

(30 de puncte)

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

1. Variabilele i și j sunt de tip întreg, iar variabila m memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 0 la 4, cu elemente numere întregi.
O expresie C/C++ a cărei valoare este egală cu produsul dintre primul element de pe linia i și ultimul element de pe coloana j din acest tablou este: **(4p.)**
- a. $m[0,i]*m[j,4]$ b. $m(0,i)*m(j,4)$
c. $m(i)(0)*m(4)(j)$ d. $m[i][0]*m[4][j]$
2. Un graf orientat are 12 arce, 3 componente tare conexe, iar fiecare vârf al său are gradul interior un număr nenul. Numărul maxim de noduri pe care le poate avea graful este: **(4p.)**
- a. 12 b. 11 c. 9 d. 8

Scrieti pe foaia de examen raspunsul pentru fiecare dintre cerintele urmatoare.

- | | |
|--|---|
| <p>3. Un arbore cu 8 noduri, numerotate de la 1 la 8, este reprezentat prin următorul vector de „tați” (3, 0, 2, 5, 2, 5, 1, 5). Determinați cel mai lung lanț elementar care are o extremitate în rădăcină și enumerați nodurile sale, în ordinea apariției în acesta. (6p.)</p> <p>4. Variabilele i și s sunt întregi, iar variabila p memorează, pentru fiecare dintre cele 20 de zone de parcare ale unui oraș, numărul de locuri închiriate pe parcursul orei curente, precum și prețul practicat de zona respectivă pentru închirierea unui loc pentru o oră.</p> | <pre>struct parcare { int nrLocuriInchiriate; int pretOra; } p[20];</pre> |
|--|---|

Fără a utiliza alte variabile decât cele menționate, scrieți o secvență de instrucțiuni în urma executării căreia variabila s să memoreze suma totală obținută în urma închirierii locurilor de parcare din oraș pe parcursul orei curente. **(6p.)**

5. Scrieți un program C/C++ care citește de la tastatură, în această ordine, un număr natural n , apoi n cuvinte, separate prin Enter, urmate de un număr natural k . Numerele citite sunt din intervalul $[1, 20]$, iar fiecare cuvânt este format din cel mult 20 de caractere, numai litere mici ale alfabetului englez. Cel puțin unul dintre cuvinte are k litere. Programul construiește în memorie, apoi afișează pe ecran, un șir care să cuprindă cuvintele citite, separate prin câte un spațiu, astfel încât toate cele care au k litere să ocupe primele poziții, iar celelalte să se regăsească în continuarea acestora, într-o ordine oarecare.

Exemplu: dacă se citesc, în această ordine, datele alăturate, unul dintre şirurile obţinute poate fi:
am un de cires mai

(10p.)	cires de mai 2
--------	-------------------------

SUBIECTUL al III-lea

(30 de puncte)

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

1. Subprogramul `f` este definit alăturat. Indicați cea mai mare valoare din intervalul `[20,25]` pe care o poate avea variabila întreagă `x`, astfel încât, în urma apelului de mai jos, să se afișeze numerele 2 4 16.
`f(x,1);`

(4p.)

```
void f(int n, int p)
{ if(n>0)
  { if(n%2==1)
    cout<<p<<" "; | printf(" ",p);
    f(n/2, p*2);
  }
}
```

- a. 21 b. 22 c. 23 d. 24

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

2. Utilizând metoda backtracking se generează, în ordine lexicografică, toate șirurile de câte 6 cifre din mulțimea $\{0,1\}$ cu proprietatea că au cel mult două cifre cu valori egale pe poziții consecutive. Primele 5 soluții generate sunt, în această ordine: 001001, 001010, 001011, 001100, 001101. Scrieți a 7-a și a 8-a soluție, în ordinea generării acestora. **(6p.)**
3. Subprogramul `minDivPrim` are un singur parametru, `n`, prin care primește un număr natural ($n \in [2, 10^9]$). Subprogramul returnează cel mai mic număr natural care are aceeași divizori primi ca `n`.
Scrieți definiția completă a subprogramului.
Exemplu: dacă `n=75`, subprogramul returnează numărul 15, iar dacă `n=7`, subprogramul returnează numărul 7. **(10p.)**
4. Se consideră șirul 1, -1, 2 ... definit astfel: $f_1=1$, $f_2=-1$, iar $f_n=1-2 \cdot f_{n-1}-f_{n-2}$, dacă $n \geq 3$ (unde n este un număr natural).
Se citește de la tastatură un număr natural, `n` ($n \in [1, 10^9]$), și se cere să se scrie în fișierul text `bac.out`, separați prin câte un spațiu, primii `n` termeni ai șirului, în ordine inversă apariției lor în acesta.
Pentru determinarea și afișarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al spațiului de memorie și al timpului de executare.
Exemplu: dacă `n=3`, fișierul conține valorile
2 -1 1
a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**

Examenul de bacalaureat național 2016
Proba E. d)
Informatică

Barem de evaluare și de notare
(comun pentru limbajele C/C++ și Pascal)

Varianta 4

Filiera teoretică, profilul real, specializările: matematică-informatică

matematică-informatică intensiv informatică

Filiera vocațională, profilul militar, specializarea matematică-informatică

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct.
- Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.
- Se vor lua în considerare atât implementările concepute pentru compilatoare pe 16 biți, cât și cele pentru compilatoare pe 32 de biți.

SUBIECTUL I

(30 de puncte)

1.	a	4p.	
2.	a) Răspuns corect: 3	6p.	
	b) Răspuns corect: 18	4p.	
	c) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
	d) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiuni de decizie conform cerinței (*) -instrucțiune repetitivă conform cerinței -atribuiri corecte -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 3p. 2p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiuni este conform cerinței.

SUBIECTUL al II - lea

(30 de puncte)

1.	d	4p.	
2.	a	4p.	
3.	Pentru rezolvare corectă -enumerare a nodurilor unui lanț -parametri pentru lanț (*)	6p. 2p. 4p.	Se acordă câte 2p. pentru fiecare parametru conform cerinței (lanț elementar, lungime maximă).

4. Pentru rezolvare corectă -acces la câmpurile unui element al tabloului (*) -determinare a sumei cerute (**)	6p. 3p. 3p.	(*) Se acordă numai 1p. dacă s-a utilizat operatorul de selecție al unui câmp, dar nu conform cerinței. (**) Se acordă numai 1p. dacă algoritmul de calcul al unei sume este principial corect, dar nu conduce la rezultatul cerut.
5. Pentru program corect -declarare corectă a unei variabile care să memoreze mai multe cuvinte -citire a datelor -determinare a șirului cerut (*) -afișare a datelor în formatul cerut -declarare a variabilelor simple, corectitudine globală a programului ¹⁾	10p. 1p. 1p. 6p. 1p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (plasare a cuvintelor de lungime k conform cerinței, plasare a cuvintelor de lungime diferită de k conform cerinței, construire șir în memorie).

SUBIECTUL al III - lea

(30 de puncte)

1.	b	4p.	
2.	Răspuns corect: 010011 010100	6p.	Se acordă câte 3p. pentru fiecare aspect al cerinței (conținut prima soluție, conținut a doua soluție, ordinea soluțiilor).
3.	Pentru subprogram corect -structură antet principial corectă -declarare corectă a parametrului -determinare a numărului cerut (*) -returnare a rezultatului -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului ¹⁾	10p. 1p. 1p. 6p. 1p. 1p.	(*) Se acordă numai 3p. dacă s-a determinat un număr cu aceiași divizori primi, dar nu este cel mai mic.
4. a)	Pentru răspuns corect -coerență a descrierii metodei (*) -justificare a unor elemente de eficiență (**)	4p. 2p. 2x1p.	(*) Se acordă punctajul chiar dacă metoda aleasă nu este eficientă. (**) Se acordă punctajul pentru elementele de eficiență justificate, conform cerinței, chiar dacă metoda aleasă nu este optimă.
b)	Pentru program corect -operații cu fișiere: declarare, pregătire în vederea scrierii, scriere în fișier -citire date, determinare și afișare a valorilor cerute (*, **) -utilizare a unui algoritm eficient (***)	6p. 1p. 4p. 1p.	(*) Se acordă punctajul chiar dacă soluția propusă nu prezintă elemente de eficiență. (**) Se acordă numai 2p. pentru un algoritm principial corect, dar care nu conduce la rezultatul cerut. (***) Se acordă punctajul numai pentru un algoritm liniar (de complexitate $O(n)$), care utilizează eficient memoria. O soluție posibilă se bazează pe forma generală a unui termen: $f_i = (i+1)/2$, dacă i este impar, respectiv $f_i = i/2$, dacă i este par.

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.