



**Evaluarea la disciplina Informatică
în cadrul examenului național de bacalaureat 2011**

Specializările: **matematică-informatică**
 matematică-informatică intensiv informatică

Introducere

Disciplina Informatică are statutul de disciplină opțională la proba E. d) pentru candidații de la filiera teoretică, profil real, specializările matematică-informatică, matematică-informatică intensiv informatică și științe ale naturii.

Subiectele de bacalaureat nu vizează conținutul unui manual anume. Manualul școlar reprezintă doar unul dintre suporturile didactice utilizate de către profesori și elevi în vederea formării competențelor prevăzute de programa școlară.

Structura probei scrise la disciplina Informatică

Pentru această disciplină există patru tipuri diferite de variante de subiecte, în funcție de specializarea elevilor și limbajul studiat:

1. Informatică – pentru specializările matematică-informatică și matematică informatică, intensiv informatică - limbaj Pascal ;
2. Informatică – pentru specializările matematică-informatică și matematică informatică, intensiv informatică - limbaj C/C++ ;
3. Informatică – pentru specializarea științe ale naturii - limbaj Pascal;
4. Informatică – pentru specializarea științe ale naturii - limbaj C/C++.

Structura probei scrise cuprinde trei subiecte (I, II, III), fiecare a câte 30 de puncte. Fiecare dintre cele trei subiecte cuprinde cinci itemi, de dificultate diferită:

- foarte ușor – 4 puncte;
- ușor – 6 puncte;
- mediu – 10 puncte;
- dificil – 6 puncte;
- foarte dificil - 4 puncte.

Itemii utilizați în subiecte sunt de tipurile:

pentru subiectul I:

1. item obiectiv cu alegere multiplă;
2. item semiobiectiv – întrebări structurate.

pentru subiectul al II-lea

1. itemi obiectivi cu alegere multiplă;
2. itemi semiobiectivi – întrebări cu răspuns scurt și itemi de completare;
3. item subiectiv – rezolvare de probleme.

pentru subiectul al III-lea

1. item obiectiv cu alegere multiplă;
2. item semiobiectiv – întrebare cu răspuns scurt sau item de completare;
3. itemi subiectivi – rezolvare de probleme.

Competențele și conținuturile menționate în programa de bacalaureat pentru disciplina Informatică pot fi puse în valoare prin oricare dintre limbajele de programare Pascal, respectiv C/C++. De aceea, cerințele pentru cele două modele, corespunzătoare celor două limbaje, sunt comune, dar limbajul de implementare/ exemplificare este diferit.

Cerințele au același grad de dificultate pentru limbajele menționate.

Cerințele cuprinse în subiecte asigură o cuprindere echilibrată a competențelor evaluate și au un grad de complexitate care permite tratarea acestora în timpul stabilit.

Competențe de evaluat la disciplina Informatică

Competențele menționate în programele de bacalaureat pentru disciplina **Informatică - specializările matematică-informatică și matematică informatică intensiv informatică, limbajele Pascal/C/C++:**

- C1. construirea algoritmilor corespunzători unor prelucrări elementare și reprezentarea lor prin intermediul programelor pseudocod și programelor scrise în limbaj de programare (Pascal sau C/C++, la alegere);
- C2. analiza rezolvării unei probleme prin urmărirea evoluției valorilor variabilelor prelucrate de algoritmul corespunzător;
- C3. abstractizarea rezolvării prin construirea unor algoritmi echivalenți;
- C4. identificarea și utilizarea tipurilor de date predefinite specifice unui limbaj de programare;
- C5. definirea și utilizarea unor tipuri de date proprii;
- C6. identificarea și utilizarea operatorilor predefiniți elementari;
- C7. identificarea și utilizarea subprogramelor predefinite elementare;
- C8. identificarea și utilizarea regulilor sintactice specifice limbajului de programare studiat;
- C9. definirea și apelul unor subprograme proprii cu înțelegerea mecanismelor de transfer prin intermediul parametrilor;
- C10. identificarea proprietăților unor structuri de date necesare în rezolvarea problemelor cu ajutorul calculatorului și utilizarea unor modele de memorare a acestora;
- C11. organizarea datelor ce intervin în rezolvarea unei probleme utilizând structuri de date adecvate;
- C12. organizarea etapelor de prelucrare ce formează un algoritm utilizând structuri de control și module de program;
- C13. folosirea unor metode sistematice de rezolvare pentru probleme de generare;
- C14. analiza unor algoritmi echivalenți de rezolvare a unei probleme în vederea alegerii algoritmului optim.

Distribuirea competențelor pe subiecte este:

Subiect	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14
I														
II														
III														

Domeniile de conținuturi prin care se evaluează competențele propuse sunt:

- Algoritmi - pseudocod
- Elementele de bază ale unui limbaj de programare (Pascal sau C, la alegere)
- Subprograme predefinite
- Tipuri structurate de date
- Fișiere text
- Algoritmi elementari
- Subprograme definite de utilizator
- Recursivitate
- Metoda backtracking (iterativă sau recursivă)
- Generarea elementelor combinatoriale
- Grafuri

Distribuirea domeniilor de conținuturi pe subiecte este:

Nr.crt	Domeniu de conținut	I	II	III
1.	Algoritmi - pseudocod			
2.	Elementele de bază ale unui limbaj de programare			
3.	Subprograme predefinite			
4.	Tipuri structurate de date: - tablouri bidimensionale - șiruri de caractere - înregistrări			
	Tipuri structurate de date: - tablouri unidimensionale			
5.	Fișiere text			
6.	Algoritmi elementari			
7.	Subprograme definite de utilizator			
8.	Recursivitate			
9.	Metoda backtracking (iterativă sau recursivă)			
10.	Generarea elementelor combinatoriale			
11.	Grafuri			

Precizări privind evaluarea probei scrise la disciplina Informatică

În cadrul examenului de bacalaureat evaluarea se realizează prin raportare la competențele de evaluat, prezentate în programa disciplinei.

La baza construirii competențelor de evaluat s-au avut în vedere categoriile: cunoaștere, comprehensiune sau înțelegere, aplicare, analiză, sinteză, evaluare.

Cunoașterea vizează identificarea de termeni, relații, procese, observarea unor fenomene, procese, nominalizarea unor concepte, culegerea de date din surse variate, definirea unor concepte.

Înțelegerea vizează compararea unor date, reprezentarea unor date, stabilirea unor relații, calcularea unor rezultate parțiale, clasificari de date, reprezentarea unor date, sortarea – discriminarea, investigarea, descoperirea, explorarea etc.

Aplicarea vizează reducerea la o schemă sau model, anticiparea unor rezultate, reprezentarea datelor, remarcarea unor invarianți, rezolvarea de probleme prin modelare și algoritmizare etc.

Analiza vizează descrierea unor stări, sisteme, procese, fenomene, generarea de idei, argumentarea unor enunțuri, demonstrarea, compararea unor rezultate etc.

Sinteza vizează formularea unor concluzii, calcularea și evaluarea unor rezultate, interpretarea rezultatelor, analiza de situații, elaborarea de strategii, relaționări între diferite tipuri de reprezentări etc.

Evaluarea vizează aplicarea, generalizarea și particularizarea, integrarea, verificarea, optimizarea, transpunerea, realizarea de conexiuni, adaptare și adecvare la context etc.

Baremul de evaluare și de notare este instrumentul pe baza căruia se apreciază lucrările elevilor.

Baremul de evaluare și de notare este elaborat cu un grad înalt de obiectivitate și aplicabilitate, astfel încât să reducă la minimum diferențele de notare între corectori.

Baremul de evaluare și de notare este proiectat pe baza notării analitice. Notarea analitică are avantajul de a asigura rigurozitatea corectării, favorizând realizarea unor aprecieri obiective.

Baremul de evaluare și de notare permite evaluarea precisă a răspunsurilor la itemii de tip alegere multiplă. În cazul itemilor de tip rezolvare de probleme, baremul de

evaluare și de notare include elemente ale răspunsului care vor fi punctate. În acest fel candidatul primește punctaj pentru rezolvări parțiale ale cerinței itemului. Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct.

Se vor puncta însă corespunzător oricare alte metode de rezolvare corectă a cerinței.

Baremele realizate sunt comune pentru limbajele Pascal și C/C++.

În evaluarea lucrărilor elevilor, se vor avea în vedere observații specifice disciplinei, menționate în barem, cum ar fi, de exemplu, cele privind validarea datelor de intrare.

Pentru exemplificare, pentru specializările matematică – informatică și matematică – informatică, intensiv informatică, sunt propuse modele de subiect și barem pentru cele două tipuri de subiecte specifice celor două limbaje.

Examenul național de bacalaureat 2011
Proba E. d)
Proba scrisă la INFORMATICĂ

Limbajul C/C++

Specializările matematică informatică și matematică informatică, intensiv informatică

MODEL

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.
- În rezolvările cerute, identificatorii utilizați trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).

SUBIECTUL I **(30 de puncte)**

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Indicați care dintre expresiile C/C++ de mai jos are valoarea 1 dacă și numai dacă numărul natural memorat în variabila întregă n este divizibil cu 2 și cu 3. **(4p.)**
- a. $(n/2==0) \ || \ (n/3!=0)$ b. $(n\%3==2) \ || \ (n\%2==3)$
c. $(n\%2==0) \ \&\& \ (n\%3!=1)$ d. $(n\%2!=1) \ \&\& \ (n\%3==0)$

2. Se consideră algoritmul alăturat descris în pseudocod:

S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întregă a numărului real z .

- a) Scrieți numărul afișat dacă pentru variabila n se citește valoarea 6451. **(6p.)**
- b) Scrieți toate numerele de cel mult trei cifre care pot fi citite pentru variabila n astfel încât, pentru fiecare dintre acestea, numărul afișat în urma executării algoritmului să fie 26. **(4p.)**

```
citește n (număr natural nenul)
m ← 0
repetă
  c ← n%10
  n ← [n/10]
  dacă c < 5 atunci
    c ← 2*c
  m ← m*10+c
până când n=0
scrie m
```

- c) Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască structura **repetă...până când** cu o structură repetitivă de alt tip. **(6p.)**
- d) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

SUBIECTUL al II-lea

(30 de puncte)

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

- Variabilele i și j sunt de tip `int` ($1 \leq i \leq 5$, $1 \leq j \leq 5$), iar variabila A memorează elementele unui tablou bidimensional, cu 5 linii și 5 coloane, numerotate de la 1 la 5. Expresia care are valoarea 1 dacă și numai dacă elementul $A[i][j]$ se află pe diagonala secundară a tabloului este: **(4p.)**
 - `i+j==4`
 - `i+j==6`
 - `i<j`
 - `i==j`
- Numărul minim de muchii ale unui graf neorientat, conex, cu 10 de noduri, este: **(4p.)**
 - 5
 - 9
 - 10
 - 45

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

- Se consideră graful orientat, cu 5 vârfuri, numerotate de la 1 la 5, reprezentat prin matricea de adiacență alăturată. Enumerați vârfurile grafului care au gradul extern egal cu cel intern. **(6p.)**

0	1	0	1	0
0	0	0	0	1
0	1	0	0	0
0	1	1	0	0
0	1	0	1	0
- Se consideră tipul `elev`, definit alăturat, ce permite declararea unor variabile care să memoreze anumite date despre un elev: numele și media sa generală anuală.

```
typedef struct {  
    char nume[20];  
    float medie;  
} elev;  
int i;
```

Utilizând tipul `elev`, declarați o variabilă, cu numele `clasa`, care să poată memora datele menționate mai sus pentru fiecare dintre cei 30 de elevi ai unei clase, apoi scrieți instrucțiunile care pot completa zona punctată din secvența de instrucțiuni de mai jos, pentru a inițializa, prin citire de la tastatură, variabila respectivă.

```
for (i=0; i<30; i++)  
    .....
```

(6p.)
- Scrieți un program C/C++ care citește de la tastatură un număr natural nenul, n ($n \leq 10$), și n cuvinte, fiecare cuvânt fiind format din cel puțin 3 și cel mult 20 de litere ale alfabetului englez. La introducerea datelor, după fiecare cuvânt se tastează Enter. Programul construiește în memorie, și apoi afișează pe ecran, șirul de caractere obținut prin preluarea primelor trei litere ale fiecărui cuvânt citit, în ordinea citirii acestora.
Exemplu: pentru $n=3$ și cuvintele
`frumos`
`BUN`
`Intelept`
se construiește șirul de caractere `fruBUNInt` **(10p.)**

SUBIECTUL al III-lea

(30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Se consideră subprogramele recursive **R1** și **R2**, definite mai jos.

```
long R1(int x, int p){
    if(p==0) return 1;
    return x*R1(x,p-1);
}

long R2(int x, int p){
    long f;
    if(p==0) return 1;
    if(p%2==0) { f=R2(x,p/2);
                return f*f;
            }
    return x*R2(x,p-1);
}
```

La apel, pentru parametrii $x=3$ și $p=5$, returnează valoarea expresiei 3^5 : **(4p.)**

- a. atât **R1**, cât și **R2** b. numai **R1** c. numai **R2** d. nici **R1**, nici **R2**

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Având la dispoziție cinci flori diferite, **lalea**, **narcisă**, **mac**, **frezie**, **garoafă**, se utilizează metoda backtracking pentru a obține toate posibilitățile de a forma buchete de câte trei flori, știind că în cadrul unui buchet nu contează ordinea de așezare a florilor. Primele patru soluții obținute sunt, în această ordine: (**lalea**, **narcisă**, **mac**), (**lalea**, **narcisă**, **frezie**), (**lalea**, **narcisă**, **garoafă**), (**lalea**, **mac**, **frezie**). Scrieți ultimele două soluții generate, în ordinea obținerii lor. **(6p.)**
3. Subprogramul **eval** are trei parametri:
- **a** și **b**, prin care primește câte un număr întreg cu cel mult patru cifre;
 - **rez**, prin care furnizează numărul obținut prin însumarea valorilor transmise prin parametrii **a** și **b**.

Exemplu: dacă $a=3$, $b=5$, în urma apelului, valoarea furnizată prin parametrul **rez** este 8.

a) Scrieți numai antetul subprogramului **eval**. **(4p.)**

b) Fișierul **BAC.IN** conține un șir de numere întregi, cu cel mult două cifre fiecare. În fișier se află cel puțin două și cel mult 100 de numere, separate prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișierul **BAC.IN** și, utilizând apeluri utile ale subprogramului **eval**, calculează suma acestora și apoi afișează pe ecran valoarea obținută.

Exemplu: dacă fișierul conține numerele de mai jos, se va afișa pe ecran valoarea 11.

2 12 -7 4 **(6p.)**

4. Numim „**k**-pereche” într-un tablou unidimensional două elemente cu valori egale ale acestuia, între care se află alte **k** elemente. Scrieți un program C/C++ care citește de la tastatură două numere naturale, **n** și **k** ($3 \leq n \leq 100$, $1 \leq k \leq n-2$), cele **n** elemente ale unui tablou unidimensional, numere naturale, fiecare cu cel mult trei cifre, apoi afișează pe ecran numărul de „**k**-perechi” din tabloul citit. **Exemplu:** dacă $n=15$, $k=3$, iar tabloul este

```
(1, 2, 5, 0, 3, 2, 9, 0, 2, 2, 0, 1, 2, 3, 7)
```

se va afișa pe ecran valoarea 4.

(10p.)

Examenul național de bacalaureat 2011
Proba E. d)
Proba scrisă la INFORMATICĂ

Limbajul Pascal

Specializările matematică informatică și matematică informatică, intensiv informatică

MODEL

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.
- În rezolvările cerute, identificatorii utilizați trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).

SUBIECTUL I

(30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Indicați care dintre expresiile `Pascal` de mai jos are valoarea `true` dacă și numai dacă numărul natural memorat în variabila întregă `n` este divizibil cu 2 și cu 3. **(4p.)**
- a. `(n div 2=0) or (n div 3<>0)` b. `(n mod 3=2) or (n mod 2=3)`
c. `(n mod 2=0) and (n mod 3<>1)` d. `(n mod 2<>1) and (n mod 3=0)`

2. Se consideră algoritmul alăturat descris în pseudocod:

S-a notat cu `x%y` restul împărțirii numărului natural `x` la numărul natural nenul `y` și cu `[z]` partea întregă a numărului real `z`.

- a) Scrieți numărul afișat dacă pentru variabila `n` se citește valoarea `6451`. **(6p.)**
- b) Scrieți toate numerele de cel mult trei cifre care pot fi citite pentru variabila `n` astfel încât, pentru fiecare dintre acestea, numărul afișat în urma executării algoritmului să fie `26`. **(4p.)**
- c) Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască structura `repetă...până când` cu o structură repetitivă de alt tip. **(6p.)**
- d) Scrieți programul `Pascal` corespunzător algoritmului dat. **(10p.)**

```
citește n (număr natural  
nenul)  
m ← 0  
repetă  
  c ← n%10  
  n ← [n/10]  
  dacă c < 5 atunci  
    c ← 2*c  
  m ← m*10+c  
până când n=0  
scrie m
```


SUBIECTUL al II-lea

(30 de puncte)

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Variabilele i și j sunt de tip `integer` ($1 \leq i \leq 5$, $1 \leq j \leq 5$), iar variabila A memorează elementele unui tablou bidimensional, cu 5 linii și 5 coloane, numerotate de la 1 la 5. Expresia care are valoarea `true` dacă și numai dacă elementul $A[i, j]$ se află pe diagonala secundară a tabloului este: (4p.)
 - a. $i+j=4$
 - b. $i+j=6$
 - c. $i < j$
 - d. $i=j$
2. Numărul minim de muchii ale unui graf neorientat, conex, cu 10 de noduri, este: (4p.)
 - a. 5
 - b. 9
 - c. 10
 - d. 45

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

3. Se consideră graful orientat, cu 5 vârfuri, numerotate de la 1 la 5, reprezentat prin matricea de adiacență alăturată. Enumerați vârfurile grafului care au gradul extern egal cu cel intern. (6p.)

0 1 0 1 0
0 0 0 0 1
0 1 0 0 0
0 1 1 0 0
0 1 0 1 0
4. Se consideră tipul `elev`, definit alăturat, ce permite declararea unor variabile care să memoreze anumite date despre un elev: numele și media sa generală anuală.

```
type elev=record  
    nume:string[20];  
    medie:real  
end;  
var i:integer;
```

Utilizând tipul `elev`, declarați o variabilă, cu numele `clasa`, care să poată memora datele menționate mai sus pentru fiecare dintre cei 30 de elevi ai unei clase, apoi scrieți instrucțiunile care pot completa zona punctată din secvența de instrucțiuni de mai jos, pentru a inițializa, prin citire de la tastatură, variabila respectivă.

```
for i:=1 to 30 do  
    .....
```

(6p.)
5. Scrieți un program `Pascal` care citește de la tastatură un număr natural, nenul, n ($n \leq 10$), și n cuvinte, fiecare cuvânt fiind format din cel puțin 3 și cel mult 20 de litere ale alfabetului englez. La introducerea datelor, după fiecare cuvânt se tastează `Enter`. Programul construiește în memorie, și apoi afișează pe ecran, șirul de caractere obținut prin preluarea primelor trei litere ale fiecărui cuvânt citit, în ordinea citirii acestora.
Exemplu: pentru $n=3$ și cuvintele
`frumos`
`BUN`
`Intelept`
se construiește șirul de caractere `fruBUNInt` (10p.)

SUBIECTUL al III-lea

(30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Se consideră subprogramele recursive R1 și R2, definite mai jos.

```
function R1(x,p:integer):longint;  
begin  
  if p=0 then R1:=1  
  else R1:=x*R1(x,p-1)  
end;
```

```
function R2(x,p:integer):longint;  
var f:longint;  
begin  
  if p=0 then R2:=1  
  else if p mod 2=0 then  
    begin f:=R2(x,p div 2);  
          R2:=f*f  
    end  
  else R2:=x*R2(x,p-1)  
end;
```

La apel, pentru parametrii $x=3$ și $p=5$, returnează valoarea expresiei 3^5 :

(4p.)

- a. atât R1, cât și R2 b. numai R1 c. numai R2 d. nici R1, nici R2

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Având la dispoziție cinci flori diferite, **lalea**, **narcisă**, **mac**, **frezie**, **garoafă**, se utilizează metoda backtracking pentru a obține toate posibilitățile de a forma buchete de câte trei flori, știind că în cadrul unui buchet nu contează ordinea de așezare a florilor. Primele patru soluții obținute sunt, în această ordine: (**lalea**, **narcisă**, **mac**), (**lalea**, **narcisă**, **frezie**), (**lalea**, **narcisă**, **garoafă**), (**lalea**, **mac**, **frezie**). Scrieți ultimele două soluții generate, în ordinea obținerii lor. **(6p.)**

3. Subprogramul **eval** are trei parametri:
- **a** și **b**, prin care primește câte un număr întreg cu cel mult patru cifre;
 - **rez**, prin care furnizează numărul obținut prin însumarea valorilor transmise prin parametrii **a** și **b**.

Exemplu: dacă $a=3$, $b=5$, în urma apelului, valoarea furnizată prin parametrul **rez** este 8.

- a) Scrieți numai antetul subprogramului **eval**. **(4p.)**

b) Fișierul **BAC.IN** conține un șir de numere întregi, cu cel mult două cifre fiecare. În fișier se află cel puțin două și cel mult 100 de numere, separate prin câte un spațiu.

Scrieți un program **Pascal** care citește toate numerele din fișierul **BAC.IN** și, utilizând apeluri utile ale subprogramului **eval**, calculează suma acestora și apoi afișează pe ecran valoarea obținută.

Exemplu: dacă fișierul conține numerele de mai jos, se va afișa pe ecran valoarea 11.

2 12 -7 4

(6p.)

4. Numim „**k**-pereche” într-un tablou unidimensional două elemente cu valori egale ale acestuia, între care se află alte **k** elemente. Scrieți un program **Pascal** care citește de la tastatură două numere naturale, **n** și **k** ($3 \leq n \leq 100$, $1 \leq k \leq n-2$), cele **n** elemente ale unui tablou unidimensional, numere naturale, fiecare cu cel mult trei cifre, apoi afișează pe ecran numărul de „**k**-perechi” din tabloul citit.

Exemplu: dacă $n=15$, $k=3$, iar tabloul este

(1, 2, 5, 0, 3, 2, 9, 0, 2, 2, 0, 1, 2, 3, 7)

se va afișa pe ecran valoarea 4.

(10p.)

Examenul național de bacalaureat 2011
Proba E. d)
Proba scrisă la Informatică

Specializările matematică-informatică și matematică-informatică intensiv informatică

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

MODEL

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct.
- Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.
- Se vor lua în considerare atât implementările concepute pentru compilatoare pe 16 biți, cât și cele pentru compilatoare pe 32 de biți.

SUBIECTUL I

30 de puncte

1.	d	4p.	
2. a)	2586	6p.	
b)	31 61 310 610 (*)	4p.	(*) Se acordă câte 1p. pentru fiecare dintre cele patru numere corecte.
c)	Pentru algoritm pseudocod corect - structură repetitivă corectă (*) - echivalența prelucrării realizate - algoritm complet - corectitudine globală	6p. 2p. 2p. 1p. 1p.	(*) Se va puncta orice formă corectă de structură repetitivă de alt tip.
d)	Pentru program corect - declararea corectă a tuturor variabilelor - citire corectă - scriere corectă - structură repetitivă cu test final corectă - structură de decizie corectă - atribuiri corecte - corectitudinea globală a programului ¹⁾	10p. 2p. 1p. 1p. 2p. 2p. 1p. 1p.	

SUBIECTUL al II-lea

30 de puncte

1) b	4p.	
2) b	4p.	
3) Răspuns corect: 3, 4 (*)	6p.	(*) Se acordă câte 3p. pentru fiecare dintre cele două noduri menționate corect.

Probă scrisă la **Informatică**

Specializările matematică-informatică și matematică-informatică intensiv informatică

Barem de evaluare și de notare

4)	Pentru răspuns corect - declararea variabilei de un tip corespunzător - accesul corect la câmpurile înregistrării - inițializarea corectă a celor două câmpuri	6p. 2p. 2p. 2x1p.	
5)	Pentru program corect - declararea corectă a tuturor variabilelor (șir de caractere și variabile simple) - citire corectă a datelor - extragerea primelor trei caractere dintr-un cuvânt - construirea în memorie a șirului cerut - afișarea rezultatului (*) - corectitudinea globală a programului ¹⁾	10p. 1+1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă punctajul și dacă șirul nu a fost construit în memorie.

SUBIECTUL al III-lea

30 de puncte

1)	a	4p.	
2)	Răspuns corect: (narcisă, frezie, garoafă), (mac, frezie, garoafă) (*)	6p.	(*) Se acordă doar 3p. dacă numai una dintre cele două soluții este corectă. Elevii nu vor fi depunctați dacă nu au delimitat soluțiile între paranteze.
3)	a) Pentru antet corect - structură antet principal corectă - declarare corectă a parametrilor de intrare - declarare corectă a parametrului de ieșire	4p. 1p. 2x1p. 1p.	
	b) Pentru program corect - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - apel corect al subprogramului - calculul corect al valorii expresiei - declarare de variabile, afișare date, corectitudinea globală a programului ¹⁾	6p. 1p. 1p. 3p. 1p.	
4)	Pentru program corect - declarare variabile: simple și tablou - accesul corect la un element al tabloului - citire tablou - determinarea unei k-perechi - numărarea tuturor k-perechilor - afișarea rezultatului - corectitudinea globală a programului ¹⁾	10p. 1+1p. 1p. 1p. 2p. 2p. 1p. 1p.	(*) Se acordă numai 3p. dacă după prelucrare s-au modificat pozițiile termenilor impari.

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