MODEL DE SUBIECT
Examenul național de bacalaureat
Informatică - Limbajul C++

Prof. Lungescu Violeta – Cristina
Colegiul Național Pedagogic "Regina Maria" Municipiul Ploiești

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică
Filieră vocațională, profil militar, specializare matematică-informatică

SUBIECTUL I (20 de puncte)
Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Indicați o expresie C/C++ echivalentă cu negarea următoarei expresii: (a>=-2) && (a<10) || (a!=b).

a. (a< -2) || !((a>=10) && (a==b))
b. (a< -2) || (a>=10) && (a!=b)
c. (a< -2) || !((a<10) || !(a!=b))
d. (a< -2) || (a>=10) && !(a!=b)

2. Subprogramul f este definit alăturat. Indicați valoarea f(6);?

```
void f (int n)
{ if (n>1)
     n=n-1;  cout<<n;
     f(n-1);  cout<<n-1;
}
```

a. 531-113 b. 531024 c. 531420 d. 420135

3. Folosind metoda Backtracking se generează toate șirurile formate din patru caractere distincte ale mulțimii {#, $, %, &}. Primele trei soluții generate sunt: #$%&, #$&%, #%$&. Care este soluția care se va genera imediat după: $#&%?
a. $%&#
b. $&#%
c. $%#&
d. $&%#

4. Un arbore cu 4 noduri, numerotate de la 1 la 4, poate fi reprezentat prin vectorul de "tați":
a. (2,0,3,1) b. (2,1,0,3) c. (4,3,4,0) d. (4,3,2,1)

5. Un graf neorientat complet are 8 noduri. Pentru a obține un graf parțial al său cu două componente conexe, fiecare dintre acestea fiind grafuri complete, numărul maxim de muchii care trebuie eliminate este:
a. 16 b. 15 c. 17 d. 14

SUBIECTUL II (40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod. S-a notat cu x%y restul împărțirii numărului natural x la numărul natural nenul y și cu [z] partea întreagă a numărului real z.
   a. Scrieți numărul afișat dacă se citesc, în această ordine, numerele 11 și 60. **(6p.)**
   b. Dacă pentru variabila a se citește valoarea 12345, scrieți:
      - prima valoare a lui x pentru care variabila nr devine 1;
      - cel mai mare număr de cinci cifre care poate

```
citește a, b (numere naturale nenule, a≤b)
nr<-0
pentru i<-a,b execută
    x<-i
    c<-x%10
    cât timp x≠0 și x%10=c execută
        x<-[x/10]
    dacă x=0 atunci
        nr<-nr+1
```

fi citit pentru variabila b astfel încât, în urma executării algoritmului, valoarea afișată să fie 5. **(6p.)**

c. Scrieți programul C/C++ corespunzător algoritmului dat**.** **(10p.)**

d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat structură repetitivă **pentru ... execută** cu o structură repetitivă cu test final. **(6p.)**

```
scrie nr
```

2. În definiția alăturată, variabilele A, B și C memorează în câmpurile x și y coordonatele punctelor A, B și C în sistemul de coordonate xOy. Variabila t memorează în A, B și C coordonatele vârfurilor unui triunghi în același sistem de coordonate.
Scrieți o instrucțiune C/C++ care să verifice dacă ΔABC este echilateral, caz în care afișează mesajul "Este echilateral", altfel, mesajul "Nu este echilateral". **(6p)**

```
struct triunghi
{  struct
     {
        int x,y;
     } A,B,C;
   } t;
```

3. Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 7 linii și 7 coloane, numerotate de la 1 la 7, având inițial toate elementele egale cu 0.
Scrieți secvența de mai jos, înlocuind punctele de suspensie, astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul alăturat.

```
for (i=1; i<=7; i++)
    for (j=1; j<=7; j++)
        …….
```

(6p.)

1	**2**	**3**	**4**	**5**	**6**	**7**
2	**2**	**2**	**2**	**2**	**2**	**8**
3	**3**	**3**	**3**	**3**	**3**	**9**
4	**4**	**4**	**4**	**4**	**4**	**10**
5	**5**	**5**	**5**	**5**	**5**	**11**
6	**6**	**6**	**6**	**6**	**6**	**12**
7	**8**	**9**	**10**	**11**	**12**	**13**

SUBIECTUL III (30 de puncte)

1. Subprogramul **simulare** are trei parametri:
   • **a și b** prin care primește câte un număr natural (a∈[0,1000], b∈[a,1000]) ;
   • **nr** prin care furnizează cel mai mare divizor comun al numerelor a și b.
   Scrieți definiția completă a subprogramului.
   Exemplu: dacă a=4, b=50, în urma apelului, nr=2. **(10p)**

2. Scrieți programul C++ care citește de la tastatură un șir de cel mult 256 de caractere, litere mici ale alfabetului englez, separate prin câte un spațiu și afișează la monitor toate sufixele de lungime cel puțin 1 ale fiecărui cuvânt, pe același rând, separate prin virgulă, în ordinea descrescătoare a lungimii sufixelor, aliniate la stânga.
   Exemplu: dacă se citește șirul **simulare de bacalaureat**, atunci se afișează:
   simulare, imulare,mulare,ulare,lare,are,re,e
   de,e
   bacalaureat,acalaureat,calaureat,alaureat,laureat,aureat,ureat,reat,eat,at,t **(10p)**

3. La un concurs sunt **n (0<n<21)** echipe de elevi care poartă tricouri de **p** culori **(0<p<11)**. Mai multe echipe pot purta tricouri de aceeași culoare. Fiecare culoare este reprezentată printr-o valoare din intervalul [1, p].
   Scrieți un program C/C++ care să afișeze pe ecran culoarea tricoului purtat de cei mai mulți elevi. Proiectați un algoritm eficient din punct de vedere al timpului de executare.
   Datele de intrare se citesc din fișierul text **CONCURS.TXT** care are următoarea structură: pe prima linie se află valorile lui **n**, respectiv **p**, separate printr-un spațiu, iar pe fiecare din următoarele **n** linii câte două valori separate printr-un spațiu ce reprezintă numărul de elevi dintr-o echipă și respectiv valoarea corespunzătoare culorii tricourilor pe care aceștia le poartă.

Exemplu: dacă fișierul **CONCURS.TXT** are următorul conținut:

5 3
20 2
20 3
30 2
20 1
10 1

atunci pe ecran se va afișa **2** deoarece tricourile de culoarea **2** sunt purtate de **50** de elevi, culoarea **1** de **30** de elevi, iar culoarea **3** de **20** de elevi**.**

a. **Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p)**
b. **Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p)**

BAREM DE EVALUARE ȘI DE NOTARE

SUBIECTUL I (20 puncte)

1d 2b 3c 4c 5a	5x4p.

SUBIECTUL al II-lea (40 puncte)

1	**a) Răspuns corect: 5**	6p	Se acordă 6p. dacă răspunsul este corect.
	b) Răspuns corect: 22222 77776	6p	Pentru fiecare număr dintre cele două se acordă câte 3p. dacă este conform cerinței și numai câte 1p. dacă are în scrierea sa 5 cifre, dar nu este primul/ maxim sau nu aparține intervalului indicat.
	c) Pentru program corect -declarare a variabilelor -citire a datelor -afișare a datelor -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului.	**10p.** 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului	**6p.** 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă de tipul cerut, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă de structură repetitivă de tipul cerut (chiar dacă este neechivalentă cu algoritmul cerut).
2	**Pentru rezolvare corectă** -instrucțiunea de decizie (*) -afișare a datelor conform cerinței -corectitudine globală a secvenței	**6p.** 4p. 1p. 1p.	(*) Se acordă 1p. pentru utilizarea unei instrucțiunii de decizie, chiar dacă condiția nu rezolvă cerința problemei; Se acordă 1p. pentru condițiile din instrucțiunea de decizie, conform cerinței; Se acordă 2p. pentru referirea corectă a structurilor definite, câte 1p. pentru câmpurile fiecarei structuri.
3	**Pentru rezolvare corectă** -acces la un element al tabloului -atribuire a valorilor indicate elementelor tabloului (*) -corectitudine globală a secvenței	**6p.** 1p. 4p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect specific (valori suport utilizate, corespondență valori-elemente suport) conform cerinței.

SUBIECTUL al III-lea (30 puncte)

1	**Pentru subprogram corect** -antet subprogram (*) -determinare a valorii cerute (**) -transmiterea prin referință -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului	**10p.** 2p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri de intrare) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect specific (algoritm pentru calculul celui mai mare divizor comun a două numere, valori suport, identificare c.m.m.d.c.) conform cerinței. Se acordă doar 2 puncte dacă a fost descris un algritm principial corect de determinare a celui mai mare divizor comun a două numere.
2	**Pentru program corect** -declarare a unei variabile care să memoreze un șir de caractere -citire a datelor -afișare a datelor șirului conform cerinței (*) -declarare a variabilelor simple, corectitudine globală a programului	**10p.** 1p. 1p. 7p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect specific (parcurgerea întregului șir, identificarea unui cuvânt din șir, eliminarea primei litere a fiecărui cuvânt, identificarea literelor suport, identificarea sufixelor unui cuvânt, afișarea sufixelor cuvântului pe același rând, afișarea sufixelor tuturor cuvintelor din șir pe rânduri diferite) conform cerinței. Se va puncta variante de trasnformare a sirului prin parcurgerea sirului caracter cu character, dar si variante care folosesc extragerea cuvintelor (functia strtok).
3	**a) Pentru răspuns corect** -coerență a descrierii algoritmului -justificare a elementelor de eficiență	**2p.** 1p. 1p.	
	b) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -verificare a proprietății cerute (*),(**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, afișare a datelor, corectitudine globală a programului	**8p.** 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar (de complexitate O(n)). O soluţie posibilă utilizează un vector de frecvență, vf (vf[x]=numărul de elevi care poartă tricouri de culoare x, având inițial toate elementele nule, care se actualizează pe măsura citirii datelor din fișier; după citirea datelor, se parcurge vectorul de frecvență și la final se afișează codul de culoare care are frecvența maximă).