



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

Specializarea științe ale naturii

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 - specializarea științe ale naturii, 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 subprogramelelor predefinite elementare;
- C8. identificarea și utilizarea regulilor sintactice specifice limbajului de programare studiat;
- C9. identificarea proprietăților unor structuri de date necesare în rezolvarea problemelor cu ajutorul calculatorului și utilizarea unor modele de memorare a acestora;
- C10. organizarea datelor ce intervin în rezolvarea unei probleme utilizând structuri de date adecvate;
- C11. organizarea etapelor de prelucrare ce formează un algoritm utilizând structuri de control;
- C12. 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
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/C++, la alegere)

Subprograme predefinite

Tipuri structurate de date

Fișiere text

Algoritmi elementari

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 unidimensionale			
5.	Fișiere text			
6.	Algoritmi elementari			

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, clasificări 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 specializarea științe ale naturii, sunt propuse modele de subiect și barem pentru cele două tipuri de subiecte, specifice celor două limbaje.

THEMA II

(30 Puncte)

Für jeden der Punkte 1 und 2 schreibt auf das Prüfungsblatt, den Buchstaben welcher der richtigen Antwort entspricht.

1. Ein C/C++ Ausdruck der als Wert den kleinsten der unterschiedlichen Zahlen gespeichert in der ganzen Variablen x beziehungsweise y hat, ist: **(4P.)**
 - a. $(x+y-\text{abs}(x-y))/2$
 - b. $(x+y+\text{abs}(x-y))/2$
 - c. $(\text{abs}(x+y)-x-y)/2$
 - d. $(\text{abs}(x-y)+\text{abs}(y-x))/2$
2. Eine C/C++ Anweisung, deren Durchführung das Speichern in der reellen Variablen x des arithmetischen Mittels der Werte der reellen Variablen x , y und z erzeugt, ist: **(4P.)**
 - a. $x = (x + y + z)/2;$
 - b. $x = x + y + z/3;$
 - c. $x = x/1/3 + y/1/3 + z/1/3;$
 - d. $x = x/3 + (y + z)/2;$

Schreibt auf das Prüfungsblatt die Antwort für jede der folgenden Anforderungen.

3. Sei die Variable `simbol`, vom Typ `char`. Schreibt die C/C++ Anweisungen durch die man auf dem Bildschirm die Nachricht `Este litera mare` anschreibt, wenn die Variable einen Großbuchstaben des englischen Alphabetes speichert oder die Nachricht `Nu este litera mare` sonst. **(6P.)**
4. Man liest eine natürliche Zahl x ($x > 1$) und man soll den Wert anschreiben den man erhält durch das Addieren der Exponenten der Faktoren die in der Zerlegung von x in Primfaktoren vorkommen.
Beispiel: wenn $x=135$ ist, dann ist der erhaltene Wert 4 ($135=3^3 \cdot 5^1$, also die verlangte Summe ist $3+1=4$).
 - a) Schreibt im Pseudocode den Lösungsalgorithmus für die gestellte Aufgabe. **(10P.)**
 - b) Erläutert die Rolle aller Variablen, die in der Bearbeitung bei Punkt a) vorkommen und gebt die Eingabe- beziehungsweise Ausgabedaten für die gestellte Aufgabe an. **(6P.)**

THEMA III

(30 Puncte)

Für Punkt 1 schreibt auf das Prüfungsblatt, den Buchstaben welcher der richtigen Antwort entspricht.

1. Sei das eindimensionale Feld v , mit den Elementen $v_1=1, v_2=7, v_3=5, v_4=3$. In dem nebenan beschriebenen Sortieralgorithmus beschriftet man mit $a \leftrightarrow b$ die Vertauschung der Werte der Variablen a und b .
Um die vier Elemente des Feldes v steigend zu sortieren, ist die Anzahl der Vertauschung die mit Hilfe der nebenstehenden Sequenz durchgeführt werden müssen:

```
wiederhole
  ok ← 1
  für i ← 1, 3 wiederhole
    wenn  $v_i > v_{i+1}$  dann
      ok ← 0
       $v_i \leftrightarrow v_{i+1}$ 
  bis ok = 1
```

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

(4P.)

Schreibt auf das Prüfungsblatt die Antwort für jede der folgenden Anforderungen.

2. Sei die Fibonacci-Folge in der die ersten beiden Glieder $f_1=1, f_2=1$ sind und das k -te ($k > 2$) Element berechnet man mit Hilfe der Formel $f_k = f_{k-1} + f_{k-2}$.
In der unteren Sequenz sind alle Variablen vom Typ `int`. Schreibt die Anweisungen, die die Auslassungspunkte ersetzen können, so dass nach dem Durchführen der Sequenz man auf dem Bildschirm die ersten 20 Glieder der Fibonacci-Folge anschreiben kann, getrennt durch je ein Leerzeichen.

```
f1=1; f2=1;
cout<<"1 1 "; | printf("1 1 ");
for(i=3;i<=20;i++){
  f3=f1+f2;
  .....
}
```

(6P.)

3. Erläutert in Umgangssprache eine Bedingung die die 100 Elemente, mit unterschiedlichen Werten eines eindimensionalen Feldes erfüllen müssen, um das Binäre Suchen auf dem Feld anwenden zu können.

(4P.)

4. Die Datei `BAC.IN` enthält einen arithmetischen Ausdruck gebildet aus natürlichen Zahlen, mit höchstens zwei Ziffern jede und die binären arithmetischen Operatoren `+` und `-`. Der Ausdruck enthält wenigstens einen und höchstens 100 Operatoren und ist richtig. In der Datei befindet sich jeder Operand beziehungsweise Operator des Ausdruckes auf je einer Reihe.
Schreibt ein `C/C++` Programm, das den Ausdruck aus der Datei `BAC.IN` einliest, ihn berechnet und nachher den erhaltenen Wert auf dem Bildschirm anschreibt.

2
+
12
-
7
+
4

Beispiel: wenn die Datei den nebenstehenden Inhalt hat, wird auf dem Bildschirm der Wert 11 angeschrieben.

(6P.)

5. Man nennt „ k -Paar“ in einem eindimensionalen Feld, zwei Elemente mit gleichen Werten, zwischen denen sich andere k Elemente befinden.
Schreibt ein `C/C++` Programm das von der Tastatur zwei natürliche Zahlen, n und k ($3 \leq n \leq 100, 1 \leq k \leq n-2$) einliest, die n Elemente eines eindimensionalen Feldes, natürliche Zahlen, jede mit höchstens drei Ziffern, dann schreibt es auf dem Bildschirm die Anzahl der „ k -Paare“ aus dem eingelesenen Feld.

Beispiel: wenn $n=15, k=3$, und das Feld ist

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

dann wird auf dem Bildschirm der Wert 4 angeschrieben.

(10P.)

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

Limbajul Pascal
Specializarea științe ale naturii

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).

THEMA I

(30 Puncte)

Für Punkt 1 schreibt auf das Prüfungsblatt, den Buchstaben welcher der richtigen Antwort entspricht.

1. Gebt an welcher der unteren `Pascal`-Ausdrückeden Wert `true` hat, wenn und nur wenn die natürliche Zahl gespeichert in der ganzen Variablen `n` teilbar durch 2 und 3 ist. **(4P.)**
- a. $(n \text{ div } 2=0) \text{ or } (n \text{ div } 3<>0)$ b. $(n \text{ mod } 3=2) \text{ or } (n \text{ mod } 2=3)$
c. $(n \text{ mod } 2=0) \text{ and } (n \text{ mod } 3<>1)$ d. $(n \text{ mod } 2<>1) \text{ and } (n \text{ mod } 3=0)$

2. Sei nebenstehender Pseudocode-Algorithmus:

Man bezeichnet mit $x\%y$ den Rest der Division der natürlichen Zahl x durch die natürliche von Null verschiedene Zahl y und mit $[z]$ den ganzen Teil der reellen Zahl z .

- a) Schreibt die Zahl, die angeschrieben wird, wenn für die Variable `n` der Wert 6451 eingelesen wird. **(6P.)**
- b) Schreibt alle dreistelligen Zahlen die für die Variable `n` eingelesen werden können, so dass für jede dieser nach dem Durchführen des Algorithmus die angeschriebene Zahl 26 sein soll. **(4P.)**

```
lies n (natürliche, von Null  
          verschiedene Zahl)  
m ← 0  
wiederhole  
  c ← n%10  
  n ← [n/10]  
  wenn c<5 dann  
    c ← 2*c  
  ■  
  m ← m*10+c  
bis n=0  
schreibe m
```

- c) Schreibt einen äquivalenten Pseudocode-Algorithmus, in dem man die Struktur `wiederhole...bis` mit einer anderen Wiederholungsstruktur ersetzen soll. **(6P.)**
- d) Schreibt das dem gegebenen Algorithmus entsprechende `Pascal`-Programm. **(10P.)**

THEMA II

(30 Puncte)

Für jeden der Punkte 1 und 2 schreibt auf das Prüfungsblatt, den Buchstaben welcher der richtigen Antwort entspricht.

1. Ein `Pascal`-Ausdruck der als Wert den kleinsten der unterschiedlichen Zahlen gespeichert in der ganzen Variablen `x` beziehungsweise `y` hat, ist: **(4P.)**
 - a. $(x+y-\text{abs}(x-y)) \text{ div } 2$
 - b. $(x+y+\text{abs}(x-y)) \text{ div } 2$
 - c. $(\text{abs}(x+y)-x-y) \text{ div } 2$
 - d. $(\text{abs}(x-y)+\text{abs}(y-x)) \text{ div } 2$
2. Eine `Pascal`-Anweisung, deren Durchführung das Speichern in der reellen Variablen `x` des arithmetischen Mittels der Werte der reellen Variablen `x`, `y` und `z` erzeugt, ist: **(4P.)**
 - a. `x := (x + y + z)/2`
 - b. `x := x + y + z/3`
 - c. `x := x/1/3 + y/1/3 + z/1/3`
 - d. `x := x/3+ (y + z)/2`

Schreibt auf das Prüfungsblatt die Antwort für jede der folgenden Anforderungen.

3. Sei die Variable `simbol` vom Typ `char`. Schreibt die `Pascal`-Anweisungen durch die man auf dem Bildschirm die Nachricht `Este litera mare` anschreibt, wenn die Variable einen Großbuchstaben des englischen Alphabetes speichert oder die Nachricht `Nu este litera mare` sonst. **(6P.)**
4. Man liest eine natürliche Zahl `x` (`x>1`) und man soll den Wert anschreiben, den man erhält durch das Addieren der Exponenten der Faktoren die in der Zerlegung von `x` in Primfaktoren vorkommen.
Beispiel: wenn `x=135` ist der erhaltene Wert `4` ($135=3^3 \cdot 5^1$, also die verlangte Summe ist $3+1=4$).
 - a) Schreibt in Pseudocode den Lösungsalgorithmus für die gestellte Aufgabe. **(10P.)**
 - b) Erläutert die Rolle aller Variablen, die in der Bearbeitung bei Punkt a) vorkommen und gebt die Eingabe- beziehungsweise Ausgabedaten für die gestellte Aufgabe an. **(6P.)**

THEMA III

(30 Puncte)

Für Punkt 1 schreibt auf das Prüfungsblatt, den Buchstaben welcher der richtigen Antwort entspricht.

1. Sei das eindimensionale Feld v , mit den Elementen $v_1=1, v_2=7, v_3=5, v_4=3$. In dem nebenan beschriebenen Sortieralgorithmus beschriftet man mit $a \leftrightarrow b$ die Vertauschung der Werte der Variablen a und b .
Um die vier Elemente des Feldes v steigend zu sortieren, ist die Anzahl der Vertauschungen, die mit Hilfe der nebenstehenden Sequenz durchgeführt werden müssen: **(4P.)**

```
wiederhole
  ok ← 1
  für i ← 1, 3 wiederhole
    wenn  $v_i > v_{i+1}$  dann
      ok ← 0
       $v_i \leftrightarrow v_{i+1}$ 
  bis ok = 1
```

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

Schreibt auf das Prüfungsblatt die Antwort für jede der folgenden Anforderungen.

2. Sei die Fibonacci-Folge in der die ersten beiden Glieder $f_1=1, f_2=1$ sind und das k -te ($k > 2$) Element berechnet man mit Hilfe der Formel $f_k = f_{k-1} + f_{k-2}$.
In der unteren Sequenz sind alle Variablen vom Typ **integer**. Schreibt die Anweisungen, die die Auslassungspunkte ersetzen können, so dass nach dem Durchführen der Sequenz man auf dem Bildschirm die ersten 20 Glieder der Fibonacci-Folge anschreiben kann, getrennt durch je ein Leerzeichen.

```
f1:=1; f2:=1;
write('1 1 ');
for i:=3 to 20 do
  begin
    f3:=f1+f2;
    .....
  end;
```

(6P.)

3. Erläutert in Umgangssprache eine Bedingung die die 100 Elemente, mit unterschiedlichen Werten eines eindimensionalen Feldes erfüllen müssen, um das Binäre Suchen auf dem Feld anwenden zu können. **(4P.)**

4. Die Datei **BAC.IN** enthält einen arithmetischen Ausdruck gebildet aus natürlichen Zahlen, mit höchstens zwei Ziffern jede und die binären arithmetischen Operatoren **+** und **-**. Der Ausdruck enthält wenigstens einen und höchstens 100 Operatoren und ist richtig. In der Datei befindet sich jeder Operand beziehungsweise Operator des Ausdruckes auf je einer Reihe. Schreibt ein **Pascal**-Programm, das den Ausdruck aus der Datei **BAC.IN** einliest, ihn berechnet und nachher den erhaltenen Wert auf dem Bildschirm anschreibt.

2
+
12
-
7
+
4

Beispiel: wenn die Datei den nebenstehenden Inhalt hat, dann wird auf dem Bildschirm der Wert 11 angeschrieben. **(6P.)**

5. Man nennt „**k**-Paar“ in einem eindimensionalen Feld, zwei Elemente mit gleichen Werten, zwischen denen sich andere **k** Elemente befinden.
Schreibt ein **Pascal**-Programm das von der Tastatur zwei natürliche Zahlen n und k ($3 \leq n \leq 100, 1 \leq k \leq n-2$) einliest, die n Elemente eines eindimensionalen Feldes, natürliche Zahlen, jede mit höchstens drei Ziffern, dann schreibt es auf dem Bildschirm die Anzahl der „**k**-Paare“ aus dem eingelesenen Feld.

Beispiel: wenn $n=15, k=3$, und das Feld ist

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

dann wird auf dem Bildschirm der Wert 4 angeschrieben.

(10P.)

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

Specializarea științe ale naturii

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) a	4p.	
2) c	4p.	
3) Pentru rezolvare corectă - determinarea unei litere mari - afișarea corectă a mesajului, conform cerinței	6p. 4p. 2p.	

4)	a)	Pentru rezolvare corectă - citirea datelor - determinarea puterii la care apare un factor prim în descompunerea cerută - determinarea corectă a numărului cerut - scrierea principial corectă a structurilor de control (*) - scrierea rezultatului	10p. 2p. 3p. 1p. 3p. 1p.	(*) Se va puncta orice formă corectă de structură repetitivă (de exemplu execută...cât timp, execută... până când, repetă...până când etc.) sau decizională
	b)	Pentru răspuns corect - menționarea rolului variabilelor utilizate (*) - date de intrare identificate corect - date de ieșire identificate corect	6p. 2p. 2p. 2p.	(*) Se acordă numai 1p. dacă s-au identificat doar o parte din variabilele utilizate sau nu pentru toate variabilele este corect menționat rolul acestora.

SUBIECTUL al III-lea

30 de puncte

1)	c	4p.	
2)	Pentru răspuns corect: - determinarea termenilor șirului, conform cerinței - afișarea termenilor indicați	6p. 4p. 2p.	
3)	Pentru răspuns corect	4p.	Un răspuns posibil este: elementele șirului trebuie să fie ordonate
4)	Pentru program corect - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - realizarea operației diferențiat, în funcție de operatorul citit - calculul corect al valorii expresiei - declarare de variabile, afișare date, corectitudinea globală a programului ¹⁾	6p. 1p. 1p. 3p. 1p.	
5)	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.