



**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
<b>I</b>														
<b>II</b>														
<b>III</b>														

**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, 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 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.



**THEMA II**

**(30 Puncte)**

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

1. Die Variablen  $i$  und  $j$  sind vom Typ `int` ( $1 \leq i \leq 5$ ,  $1 \leq j \leq 5$ ) und die Variable  $A$  speichert die Elemente eines bidimensionalen Feldes mit 5 Reihen und 5 Spalten, nummeriert von 1 bis 5. Der Ausdruck, der den Wert 1 hat, wenn und nur wenn das Element  $A[i][j]$  sich auf der Nebendiagonale des Feldes befindet, ist: **(4P.)**  
a.  $i+j==4$                       b.  $i+j==6$                       c.  $i < j$                       d.  $i==j$
2. Die minimale Kantenanzahl eines ungerichteten, konnexen Graphs mit 10 Knoten, ist: **(4P.)**  
a. 5                      b. 9                      c. 10                      d. 45

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

3. Sei der gerichtete Graph mit 5 Spitzen, nummeriert von 1 bis 5, dargestellt durch die nebenstehende Adjazenzmatrix. Zählt die Spitzen des Graphs, auf die den äußeren Grad gleich mit dem inneren haben. **(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. Sei der Typ `elev`, nebenstehend definiert, der das Deklarieren einer Variablen erlaubt die gewisse Daten über einen Schüler speichern soll: Name und die Jahresmittelnote.  
Mit Hilfe des Typs `elev`, deklariert eine Variable, mit dem Namen `clasa`, die die oberen Daten speichern soll für jeden der 30 Schüler einer Klasse, dann schreibt die Anweisungen die die Auslassungspunkte ersetzen können in der unteren Sequenz, um durch Einlesen von der Tastatur, die betreffenden Variable zu initialisieren.  

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

**(6P.)**

```
typedef struct {  
    char nume[20];  
    float medie;  
}elev;  
  
int i;
```
5. Schreibt ein C/C++ Programm das von der Tastatur eine natürliche, von Null verschiedene Zahl  $n$  ( $n \leq 10$ ) einliest und  $n$  Wörter, jedes Wort ist aus wenigstens 3 und höchstens 20 Buchstaben des englischen Alphabetes gebildet. Bei der Dateneingabe wird nach jedem Wort Enter eingegeben. Das Programm baut im Speicher und schreibt nachher auf dem Bildschirm, die Folge die man erhält durch das Übernehmen der ersten drei Buchstaben aus jedem Wort, in der Reihenfolge in der diese eingelesen wurden.  
**Beispiel:** für  $n=3$  und die Wörter  
`frumos`  
`BUN`  
`Intelept`  
baut man die Zeichenfolge `fruBUNInt` **(10P.)**

**THEMA III**

**(30 Puncte)**

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

1. Seien die rekursiven Unterprogramme R1 und R2, weiter unten definiert.

```
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);  
}
```

Bei dem Aufruf für die Parameter  $x=3$  und  $p=5$ , liefert es den Wert des Ausdrucks  $3^5$ : (4P.)

- a. sowohl R1, als auch R2      b. nur R1      c. nur R2      d. weder R1, noch R2

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

2. Wenn man zur Verfügung fünf verschiedene Blumen hat, **lalea**, **narcisă**, **mac**, **frezie**, **garoafă**, benützt man die Backtracking Methode um alle Möglichkeiten anzugeben durch die man Sträuße von je drei Blumen bilden kann, wenn man weiß, dass innerhalb eines Straußes die Reihenfolge in der die Blumen angeordnet sind nicht wichtig ist. Die ersten vier erhaltenen Lösungen, in dieser Reihenfolge sind: (**lalea**, **narcisă**, **mac**), (**lalea**, **narcisă**, **frezie**), (**lalea**, **narcisă**, **garoafă**), (**lalea**, **mac**, **frezie**). Schreibt die letzten zwei erzeugten Lösungen, in der Reihenfolge in der sie erhalten wurden. (6P.)

3. Das Unterprogramm **eval** hat drei Parameter:

- **a** und **b**, durch die es je eine ganze Zahl erhält mit höchstens vier Ziffern;
- **rez**, durch den es eine Zahl liefert, erhalten durch das Addieren der Werte zugeschickt durch die Parameter **a** und **b**.

**Beispiel:** wenn  $a=3$ ,  $b=5$ , wird nach dem Aufruf, der Wert geliefert durch den Parameter **rez**, 8 sein.

- a) Schreibt nur die Kopfzeile des Unterprogramms **eval**. (4P.)

b) Die Datei **BAC.IN** enthält eine Folge von ganzen Zahlen, mit höchstens zwei Ziffern jede. In der Datei befinden sich weingstens zwei und höchstens 100 Zahlen, getrennt durch je ein Leerzeichen.

Schreibt ein C/C++ Programm, das alle Zahlen aus der Datei **BAC.IN** einliest und das mit Hilfe von gültigen Aufrufen des Unterprogramms **eval**, die Summe dieser Zahlen berechnet und nachher den erhaltenen Wert auf dem Bildschirm anschreibt.

**Beispiel:** wenn die Datei die untenstehenden Zahlen enthält, wird auf dem Bildschirm der Wert 11 angeschrieben.

2 12 -7 4 (6P.)

4. 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**

**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).

**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.  $(n \div 2=0) \text{ or } (n \div 3<>0)$       b.  $(n \bmod 3=2) \text{ or } (n \bmod 2=3)$   
c.  $(n \bmod 2=0) \text{ and } (n \bmod 3<>1)$       d.  $(n \bmod 2<>1) \text{ and } (n \bmod 3=0)$

**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. Die Variablen  $i$  und  $j$  sind vom Typ `integer` ( $1 \leq i \leq 5, 1 \leq j \leq 5$ ) und die Variable `a` speichert die Elemente eines bidiemnsionalen Feldes mit 5 Reihen und 5 Spalten, nummeriert von 1 bis 5. Der Ausdruck, der den Wert `true` hat, wenn und nur wenn das Element `a[i][j]` sich auf der Nebendiagonale des Feldes befindet, ist: **(4P.)**
  - a.  $i+j=4$
  - b.  $i+j=6$
  - c.  $i < j$
  - d.  $i=j$
2. Die minimale Kantenanzahl eines ungerichteten, konnexen Graphs mit 10 Knoten, ist: **(4P.)**
  - a. 5
  - b. 9
  - c. 10
  - d. 45

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

3. Sei der gerichtete Graph mit 5 Spitzen, nummeriert von 1 bis 5, dargestellt durch die nebenstehende Adjazenzmatrix. Zählt die Spitzen des Graphs, auf die den äußeren Grad gleich mit dem inneren haben. **(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. Sei der Typ `elev`, nebenstehend definiert, der das Deklarieren einer Variablen erlaubt die gewisse Daten über einen Schüler speichern soll: Name und die Jahresmittelnote.  
Mit Hilfe des Typs `elev`, deklariert eine Variable mit dem Namen `clasa`, die die oberen Daten speichern soll für jeden der 30 Schüler einer Klasse, dann schreibt die Anweisungen die die Auslassungspunkte ersetzen können in der unteren Sequenz, um durch Einlesen von der Tastatur, die betreffenden Variable zu initialisieren.  
`for i:=1 to 30 do`  
..... **(6P.)**

```
type elev=record
    nume:string[20];
    medie:real
end;
var i:integer;
```
5. Schreibt ein `Pascal`-Programm das von der Tastatur eine natürliche, von Null verschiedene Zahl  $n$  ( $n \leq 10$ ) einliest und  $n$  Wörter, jedes Wort ist aus wenigstens 3 und höchstens 20 Buchstaben des englischen Alphabetes gebildet. Bei der Dateneingabe wird nach jedem Wort Enter eingegeben. Das Programm baut im Speicher und schreibt nachher auf dem Bildschirm, die Folge die man erhält durch das Übernehmen der ersten drei Buchstaben aus jedem Wort, in der Reihenfolge in der diese eingelesen wurden.  
**Beispiel:** für  $n=3$  und die Wörter  
`frumos`  
`BUN`  
`Intelept`  
baut man die Zeichenfolge `fruBUNInt` **(10P.)**

**THEMA III**

**(30 Puncte)**

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

1. Seien die rekursiven Unterprogramme R1 und R2, weiter unten definiert.

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

Bei dem Aufruf für die Parameter  $x=3$  und  $p=5$ , liefert es den Wert des Ausdruckes  $3^5$ : (4P.)

- a. sowohl R1, als auch R2    b. nur R1    c. nur R2    d. weder R1, noch R2

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

2. Wenn man zur Verfügung fünf verschiedene Blumen hat: **lalea**, **narcisă**, **mac**, **frezie**, **garoafă**, benützt man die Backtracking-Methode um alle Möglichkeiten anzugeben durch die man Strauße von je drei Blumen bilden kann, wenn man weiß, dass innerhalb eines Straußes die Reihenfolge in der die Blumen angeordnet sind nicht wichtig ist. Die ersten vier erhaltenen Lösungen, in dieser Reihenfolge sind: (**lalea**, **narcisă**, **mac**), (**lalea**, **narcisă**, **frezie**), (**lalea**, **narcisă**, **garoafă**), (**lalea**, **mac**, **frezie**). Schreibt die letzten zwei erzeugten Lösungen, in der Reihenfolge in der sie erhalten wurden. (6P.)

3. Das Unterprogramm **eval** hat drei Parameter:

- **a** und **b**, durch die es je eine ganze Zahl erhält mit höchstens vier Ziffern;
- **rez**, durch den es eine Zahl liefert, erhalten durch das addieren der Werte zugeschickt durch die Parameter **a** und **b**.

**Beispiel:** wenn  $a=3$ ,  $b=5$  ist, dann wird nach dem Aufruf, der Wert geliefert durch den Parameter **rez**, 8 sein.

- a) Schreibt nur die Kopfzeile des Unterprogramms **eval**. (4P.)

b) Die Datei **BAC.IN** enthält eine Folge von ganzen Zahlen, mit höchstens zwei Ziffern jede. In der Datei befinden sich wenigstens zwei und höchstens 100 Zahlen, getrennt durch je ein Leerzeichen.

Schreibt ein **Pascal**-Programm, das alle Zahlen aus der Datei **BAC.IN** einliest und als mit Hilfe von gültigen Aufrufen des Unterprogramms **eval**, die Summe dieser Zahlen berechnet und nachher den erhaltenen Wert auf dem Bildschirm anschreibt.

**Beispiel:** wenn die Datei die unterstehenden Zahlen enthält, wird auf dem Bildschirm der Wert 11 angeschrieben.

2 12 -7 4 (6P.)

4. 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)

Wird auf dem Bildschirm der Wert 4 angeschrieben.

(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)	<b>Pentru algoritm pseudocod corect</b> - 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)	<b>Pentru program corect</b> - 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 <sup>1)</sup>	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

<b>4)</b>	<b>Pentru răspuns corect</b> - declararea variabilei de un tip corespunzător - accesul corect la câmpurile înregistrării - inițializarea corectă a celor două câmpuri	<b>6p.</b>  2p. 2p.  2x1p.	
<b>5)</b>	<b>Pentru program corect</b> - 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 <sup>1)</sup>	<b>10p.</b>  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**

<b>1)</b>	<b>a</b>	<b>4p.</b>	
<b>2)</b>	<b>Răspuns corect:</b> (narcisă, frezie, garoafă), (mac, frezie, garoafă) (*)	<b>6p.</b>	(*) 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.
<b>3) a)</b>	<b>Pentru antet corect</b> - structură antet principal corectă - declarare corectă a parametrilor de intrare - declarare corectă a parametrului de ieșire	<b>4p.</b> 1p.  2x1p.  1p.	
<b>b)</b>	<b>Pentru program corect</b> - 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 <sup>1)</sup>	<b>6p.</b>  1p. 1p. 3p.  1p.	
<b>4)</b>	<b>Pentru program corect</b> - 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 <sup>1)</sup>	<b>10p.</b> 1+1p.  1p. 1p. 2p. 2p. 1p.  1p.	(*) Se acordă numai 3p. dacă după prelucrare s-au modificat pozițiile termenilor impari.

<sup>1)</sup> Corectitudinea globală vizează structura, sintaxa și alte greșeli neprecizate în barem.