect specific (tratare a cazului în care câmpul nota1 corespunde unei note din mulțimea {do, re, mi}, tratare a cazului în care câmpul nota1 corespunde unei note din mulțimea {fa, sol}, utilizarea unor notații englezești ale notelor pe baza unor constante de tip caracter, corelare mesaje-note) conform cerinței.

SUBIECTUL al III - lea

(30 de puncte)

1.	Pentru subprogram corect -antet al subprogramului (*) -verificare a proprietății cerute (**) -instrucțiune/instrucțiuni de returnare a rezultatului -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului ¹⁾	10p. 2p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametru de intrare) conform cerinței. (**) Se acordă câte 1p. pentru fiecare aspect specific (algoritm de bază pentru verificare a unei proprietăți/numărare pentru o serie de valori, identificare a unui divizor al unui număr, divizori suport parcurși pentru determinarea unui număr prim, identificare a unor divizori primi ai unui număr, identificare a unor divizori primi consecutivi ai unui număr, număr de divizori primi suport în cadrul produsului) conform cerinței.
2.	Pentru program corect -declarare a unei variabile care să permită memorarea unui tablou bidimensional -citire a datelor -modificare a tabloului conform cerinței (*) -afișare a datelor în formatul cerut -declarare a variabilelor de tipuri simple, corectitudine globală a programului ¹⁾	10p. 1p. 1p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect specific (identificare a unui rând/a unei parcele cu flori/gazon, algoritm de bază pentru determinarea unei valori minime/maxime pentru o parcelă cu gazon dintr-o serie de valori, parcele suport parcurse pe un rând pentru determinarea valorii minime/maxime, identificare a unui rând cu gazon vecin anterior unui rând cu flori, înălțimi suport păstrate/actualizate pe un rând cu gazon, perechi de rânduri suport verificate/actualizate) conform cerinței.
3.	a. Pentru răspuns corect -descriere coerentă a algoritmului (*) -justificare a elementelor de eficiență b. Pentru program corect -operații cu fișiere: declarare, pregătire în vederea scrierii, scriere în fișier -determinare a valorilor 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 (de complexitate $O(n)$), care utilizează eficient memoria. O soluție posibilă memorează ultimii trei termeni afișați pe poziții consecutive, în ordine descrescătoare, z , y și x ($z > y \geq x$) și determină termenul curent (crt) pe baza celor trei termeni afișați anterior, cu formula $crt = [(4 \cdot y - 3 \cdot x - z) / 2]$; inițial se afișează, în ordine descrescătoare, cei trei termeni citiți, apoi, la fiecare pas, se determină și se afișează termenul curent și se actualizează cei trei termeni z , y și x ; algoritmul se încheie după afișarea a două valori 0.

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