

**Examenul de bacalaureat național 2015**  
**Proba E. d)**  
**Informatică**  
**Limbajul C/C++**

**MODEL**

*Filiera teoretică, profilul real, specializările: matematică-informatică*

*matematică-informatică intensiv informatică*

*Filiera vocațională, profilul militar, specializarea matematică-informatică*

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv 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).
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.

**SUBIECTUL I**

**(30 de puncte)**

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Valoarea expresiei `c/c++` alăturate este: (4p.) |  $5 \cdot 9 / 2 \cdot 3$
- a. 7                      b. 7.5                      c. 66                      d. 67.5

**2. Se consideră algoritmul alăturat, reprezentat în pseudocod.**

S-a notat cu  $x \% y$  restul împărțirii numărului natural  $x$  la numărul natural nenul  $y$  și cu  $[z]$  partea întregă a numărului real  $z$ .

- a) Scrieți numărul afișat dacă se citește valoarea 216553. (6p.)
- b) Scrieți două numere de trei cifre care pot fi citite, astfel încât în urma executării algoritmului, pentru fiecare dintre acestea, să se afișeze valoarea 1. (4p.)
- c) Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască structura **cât timp...execută** cu o structură repetitivă cu test final. (6p.)
- d) Scrieți programul `C/C++` corespunzător algoritmului dat. (10p.)

```
citește n
  (număr natural nenul)
m ← 0
p ← 1
cât timp n ≥ 10 execută
  c1 ← n % 10
  n ← [n / 10]
  c2 ← n % 10
  dacă c1 ≠ c2 atunci
    dacă c1 > c2 atunci
      m ← m + (c1 - c2) * p
    altfel
      m ← m + (c2 - c1) * p
  scrie m
  p ← p * 10
scrie m
```

**SUBIECTUL al II-lea**

**(30 de puncte)**

**Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Un arbore cu 4 noduri, numerotate de la 1 la 4, poate fi reprezentat prin vectorul de „tați”:  
**(4p.)**

- a. (2, 0, 3, 1)      b. (2, 1, 0, 3)      c. (4, 3, 2, 1)      d. (4, 3, 4, 0)

2. Variabila `s` poate memora un șir cu maximum 20 de caractere. În urma executării secvenței de instrucțiuni alăturate se afișează:  
**(4p.)**

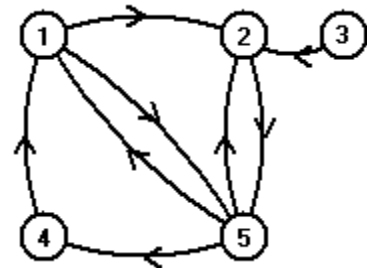
```
strcpy(s, "3x2y1");  
s[2]='x'+2;  
strcpy(s, s+1);  
strcpy(s+3, s+4);  
cout<<s; | printf("%s", s);
```

- a. `xzy`      b. `xyy`      c. `3x417`      d. `42241`

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

3. Scrieți listele de adiacență prin care poate fi reprezentat graful orientat cu 5 vârfuri, numerotate de la 1 la 5, ilustrat în figura alăturată.

**(6p.)**



4. Se consideră graful neorientat cu 9 noduri, numerotate de la 1 la 9, și muchiile  $[1, 2]$ ,  $[1, 4]$ ,  $[2, 3]$ ,  $[2, 5]$ ,  $[2, 7]$ ,  $[3, 7]$ ,  $[4, 5]$ ,  $[5, 6]$ ,  $[5, 8]$ ,  $[6, 9]$ ,  $[8, 9]$ . Scrieți muchiile care pot forma un graf parțial al său care să fie arbore. **(6p.)**

5. Scrieți un program C/C++ care citește de la tastatură două numere naturale,  $m$  și  $n$  ( $2 \leq m \leq 20$ ,  $2 \leq n \leq 20$ ), și construiește în memorie un tablou bidimensional  $A$ , cu  $m$  linii și  $n$  coloane, astfel încât parcurgându-l linie cu linie, de jos în sus, și fiecare linie de la dreapta la stânga, să se obțină șirul primelor  $n \cdot m$  numere naturale, impare, care **NU** sunt divizibile cu 3, ordonat strict crescător. Programul afișează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele de pe aceeași linie fiind separate prin câte un spațiu.

**Exemplu:** pentru  $m=4$  și  $n=3$  se obține tabloul alăturat.

**(10p.)**

35	31	29
25	23	19
17	13	11
7	5	1

**SUBIECTUL al III-lea**

**(30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking, se generează toate posibilitățile de a forma succesiuni de câte 5 tipuri de creații literare din mulțimea {baladă, basm, eseu, epopee, poem}, astfel încât în fiecare succesiune eseu precede epopeea. Două succesiuni sunt distincte dacă tipurile de creații literare sunt în altă ordine.  
Primele cinci soluții generate sunt, în această ordine, (baladă, basm, eseu, epopee, poem), (baladă, basm, eseu, poem, epopee), (baladă, basm, poem, eseu, epopee), (baladă, eseu, basm, epopee, poem), (baladă, eseu, basm, poem, epopee). Imediat înainte de (poem, eseu, epopee, baladă, basm) este generată soluția: **(4p.)**
- a. (basm, baladă, epopee, eseu, poem)      b. (basm, baladă, eseu, epopee, poem)  
c. (poem, eseu, basm, epopee, baladă)      d. (poem, basm, eseu, epopee, baladă)

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul `f`, definit alături. Scrieți valorile `f(10,5)` și `f(20,9)`. **(6p.)**
- ```
int f(int a, int b)
{ if (a<b) return a;
  if (a%b==0) return b;
  return f(a-1,b+2);
}
```
3. Un număr natural nenul se numește **subperfect** dacă este strict mai mic decât suma divizorilor săi proprii (divizori naturali diferiți de 1 și de el însuși).  
**Exemplu:** 12 este număr subperfect pentru că  $12 < 2+3+4+6$ .  
Se consideră subprogramul `subperfect`, cu doi parametri, `a` și `b`, prin care primește câte un număr natural ( $2 \leq a < b \leq 10^9$ ). Subprogramul afișează pe ecran, separate prin câte un spațiu, în ordine descrescătoare, toate numerele subperfecte din intervalul `[a,b]`. Dacă în interval nu există astfel de numere, subprogramul afișează pe ecran mesajul `nu exista`.  
Scrieți definiția completă a subprogramului.  
**Exemplu:** pentru `a=10` și `b=20`, se afișează pe ecran: 20 18 12 **(10p.)**
4. Fișierul `bac.txt` conține pe prima linie un număr natural, `n` ( $1 \leq n \leq 10^6$ ), iar pe a doua linie cel mult 1000000 de numere naturale de forma  $2^p$  ( $0 \leq p \leq 9$ ), separate prin câte un spațiu. Se cere să se afișeze pe ecran numărul care ar apărea pe poziția `n` în șirul ordonat descrescător obținut din toate numerele aflate pe a doua linie a fișierului. Dacă șirul are mai puțin de `n` termeni, se afișează pe ecran mesajul `Nu exista`.  
Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al timpului de executare.  
**Exemplu:** dacă fișierul `bac.txt` conține numerele  
5  
16 32 1 64 128 32 128 32  
atunci pe ecran se afișează valoarea  
32  
a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**  
b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**