

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

Varianta 64

- ♦ 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. Variabila **e** este utilizată pentru a memora următoarele date referitoare la un elev ce participă la examenul de bacalaureat: **numele**, **prenumele** și **media** obținută la examen. **Numele** și **prenumele** conțin fiecare cel mult 15 caractere iar **media** este o valoare reală care urmează să fie utilizată în diverse calcule matematice. Stabiliți care dintre următoarele declarații sunt corecte și folosesc în mod eficient spațiul de memorie:
 - a. `char e[15][15];`
 - b. `struct{char nume,prenume[15];float media;} e;`
 - c. `struct{char nume[16],prenume[16];float media[5];} e;`
 - d. `struct{char nume[16],prenume[16];float media;} e;`
2. Pentru un graf orientat $G(X,V)$ cu n noduri numerotate cu numerele distincte $1,2,\dots,n$, și reprezentat prin matricea de adiacență **a**, secvența de instrucțiuni alăturată descrisă în limbajul pseudocod determină în variabila **nr**:


```

      nr ← 0
      citește k {k natural, k ≤ n}
      pentru i ← 1, n execută
      |   dacă aki = 1 atunci nr ← nr + 1
      |   ■
      |   ■
      |   ■
      
```

 - a. gradul nodului **k**
 - b. gradul exterior al nodului **k**
 - c. gradul interior al nodului **k**
 - d. numărul de elemente egale cu 1 din matricea de adiacență
3. Fie graful neorientat $G(X,V)$, cu $X=\{1,2,3,4,5\}$ și $V=\{[1,2],[2,3],[3,1],[3,4],[4,5],[5,1],[5,3]\}$. Stabiliți care dintre propozițiile următoare este adevărată:
 - a. Numărul vârfurilor de grad par este egal cu numărul vârfurilor de grad impar.
 - b. Matricea de adiacență asociată grafului **G** nu este simetrică față de diagonală secundară.
 - c. Cel mai scurt lanț de la vârful 1 la vârful 4 are lungimea 3
 - d. Subgraful generat de vârfurile $\{1,2,4\}$ nu este conex.
4. Se consideră o stivă implementată prin intermediul vectorului **a** cu elementele : **a**[0]=0, **a**[1]=10, **a**[2]=20, **a**[3]=30, **a**[4]=40, **a**[5]=50. Dacă cel de-al doilea element, începând de la baza stivei, este 10, atunci primul element care iese din stivă este:
 - a. **a**[6]
 - b. **a**[1]
 - c. **a**[5]
 - d. **a**[0]
5. Funcția **numar(a)**, unde **a** este număr întreg, returnează cel mai mic întreg mai mare decât **a**, dacă **a** are cifra unităților egală cu cifra zecilor sau cel mai mare întreg mai mic decât **a** în caz contrar. Două valori **x** și **y** pentru care funcția **numar** returnează același rezultat sunt:
 - a. 155,156
 - b. 100,101
 - c. 12,11
 - d. 124,122
6. Se consideră definiția alăturată a subprogramului **f**. Ce se va afișa în urma apelului **f(13)**?


```

      void f(int n)
      { if (n>0) f(n/10);
        printf("%d",n%10);/cout<<n%10;
      }
      
```

 - a. 013
 - b. 310
 - c. 31
 - d. 13
7. Dacă **a,b,z** sunt variabile reale și **a ≤ b**, atunci care dintre expresiile următoare are valoarea adevărată dacă și numai dacă **z ∉ [a,b]**?
 - a. **z < a || z > b**
 - b. **z > a || z > b**
 - c. **z < a && z > b**
 - d. **z >= a && z <= b**
8. Generarea matricelor pătratice de ordinul **n**, cu elemente 0 și 1, cu proprietatea că pe fiecare linie și pe fiecare coloană există un singur element egal cu 1, se poate realiza utilizând metoda backtracking. Algoritmul utilizat este echivalent cu algoritmul de generare a:
 - a. produsului cartezian
 - b. aranjamentelor
 - c. combinațiilor
 - d. permutărilor

SUBIECTUL II (20 de puncte)

Se consideră programul pseudocod alăturat care prelucrează numerele naturale nenule dintr-un tablou unidimensional a , de dimensiune n . S-a notat cu $x \% y$ restul împărțirii numerelor întregi x și y .

1. Pentru $n=7$ și $a=(3,7,5,49,21,77,70)$ precizați ce se afișează în urma executării algoritmului. (7p.)
2. Pentru $n=7$ dați un exemplu de valori pentru a , astfel încât variabila nr să memoreze la finalul executării algoritmului cea mai mare valoare posibilă. (3p.)
3. Pentru $n=5$ scrieți un exemplu de valori pentru a astfel încât rezultatul afișat în urma executării algoritmului să fie: 0 0. (2p.)
4. Scrieți programul C/C++ corespunzător algoritmului din enunț. (8p.)

```

citește n
(n număr natural, 1<n≤100)
pentru i←1,n execută
    citește ai
    max←0; nr←0;
    pentru i←1,n execută
        dacă ai%7=0 atunci
            nr←nr+1
            dacă ai>max atunci
                max←ai
scrie max,nr

```

SUBIECTUL III (30 de puncte)

1. Scrieți un program C/C++ care citește de la tastatură două numere naturale nenule n și m ($n<100$, $m<100$), determină și afișează pe ecran toate perechile a, b de numere naturale nenule distincte, cel mult egale cu n , care au cel mai mic multiplu comun egal cu m . Perechile se vor afișa pe câte o linie, iar numerele corespunzătoare fiecărei perechi vor fi separate printr-un spațiu, ca în exemplu.
De exemplu, pentru $n=6$ și $m=6$ se afișează perechile:
1 6
2 3
2 6
3 6 (10p.)
2. Două tablouri unidimensionale a și b , cu elementele a_1, a_2, \dots, a_n , respectiv b_1, b_2, \dots, b_n sunt în relația $a \leq b$ dacă: $a_1 \leq b_1, a_2 \leq b_2, \dots, a_n \leq b_n$.
Scrieți definiția completă a unui subprogram în limbajul C/ C++ care primește:
- prin intermediul parametrilor a și b două tablouri unidimensionale cu același număr de elemente de tip `int`;
- prin intermediul parametrului n (n număr natural, $1 < n \leq 1000$) numărul de elemente pe care îl are fiecare dintre cele două tablouri.
Subprogramul returnează valoarea 1 dacă $a \leq b$ și 0 în caz contrar. (10p.)
3. Într-o listă simplu înlănțuită alocată dinamic fiecare element conține în câmpul `info` un număr din șirul Fibonacci și în câmpul `adru` adresa elementului următor din listă. Numerele Fibonacci din listă sunt consecutive, în ordine crescătoare, astfel că primele două elemente ale listei rețin fiecare valoarea 1. Lista are cel mult 30 de elemente.
a) Definiți tipurile de date necesare prelucrării listei. (1p.)
b) Scrieți doar antetul subprogramului `sub` cu doi parametri p și n , care creează lista din enunț. Parametrul n reprezintă numărul de elemente din listă (n natural, $2 \leq n \leq 30$) iar subprogramul returnează prin intermediul parametrului p adresa primului element din listă. (5p.)
c) Scrieți programul C/ C++ care construiește lista utilizând subprogramul `sub`, determină două elemente ale listei pentru care diferența dintre numerele Fibonacci conținute este maximă și afișează numerele Fibonacci reținute de cele două elemente. (4p.)