

Examenul de bacalaureat 2005
Probă scrisă la INFORMATICĂ, Varianta Pascal

VARIANTA 1

Proba E

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

I. Pentru fiecare dintre itemii de la 1 la 8, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 5 puncte.

1. Se consideră secvența Pascal alăturată. Instrucțiunea `write(k)` se execută de:
- ```
for i:=1 to 10 do begin
for j:=1 to i do write(j);
for k:=9 downto 1 do write(k)end
```
- a. 9 ori                      b. 55 de ori                      c. 90 de ori                      d. 495 de ori
2. Un arbore binar format din 9 noduri, numerotate de la 1 la 9, este memorat cu ajutorul următorilor vectori `st` (stânga) și `dr` (dreapta): `st=(8,4,5,0,0,0,2,0,0)` `dr=(9,6,0,0,0,1,3,0,0)`. Rădăcina arborelui este nodul:
- a. 1                      b. 2                      c. 6                      d. 7
3. Numărul minim de componente conexe ale unui graf neorientat cu 9 muchii și 12 noduri este:
- a. 2                      b. 3                      c. 4                      d. 5
4. Variabila `n` este declarată prin `n:integer`. Care dintre următoarele expresii este egală cu cifra zecilor (penultima) a numărului memorat de variabila `n`?
- a. `n mod 10`                      b. `n mod 100 div 10`                      c. `n div 10`                      d. `n div 100 mod 10`
5. Se consideră algoritmul care determină toate permutările distincte de `n` obiecte (numerotate de la 1 la `n`) în care pe orice poziție de rang par se află o valoare pară. De exemplu, pentru `n=5`, primele trei permutări generate în ordine lexicografică sunt: (1,2,3,4,5), (1,2,5,4,3), (1,4,3,2,5). Pentru `n=4`, numărul total de astfel de permutări este:
- a. 2                      b. 4                      c. 8                      d. 12
6. Care este antetul corect al subprogramului `fibonacci`, astfel încât secvența alăturată să realizeze afișarea termenului de rang `n` (`n ≤ 45`) din șirul lui Fibonacci?
- ```
readln(n);
fibonacci(t,n);
writeln(t)
```
- a. `procedure fibonacci(t:longint; var n:integer);`
b. `procedure fibonacci(a:integer; b:longint);`
c. `procedure fibonacci(var t:longint; n:integer);`
d. `function fibonacci(n:integer; var t:longint);`
7. Într-o listă circulară simplu înlănțuită fiecare element reține în câmpul `next` adresa elementului următor al listei. Știind că, pentru variabila `p` ce memorează adresa unui element oarecare din listă, este adevărată relația `p^.next=p^.next^.next^.next`, stabiliți numărul maxim de componente din care poate fi formată lista.
- a. 1 b. 2 c. 3 d. 4
8. Cu ce expresie trebuie înlocuite punctele de suspensie din definiția subprogramului alăturat astfel încât apelul `cif(n)` să returneze numărul de cifre ale numărului natural nenul `n`?
- ```
function cif(n:longint):byte;
begin
if n=0 then cif:=0
else cif:=...
end;
```
- a. `cif(n div 10) + 1`                      b. `n mod 10 + cif(n div 10)`  
c. `n div 10 + cif(n mod 10)`                      d. `1 + cif(n mod 10)`

II. Se consideră programul pseudocod alăturat :

S-a notat cu  $x\%y$  restul împărțirii numărului natural  $x$  la numărul natural  $y$ . S-a folosit notația  $[x]$  pentru a desemna partea întreagă a numărului real  $x$ .

1. Ce valoare se afișează pentru  $n=35$ ? (5p.)
2. Determinați o valoare pentru variabila  $n$  astfel încât, conform algoritmului dat, să nu se afișeze nimic. (3p.)
3. Scrieți programul Pascal echivalent cu algoritmul dat. (8p.)
4. Scrieți algoritmul pseudocod sau programul Pascal echivalent cu algoritmul dat, dar care să utilizeze un alt tip de structură repetitivă. (4p.)

```
citește n (nr. natural, n>1)
i←1
┌Repetă
│ i←i+1
│ s←[n/i]
└până când n%i=0
┌Dacă i<n atunci
│ scrie s
└
```

III.

1. Se citesc  $n$  ( $n$  număr natural,  $1 < n < 100$ ) și apoi 2 numere întregi  $a, b$  ( $0 < a, b < 1000$ ). Să se scrie programul Pascal care construiește în memorie vectorul  $v$  cu  $n$  componente numere întregi (numerotate

$$\text{de la } 1 \text{ la } n) \text{ în care: } v_i = \begin{cases} a & \text{pentru } i = 1 \\ b & \text{pentru } i = 2 \\ v_{i-1} + v_{i-2} & \text{pentru } i > 2 \text{ și } i \text{ impar} \\ |v_{i-1} - v_{i-2}| & \text{pentru } i > 2 \text{ și } i \text{ par} \end{cases}$$

S-a folosit notația  $|x|$  pentru valoarea absolută (modulul) numărului  $x$ . Programul va afișa în final componentele vectorului, în linie, cu spații între orice două componente succesive.

De exemplu, pentru  $n=8$   $a=2$  și  $b=5$ , se construiește și se afișează vectorul:

2 5 7 2 9 7 16 9 (10p.)

2. Subprogramul `ca1c` primește prin intermediul parametrului  $n$  un număr natural nenul de cel mult patru cifre. El returnează numărul de cifre ale numărului obținut prin concatenarea (lipirea) numerelor  $1, 2, \dots, n$ . De exemplu, pentru  $n=12$ , numărul de cifre ale numărului  $123456789101112$  este 15.
  - a) Alegeți un algoritm eficient de calcul și scrieți definiția subprogramului `ca1c`. (5p.)
  - b) Descrieți metoda utilizată și explicați în ce constă eficiența ei (3-4 rânduri). (2p.)
  - c) Scrieți programul care citește de la tastatură două numere naturale nenule  $a, b$  ( $1 < a < b < 10000$ ) și determină numărul de cifre ale numărului obținut prin concatenarea numerelor  $a, (a+1), \dots, (b-1), b$ , folosind apeluri ale subprogramului `ca1c` definit la punctul a). (3p.)
3. Fișierul text **BAC.TXT** conține, pe prima linie, un șir de exact 1000 de caractere (litere, cifre și spații). Scrieți programul Pascal care citește de la tastatură un număr natural  $n$  ( $0 < n \leq 1000$ ) și afișează pe ecran câte caractere cifră există printre primele  $n$  caractere din fișier.  
De exemplu, dacă **BAC.TXT** conține:  

```
mx 12 5 3a63abc12 fcccc...c
```

981 litere c

iar de la tastatură se citește valoarea 11, atunci se va afișa pe ecran numărul 5, deoarece în șirul format din primele 11 caractere există cinci caractere cifră (cele subliniate). (10p.)

**Examenul de bacalaureat 2005**  
**Probă scrisă la INFORMATICĂ, Varianta C/C++**

**VARIANTA 1**

**Proba E**

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

I. Pentru fiecare dintre itemii de la 1 la 8, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 5 puncte.

1. Se consideră secvențele C și respectiv C++ următoare:

|                                                                                                                             |                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <pre>for (i=1;i&lt;=10;i++) {<br/>  for (j=1;j&lt;=i;j++) printf("%d",j);<br/>  for (k=9;k&gt;0;k--) printf("%d",k);}</pre> | <pre>for (i=1;i&lt;=10;i++) {<br/>  for (j=1;j&lt;=i;j++) cout&lt;&lt;j;<br/>  for (k=9;k&gt;0;k--) cout&lt;&lt;k;}</pre> |
|-----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|

În oricare dintre variante, instrucțiunea care afișează valoarea variabilei k se execută de:

- a. 9 ori                      b. 55 de ori                      c. 90 de ori                      d. 495 de ori
2. Un arbore binar format din 9 noduri, numerotate de la 1 la 9, este memorat cu ajutorul următorilor vectori *st* (stânga) și *dr* (dreapta): *st*=(8,4,5,0,0,0,2,0,0) *dr*=(9,6,0,0,0,1,3,0,0)  
Rădăcina arborelui este nodul:
- a. 1                              b. 2                              c. 6                              d. 7
3. Numărul minim de componente conexe ale unui graf neorientat cu 9 muchii și 12 noduri este:
- a. 2                              b. 3                              c. 4                              d. 5
4. Variabila *n* este declarată prin `int n;`. Care dintre următoarele expresii este egală cu cifra zecilor (penultima) a numărului memorat de variabila *n*?
- a. `n%10`                      b. `(n%100)/10`                      c. `n/10`                      d. `(n/100)%10`
5. Se consideră algoritmul care determină toate permutările distincte de *n* obiecte (numerotate de la 1 la *n*) în care pe orice poziție de rang par se află o valoare pară. De exemplu, pentru *n*=5, primele trei permutări generate în ordine lexicografică sunt: (1,2,3,4,5), (1,2,5,4,3), (1,4,3,2,5). Pentru *n*=4, numărul total de astfel de permutări este:
- a. 2                              b. 4                              c. 8                              d. 12
6. Care este antetul corect al subprogramului *fib*, astfel încât secvența alăturată să realizeze afișarea termenului de rang *n* (*n*≤45) din șirul lui Fibonacci?
- |                                                                                                  |  |
|--------------------------------------------------------------------------------------------------|--|
| <pre>scanf("%d",&amp;n); /cin&gt;&gt;n;<br/>fib(t,n);<br/>printf("%ld",t); /cout&gt;&gt;t;</pre> |  |
|--------------------------------------------------------------------------------------------------|--|
- a. `void fibo(long t, int& n)`  
b. `void fibo(int a, long b)`  
c. `void fibo(long& t, int n)`  
d. `long fibo(int n)`
7. Într-o listă circulară simplu înlănțuită fiecare element reține în câmpul *next* adresa elementului următor al listei. Știind că, pentru variabila *p* ce reprezintă adresa unui element oarecare din listă, este adevărată relația `p->next==p->next->next->next`, stabiliți care este numărul maxim de componente din care poate fi formată lista.
- a. 1                              b. 2                              c. 3                              d. 4
8. Cu ce expresie trebuie înlocuite punctele de suspensie din definiția subprogramului alăturat astfel încât apelul `cif(n)` să returneze numărul de cifre ale numărului natural nenul *n*?
- |                                                                                    |  |
|------------------------------------------------------------------------------------|--|
| <pre>int cif( long n)<br/>{ if(n==0) return 0;<br/>  else return ... ;<br/>}</pre> |  |
|------------------------------------------------------------------------------------|--|
- a. `cif(n/10)+ 1`                      b. `n%10 + cif(n/10)`  
c. `n/10 + cif(n%10)`                      d. `1 + cif(n%10)`

II. Se consideră programul pseudocod alăturat :

S-a notat cu  $x\%y$  restul împărțirii numărului natural  $x$  la numărul natural  $y$ . S-a folosit notația  $[x]$  pentru a desemna partea întreagă a numărului real  $x$ .

1. Ce valoare se afișează pentru  $n=35$ ? (5p.)
2. Determinați o valoare pentru variabila  $n$  astfel încât, conform algoritmului dat, să nu se afișeze nimic. (3p.)
3. Scrieți programul C/C++ echivalent cu algoritmul dat. (8p.)
4. Scrieți algoritmul pseudocod sau programul C/C++ echivalent cu algoritmul dat, dar care să utilizeze un alt tip de structură repetitivă. (4p.)

```
citește n (nr. natural, n>1)
i←1
┌Repetă
│ i←i+1
│ s←[n/i]
└până când n%i=0
┌Dacă i<n atunci
│ scrie s
└
```

III.

1. Se citesc  $n$  ( $n$  număr natural,  $1 < n < 100$ ) și apoi 2 numere întregi  $a, b$  ( $0 < a, b < 1000$ ). Să se scrie programul C/C++ care construiește în memorie vectorul  $v$  cu  $n$  componente numere întregi (numerotate

$$\text{de la } 1 \text{ la } n) \text{ în care: } v_i = \begin{cases} a & \text{pentru } i = 1 \\ b & \text{pentru } i = 2 \\ v_{i-1} + v_{i-2} & \text{pentru } i > 2 \text{ și } i \text{ impar} \\ |v_{i-1} - v_{i-2}| & \text{pentru } i > 2 \text{ și } i \text{ par} \end{cases}$$

S-a folosit notația  $|x|$  pentru valoarea absolută (modulul) numărului  $x$ . Programul va afișa în final componentele vectorului, în linie, cu spații între orice două componente succesive.

De exemplu, pentru  $n=8$   $a=2$  și  $b=5$ , se construiește și se afișează vectorul:

2 5 7 2 9 7 16 9 (10p.)

2. Subprogramul `ca1c` primește prin intermediul parametrului  $n$  un număr natural nenul de cel mult patru cifre. El returnează numărul de cifre ale numărului obținut prin concatenarea (lipirea) numerelor  $1, 2, \dots, n$ . De exemplu, pentru  $n=12$ , numărul de cifre ale numărului 123456789101112 este 15.
  - a) Alegeți un algoritm eficient de calcul și scrieți definiția subprogramului `ca1c`. (5p.)
  - b) Descrieți metoda utilizată și explicați în ce constă eficiența ei (3-4 rânduri). (2p.)
  - c) Scrieți programul care citește de la tastatură două numere naturale nenule  $a, b$  ( $1 < a < b < 10000$ ) și determină numărul de cifre ale numărului obținut prin concatenarea numerelor  $a, (a+1), \dots, (b-1), b$ , folosind apeluri ale subprogramului `ca1c` definit la punctul a). (3p.)
3. Fișierul text **BAC.TXT** conține, pe prima linie, un șir de exact 1000 de caractere (litere, cifre și spații). Scrieți programul C/C++ care citește de la tastatură un număr natural  $n$  ( $0 < n \leq 1000$ ) și afișează pe ecran câte caractere cifră există printre primele  $n$  caractere din fișier.

De exemplu, dacă **BAC.TXT** conține:

`mx 12 5 3a63abc12 fcccc...c`

981 litere c

iar de la tastatură se citește valoarea 11, atunci se va afișa pe ecran numărul 5, deoarece în șirul format din primele 11 caractere există cinci caractere cifră (cele subliniate). (10p.)