

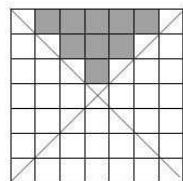
EXAMENUL DE BACALAUREAT – 2007
Proba scrisă la INFORMATICĂ
PROBA E, limbajul C/C++

Varianta 31

- ♦ Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- ♦ Timpul efectiv de lucru este de 3 ore.
- ♦ În programele cerute la subiectele II și III, 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 (40 de puncte)

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

1. Dintre tipurile simple de date face parte tipul:
 - a. `float`
 - b. `fișier`
 - c. `pointer`
 - d. `tablou`
2. Pentru căutarea unei valori între elementele unui vector ordonat descrescător, vom utiliza un algoritm eficient de tip:
 - a. interschimbare
 - b. quick sort
 - c. căutare binară
 - d. backtracking
3. Un arbore cu rădăcină având 9 noduri, numerotate de la 1 la 9, este memorat cu ajutorul vectorului de tați $t = (2, 5, 5, 3, 0, 2, 4, 6, 6)$. Ascendenții nodului 6 sunt:
 - a. nodurile 1 și 4
 - b. doar nodul 2
 - c. nodurile 8 și 9
 - d. nodurile 2 și 5
4. Având la dispoziție cifrele 0, 1 și 2 putem genera, în ordine crescătoare, numere care au suma cifrelor egală cu 2 astfel: 2, 11, 20, 101, 110, 200, etc. Folosind acest algoritm generați numere cu cifrele 0, 1 și 2 care au suma cifrelor egală cu 3. Care va fi al șaptelea număr din această generare?
 - a. 120
 - b. 1002
 - c. 201
 - d. 210
5. Într-o matrice pătratică de dimensiune n , notăm cu $a(i, j)$ elementul situat pe linia i și coloana j ($1 \leq i \leq n$ și $1 \leq j \leq n$). Diagonala principală și cea secundară determină în matrice patru zone triunghiulare. Ce condiție trebuie să îndeplinească indicii elementelor din triunghiul superior al matricei?
 
 - a. $i < j$ și $(i+j) < n+1$
 - b. $i < j$ sau $(i+j) < n$
 - c. $i > j$ și $(i+j) < n$
 - d. $i+j = n$ și $i > j$
6. Cele 4 prietene Dana, Alina, Oana și Maria doresc să stea împreună în clasă, într-o bancă cu 3 locuri. În câte modalități se pot aranja în bancă știind că unul dintre cele 3 locuri îl va ocupa întotdeauna Oana.
 - a. 36
 - b. 24
 - c. 18
 - d. 12
7. Se consideră graful orientat cu 8 noduri, definit cu ajutorul listelor de adiacență alăturate. În acest graf, nodul 1 este legat prin drumuri de lungime 2 de nodurile:

1: 4, 5, 6	5: 4, 1
2: 3, 4	6: 1, 4
3: 4	7: 1, 8
4: 3, 6	8:

 - a. 7, 8
 - b. 5, 6, 4
 - c. 3, 4, 6
 - d. 2
8. Secvența alăturată de program, va afișa :


```
a=99;
while (a>=1)
{ cout<<a;    / printf("%d",a);
  a=a-2; }
```

 - a. toate numerele naturale de două cifre.
 - b. numerele naturale impare mai mici decât 100
 - c. toate numerele întregi mai mici decât 99
 - d. numerele naturale pare, mai mari decât 1

SUBIECTUL II (20 de puncte)

Se consideră programul pseudocod alăturat:

S-a notat cu $x \div y$ restul împărțirii numerelor întregi x și y și cu $[x]$ partea întreagă a numărului real x .

1. Care sunt valorile afișate pentru $n=2456753$? (5p.)
2. Stabiliți o valoare de trei cifre pentru variabila n astfel încât, după executarea programului, a să conțină inversul (definit ca numărul obținut din cifrele numărului inițial, așezate exact în ordine inversă) valorii inițiale a lui n . (2p.)

```

citește n
(numar natural)
a ← 0
cât timp n % 2 ≠ 0 execută
| a ← a * 10 + n % 10
| n ← [n / 10]
scrie n, a

```

3. Scrieți programul C/C++ corespunzător algoritmului dat. (8p.)
4. Scrieți în limbajul C/C++ un subprogram recursiv **sub** astfel încât secvența :

```

citește n
a = 0
sub(n, a)
afișează n, a

```

să fie echivalentă cu algoritmul dat, pentru orice valoare naturală a lui n . (5p.)

SUBIECTUL III (30 de puncte)

1. Se consideră subprogramul **max_diviz** care are doi parametri, a și b (numere naturale având maximum 9 cifre fiecare) și care returnează cel mai mare dintre **divizorii comuni de două cifre** ai numerelor a și b . Dacă cele două numere nu au un astfel de divizor, subprogramul va returna 1.

- a) Scrieți definiția completă a subprogramului **max_diviz**. (4p.)
- b) Scrieți programul C/C++ care citește de la tastatură un număr natural nenul a cu maximum 9 cifre, și determină, folosind apeluri ale subprogramului **max_diviz**, cel mai mare număr de două cifre care divide pe a . Se va afișa pe ecran numărul astfel determinat sau mesajul: **NU exista divizor de 2 cifre**, în cazul în care a nu are nici un divizor de două cifre. De exemplu: pentru $a=160$ se va afișa 80. (6p.)

2. În fișierul **MATRICE.IN** se află memorate elementele unei matrice; pe prima linie se află numerele naturale n și m ($n \leq 10$, $m \leq 10$) separate printr-un spațiu, care reprezintă numărul de linii, respectiv coloane ale matricei, iar pe următoarele n linii câte m numere întregi, separate prin câte un spațiu, reprezentând elementele matricei. Scrieți programul C/C++ care citește din fișierul **MATRICE.IN** elementele matricei și afișează pe ecran numărul coloanelor tabloului formate numai din valori distincte. (10p.)

Exemplu:

```

MATRICE.IN
4 5
1 2 1 5 10
7 3 3 2 8
6 3 4 9 15
4 7 1 12 0

```

Trei coloane au elemente distincte (1, 4 și 5).

Pe ecran se va afișa mesajul: 3

3. Se consideră definite subprogramele:
 - **create**, care citește de la tastatură mai multe cuvinte de cel mult 30 de litere fiecare și creează o listă simplă înălțuită ce conține în fiecare nod al său, în câmpul **info**, câte un cuvânt dintre cele citite, iar în câmpul **adr**, adresa următorului nod din listă. Subprogramul are un singur parametru, p reprezentând adresa de început a listei nou create. În listă cuvintele se vor memora în ordinea citirii lor de la tastatură.

- **cautare**, care caută în lista creată un anumit cuvânt; subprogramul are ca parametri variabila p prin care se transmite adresa de început a listei și variabila x prin care se transmite cuvântul căutat; subprogramul returnează adresa primului nod care conține cuvântul căutat, sau **NULL** în cazul în care acest cuvânt nu este găsit în listă.

- a) Scrieți declarațiile de date ce definesc lista, antetul subprogramului **create** și definiția completă a subprogramului **cautare**. (7p.)

- b) Scrieți programul C/C++ care construiește o listă cu ajutorul subprogramului **create** și caută în lista creată cuvântul **bacalaureat** folosind subprogramul **cautare**. Programul afișează pe ecran un mesaj dacă acest cuvânt a fost găsit sau nu în lista creată. (3p.)

Exemplu: pentru cuvintele introduse: **Succes**, **la**, **bacalaureat**, **tuturor**, **elevilor** se creează lista: **Succes** → **la** → **bacalaureat** → **tuturor** → **elevilor** și se afișează pe ecran mesajul: Cuvântul **bacalaureat** a fost găsit în listă.